Lösung nichtlinearer Gleichungen mit einer Variablen

Manfred Hanke (Gruppe 5: Hanke/Nieva)

29. Nov 2006

1 Zielsetzung

Es sollen die Nullstellen der Funktionen $f(x) = x^3 - 5x^2 + 100x - 400$ und $g(x) = e^{-x} - \frac{0.1}{1+x}$ bestimmt werden.

2 Methoden

Zur Lösung von $y(x) = 0$ bieten sich folgende Methoden an:

2.1 Bisektions-Verfahren

Ist eine Nullstelle zwischen x_1 und x_2 eingeschlossen, kann man iterativ das Intervall $[x_1,\ x_2]$ durch $x = \frac{x_1+x_2}{2}$ halbieren, überprüfen, ob die Nullstelle in $[x_1,\ x]$ oder $[x,\ x_2]$ liegt und dann entweder x_2 oder x_1 durch x ersetzen.

2.2 Newton-/Tangenten-Verfahren

Mit bekannter Ableitung $y'(x)$ der Funktion $y(x)$ werden aus einem anfänglichem Näherungswert x_1 rekursiv $x_{i+1} = x_i - \frac{y(x_i)}{y'(x_i)}$ berechnet.

2.3 Regula Falsi / Sekanten-Verfahren

Bei unbekannter Ableitungsfunktion kann $y'(x_i) \approx \frac{y(x_i)-y(x_{i-1})}{x_i-x_{i-1}}$ approximiert und in $x_{i+1} = x_i - \frac{y(x_i)}{y'(x_i)}$ eingesetzt werden.

2.4 Fixpunktverfahren

Kann $y(x) = 0$ zu $x = X(x)$ umformuliert werden, kann man aus einem Startwert x_1 iterativ $x_{i+1} = X(x_i)$ berechnen.

3 Durchführung

3.1 Bisektions-Verfahren

Dieses Verfahren wird in der `SUBROUTINE Bisect` umgesetzt, der neben der Funktion y die eine Nullstelle einschließende Anfangswerte $x_{1,0}$ und $x_{2,0}$ übergeben werden müssen.
Eine Nullstelle ist genau dann von x_1 und x_2 eingeschlossen, wenn entweder $y(x_1) < 0 \wedge y(x_2) > 0$ oder $y(x_1) > 0 \ \wedge\ y(x_2) < 0$ gilt, d.h. genau dann, wenn $y(x_1) \cdot y(x_2) < 0$.
Zuerst wird diese Bedingung für die Anfangswerte überprüft und das Programm bei ungültigen Übergabewerten abgebrochen. Dann wird in einer Schleife der Mittelwert x berechnet und mit einer einzigen `IF`-Abfrage überprüft, ob x_1 oder x_2 durch x ersetzt werden muß. Die Schleife wird solange wiederholt, bis der Abstand von x_1 und x_2 kleiner als 10^{-6} ist.

3.2 Newton-/Tangenten-Verfahren

Dieses Verfahren wird in der `SUBROUTINE Tangnt` umgesetzt, der die Funktion y, ihre Ableitungsfunktion y', sowie ein Anfangswert x_0 übergeben werden muß. In einer Schleife wird zuerst die Ableitung $y'(x)$ berechnet. Wenn diese Ableitung von Null verschieden ist, wird der neue Wert für x berechnet und einer Hilfsvariablen `xnew` zugewiesen, ansonsten bricht das Programm ab. Die Schleife wird solange wiederholt, bis die Änderung von `x` auf `xnew` kleiner als 10^{-6} ist. Selbstverständlich wird am Anfang der Schleife `x` durch `xnew` ersetzt.

3.3 Regula Falsi / Sekanten-Verfahren

Dieses Verfahren wird in der `SUBROUTINE RegFal` umgesetzt, der die Funktion y und zwei Anfangswerte x_1 und x_2 übergeben werden müssen.
Die Durchführung erfolgt analog zur `SUBROUTINE Tangnt`, nur daß der vorherige Wert in der Hilfsvariablen `xold` mitgenommen wird und die Ableitung dann aus dem Differenzenquotienten berechnet wird.

3.4 Fixpunktverfahren

Eine mögliche Umformulierung von $f(x) = \mathbf{x} \cdot (x^2 - 5x + 100) - 400 = 0$ ist $\mathbf{x} = \frac{400}{x^2-5x+100}$.
$g(x) = e^{-x} - \frac{0.1}{1+\mathbf{x}} = 0$ kann zu $\mathbf{x} = 0.1 \cdot e^x - 1$ umgestellt werden. Eine andere Möglichkeit ist, $g(x) = e^{-\mathbf{x}} - \frac{0.1}{1+x} = 0$ nach $\mathbf{x} = \ln(\frac{1+x}{0.1})$ ‘aufzulösen’.

Das Fixpunktverfahren wird in der `SUBROUTINE FixPt` umgesetzt, der neben der zu iterierenden Funktion `fixpty` ein Anfangswert x_0 übergeben werden muß.
Die Durchführung ist denkbar einfach: In einer Schleife wird solange `xnew` der Rückgabewert von `FixPt(x)` zugewiesen, bis die Änderung kleiner als 10^{-6} ist. Am Anfang der Schleife wird wieder `x` durch `xnew` ersetzt.

3.5 Hauptprogramm

Im Hauptprogramm werden die als Übergabeparameter benötigten Funktionen mit `EXTERNAL` deklariert und anschließend obige `SUBROUTINE`s für die Nullstellen von f bzw. g mit geeigneten Anfangsbedingungen aufgerufen.

4 Fortran-Code

```
      REAL FUNCTION f(x)
      REAL x
      f = x**3 - 5*x**2 + 100*x - 400
      END
c
      REAL FUNCTION f_(x)  ! derivative of FUNCTION f(x)
      REAL x
      f_ = 3*x**2 - 10*x + 100
      END
c
      REAL FUNCTION fixptf(x)  ! f(x) = 0  <=>  x = fixptf(x)
      REAL x
      fixptf = 400/(x**2 - 5*x + 100)
      END
c
c
      REAL FUNCTION g(x)
      REAL x
      IF(1.0+x .NE. 0.0) THEN
        g = exp(-x) - 0.1/(1.0+x)
      ELSE
        WRITE(*,*) "ERROR: g(-1) is undefined"
        STOP
      ENDIF
      END
c
      REAL FUNCTION g_(x)  ! derivative of FUNCTION g(x)
      REAL x
      IF(1.0+x .NE. 0.0) THEN
        g_ = -exp(-x) + 0.1/(1.0+x)**2
      ELSE
        WRITE(*,*) "ERROR: g_(-1) is undefined"
        STOP
      ENDIF
      END
c
      REAL FUNCTION fptg1(x)  ! g(x) = 0  <=>  x = fptg1(x)
      REAL x
      fptg1 = 0.1*EXP(x)-1
      END
c
      REAL FUNCTION fptg2(x)  ! g(x) = 0  <=>  x = fptg2(x)
      REAL x
      fptg2 = LOG((1.0+x)/0.1)
      END
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      SUBROUTINE Bisect(y, x10, x20)
      REAL y, x10, x20, x1, x2, x, prod
      x1 = x10
      x2 = x20
      WRITE(*,'(2A12)') 'x1', 'x2'
      WRITE(*,'(2F12.6)') x1, x2
      IF(y(x1)*y(x2) .GT. 0.0) THEN
c       ! y(x1) and y(x2) have the same sign
        WRITE(*,*) "ERROR: bad choice of x1 and x2"
        RETURN
      ENDIF
c     If y is continuous, there has to be a zero between x1 and x2.
10    x = 0.5*(x1+x2)
      IF(y(x1)*y(x) .LE. 0.0) THEN
c       ! sign change between x1 and x
        x2 = x
      ELSE
c       ! no sign change between x1 and x => zero between x and x2
        x1 = x
      ENDIF
      WRITE(*,'(2F12.6)') x1, x2
      IF(ABS(x1-x2) .GT. 1e-6) GOTO 10
      END
c
      SUBROUTINE Tangnt(y, y_, x0)
      REAL y, y_, x0, x, xnew, deriv
      WRITE(*,'(A12)') 'x'
      WRITE(*,'(F12.6)') x0
```

```
      xnew = x0
20    x = xnew
      deriv = y_(x)
      IF(deriv .NE. 0.0) THEN
        xnew = x - y(x)/deriv
      ELSE
        WRITE(*,*) "ERROR: division by zero"
        RETURN
      ENDIF
      WRITE(*,'(F12.6)') xnew
      IF( ABS(x-xnew) .GT. 1e-6) GOTO 20
      END
c
      SUBROUTINE RegFal(y, x1, x2)  ! Regula Falsi
      REAL y, x1, x2, xold, x, xnew, deriv
      x = x1
      xnew = x2
      WRITE(*,'(A12)') 'x'
      WRITE(*,'(F12.6)') x
      WRITE(*,'(F12.6)') xnew
30    xold = x
      x = xnew
      deriv = (y(x)-y(xold))/(x-xold)
      IF(deriv .NE. 0.0) THEN
        xnew = x - y(x)/deriv
      ELSE
        WRITE(*,*) "ERROR: division by zero"
        RETURN
      ENDIF
      WRITE(*,'(F12.6)') xnew
      IF( ABS(x-xnew) .GT. 1e-6) GOTO 30
      END
c
      SUBROUTINE FixPt(fixpty, x0)  ! fixed point
      REAL x0, x, xnew
      WRITE(*,'(A12)') 'x'
      WRITE(*,'(F12.6)') x0
      xnew = x0
40    x = xnew
      xnew = fixpty(x)
      WRITE(*,'(F12.6)') xnew
      IF( ABS(x-xnew) .GT. 1e-6) GOTO 40
      END
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      PROGRAM Zeros
c
      EXTERNAL f, f_, fixptf, g, g_, fptg1, fptg2
c
c     ! f(4.1467) = 0
      CALL Bisect(f, 0.0, 10.0)
      CALL Tangnt(f, f_, 0.0)
      CALL RegFal(f, 0.0, 10.0)
      CALL FixPt(fixptf, 0.0)
c
c     ! g(-0.9618) = 0
      CALL Bisect(g, -0.999, 0.0)
      CALL Tangnt(g, g_, -0.999)
      CALL RegFal(g, -0.97, -0.96)
      CALL FixPt(fptg1, -0.999)  ! use fptg1
c
c     ! g(3.8897) = 0
      CALL Bisect(g, 0.0, 10.0)
      CALL Tangnt(g, g_, 0.0)
      CALL RegFal(g, 0.0, 5.0)  ! use fptg2
      CALL FixPt(fptg2, 10.0)
      END
```

5 Ergebnisse

Die einzige Nullstelle von f liegt bei $x = 4.146723$.
g besitzt zwei Nullstellen: bei $x = -0.961778$ und $x = 3.889720$.

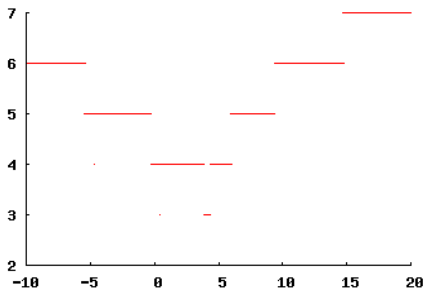
5.1 Bisektions-Verfahren

Das Bisektions-Verfahren konvergiert immer gegen die Nullstelle, wenn zwei Startwerte vorgegeben sind, die die Nullstelle der stetigen Funktion einschließen. Da die Länge des Intervalls pro Iterationsschritt halbiert wird, ist die Konvergenzgeschwindigkeit für gleiche Anfangslänge des Intervalls immer gleich: $|x_{1,n} - x_{2,n}| = \frac{|x_{1,0}-x_{2,0}|}{2^n} \overset{!}{<} 10^{-6}$

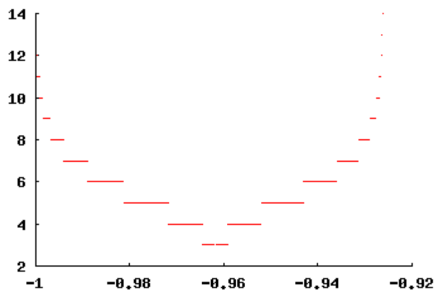
5.2 Newton-/Tangenten-Verfahren

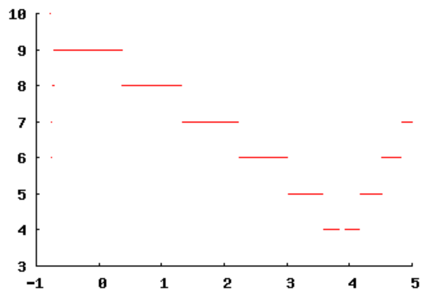
Das Newton-Verfahren scheint für f immer gegen die Nullstelle zu konvergieren, da f keine Extrema aufweist, so daß die Tangente in eine "falsche" Richtung zeigen könnte.

Die Abbildung zeigt die Zahl der benötigten Iterationen bei der Nullstellensuche von f mit dem Tangenten-Verfahren in Abhängigkeit vom Startwert.



Bei g ist der Bereich um die erste Nullstelle wegen dem Pol bei $x = -1$ problematisch. Das Newton-Verfahren konvergiert nur für Startwerte $-1 < x_0 < -0.927$ gegen $x = -0.961778$. Die zweite Nullstelle $x = 3.889720$ hingegen kann – wegen der Asymptotik von g – mit beliebigen Startwerten nach dem Maximum bei $x \approx -0.769$ erreicht werden.





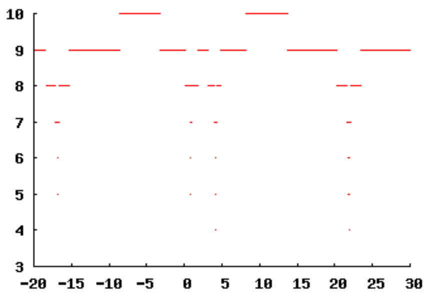
In den beiden Abbildungen werden wieder die benötigten Iterationen, für die Konvergenz zur jeweiligen Nullstelle von g mit dem Tangenten-Verfahren dargestellt.

5.3 Regula Falsi / Sekanten-Verfahren

Wegen der strengen Monotonie von f konvergiert das Sekanten-Verfahren für f in allen betrachteten Fällen gegen die Nullstelle. Es ist wesentlich schneller als das Bisektions-Verfahren, obwohl es keine zusätzlichen Berechnungen erfordert.
Für die Nullstellen von g ist die Situation schwieriger, weil wegen des Pols Falschabschätzungen der angemessenen Steigung auftreten können. Bei hinreichend gutem Einschluß der Nullstelle ist die Konvergenz jedoch auch hier gut und schneller als beim Bisektionsverfahren.

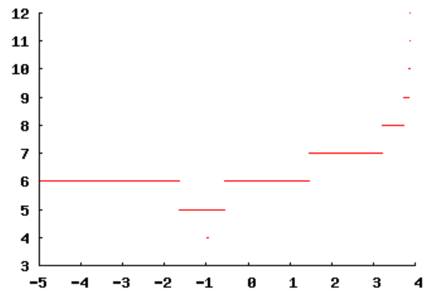
5.4 Fixpunkt-Verfahren

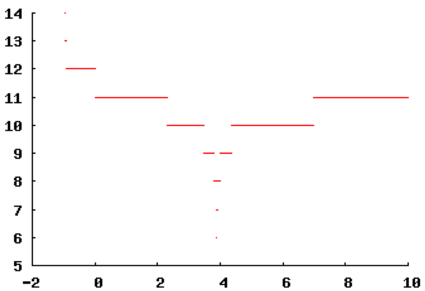
Das Fixpunktverfahren für f ist in der Form $x = \frac{400}{x^2-5x+100}$ recht stabil, da der Nenner nie verschwindet und zu große Anfangswerte nach der ersten Iteration in die Nähe von $x = 0$ gebracht werden.
Die Abbildung zeigt wieder die Anzahl der benötigten Iteration in Abhänigkeit vom Startwert. Besonders interessant sind die Resonanzstrukturen um $x = -16.8658$ und $x = 21.8652$, die zu einer anderen Struktur um $x = 0.8536$ führen, und diese führt direkt zur unmittelbaren Resonanz um die Nullstelle.



Zahl der Iterationen bei f (`FixPt`)

Das Fixpunktverfahren für $g(x) = 0$ in der Form $x = 0.1 \cdot e^x - 1$ konvergiert für Startwerte $x_0 < 3.889720$ (unterhalb der zweiten Nullstelle) gegen die erste Nullstelle $x = -0.961778$, für $x_0 > 3.889720$ divergiert das Verfahren.
Umgekehrt konvergiert die zweite Formulierung $x = \ln(\frac{1+x}{0.1})$ nur für Startwerte nach der ersten Nullstelle, und dann immer gegen die zweite, ansonsten wird das Argument des Logarithmus' negativ.





Diese beiden Graphen zeigen die Zahl der benötigten Iterationen beim Fixpunktverfahren für g mit den beiden angegebenen Varianten $x = 0.1 \cdot e^x - 1$ bzw. $x = \ln(\frac{1+x}{0.1})$.

5.5 Schlußfolgerung

Welches Verfahren zur Nullstellenbestimmung optimal ist, hängt stark von der Situation ab.

| Mh