

Numerik Starschnitt Nichtlineare GS

Die Lösung ergibt sich als Grenzwert einer Folge von Näherungen. Für die Brauchbarkeit und Effizienz eines Verfahrens ist das Konvergenzverhalten der Näherungsfolge gegen die Lösung entscheidend.

Fixpunktiteration $x^{(k+1)} = F(x^{(k)})$ $k=0,1,2,\dots$ z.B. $x = e^x$ auf $[0.5, 0.6]$

Die Fixpunktiteration (auch Methode der sukzessiven Approximation genannt) kann als Folge der $x^{(k)}$ reelle Zahlen, Vektoren oder auch Funktionen haben. $F(x)$ ist eine Abb. der betreffenden Menge in sich selbst. Ziel ist es die Fixpunktgleichung $x = F(x)$ zu lösen. Die Lösung x ist der Fixpunkt der Abbildung $F(x)$.

Banachscher Fixpunktsatz
Sei A eine abgeschlossene Teilmenge eines Banach-Raumes B und F eine kontrahierende Abb. von A in A . Dann gilt:
a) F besitzt genau einen Fixpunkt s in A .
b) Für jeden Startwert $x^{(0)}$ konvergiert $x^{(k)}$ gegen s .
c) Fehlerabschätzung:
 $\|x^{(k)} - s\| \leq L^k \|x^{(0)} - s\|$ für $0 \leq k < \infty$
d) $\|x^{(k)} - x^{(k-1)}\| \leq L^k \|x^{(0)} - x^{(1)}\|$ für $0 \leq k < \infty$
e) $\|x^{(k)} - x^{(k-1)}\| \leq L^k \|x^{(0)} - x^{(1)}\|$ für $0 \leq k < \infty$
f) positive Fehlerabschätzung für $k < \infty$

eine stetige nichtlineare Funktion $f(x)=0$ z.B. $\cos(x)$ $\cosh(x)+1=0$, börsartig, weil transzendent

Mit Hilfe des **Fehlers** $e^{(k)} = x^{(k)} - s$ und des Mittelwertsatzes der Differentialrechnung (bzw. Taylorreihenentwicklung) ergibt sich für F kontrahierend und p -mal stetig diff'bar
 $\lim_{k \rightarrow \infty} |e^{(k+1)}| \leq |F'(s)|^p |e^{(k)}|$

$|e^{(k)}|$ ist für $p=1$ linear konvergent, für $p=2$ quadratisch, ..., p bestimmt also die **Konvergenzordnung**.
 $q = |e^{(k+1)}|/|e^{(k)}|^p$ heißt **Konvergenzquotient**.
 $K = 1/p \cdot |F'(s)|$ heißt die **asymptotische Fehlerkonstante**.
Aus hoher Konvergenzordnung p und kleiner Fehlerkonstante K folgt sehr rasche Konvergenz.

Intervallschachtelung
Man beginnt mit dem Intervall $I = [a, b]$, so dass ein Vorzeichenwechsel existiert (Stetigkeit!). Man halbiert das Intervall und fährt mit dem entsprechenden Abschnitt fort. Bis s im Inneren eines hinreichend kleinen Intervalls liegt. Konvergenzordnung $p=1$. Fehlerkonstante $K = 2^{-p} \cdot 2^{-1}$ (Intervalllänge).

Regula falsi
Mit Hilfe der Funktionswerte der Intervallgrenzen $x^{(0)}$ und $x^{(1)}$ wird eine Gerade (lineare Interpolierende) bestimmt. Ihr Schnittpunkt mit der x -Achse $x^{(2)}$ resultiert aus der Nullstelle der Geradengleichung. Das Vorzeichen entscheidet, welches Intervall weiterverwendet wird (Vorzeichenwechsel). ... Man trifft die gesuchte Nullstelle aus numerischen Gründen nie exakt. Konvergenzordnung $p=1$.

Gauss'sches Iterationsverfahren
Iterationsvorschrift $x^{(k+1)} = x^{(k-1)} - y_{k-1} \cdot (x^{(k)} - x^{(k-1)}) / (y_k - y_{k-1})$
Beweis: Aus der Geradengleichung $y = mx + b$ der Sekante folgt durch Einsetzen eines bekannten Punktes b und m aus dem Steigungsdreieck zweier Punkte. Anschließend die Nullstelle einsetzen mit $y=0$ und $x=x^{(k-1)}$ und entsprechend auflösen.

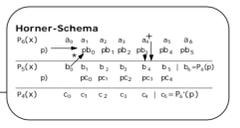
Sekantenmethode
Modifikation von Regula falsi, allerdings brauchen $x^{(0)}$ und $x^{(1)}$ die Lösung nicht einzuschließen, denn ein Vorzeichenwechsel wird nicht mehr verlangt. Wieder ist $x^{(2)}$ die Stelle des Schnittpunktes der linear **Interpolierenden** mit der x -Achse. Mit $x^{(2)}$ und der nächst näheren Stelle wird weitergerechnet. Konvergenzordnung $p=1,618$.
Problem: Geeignete Startwerte. Diese Methode gehört nicht mehr zu den Fixpunktiterationen, weil jeweils zwei Werte für die Berechnung verwendet werden (man nennt es deshalb **zweistufiges** Iterationsverfahren).

Newtonverfahren
Voraussetzung: f stetig differenzierbar und die Ableitung ohne großen Aufwand berechenbar. Wie Sekantenmethode, allerdings wird hier der Schnittpunkt der Tangente mit der x -Achse berechnet, um anschließend an dieser Stelle fortzufahren. Iterationsvorschrift:
 $x^{(k+1)} = x^{(k)} - f(x^{(k)}) / f'(x^{(k)})$
Konvergenzordnung $p=2$.
Trotz der quadratischen Konvergenz hat das Verfahren den Nachteil, daß neben den Funktionswerten immer eine Ableitung berechnet werden muß. Mit demselben Aufwand können zwei Schritte der Sekantenmethode durchgeführt werden. Die Konvergenzordnung für einen **Doppelschritt** der Sekantenmethode ist $2p_k = p+1 = 2,618$, also größer. Reelle Startwerte konvergieren nur gegen reelle Nullstellen.

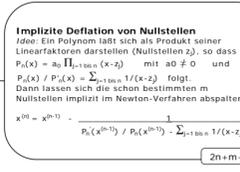
Verfahren von Muller
Statt eines Interpolationspolynoms 1. Grades wird ein Polynom 2. Grades (quadratisch Interpolierende) verwendet: $P_2(x) = A(x-x^{(k)})^2 + B(x-x^{(k-1)})(x-x^{(k)}) + C = 0$, um aus seiner Nullstelle $x^{(k+1)}$, die am nächsten zu $x^{(k)}$ liegt, eine neue Näherung zu ermitteln. Die Koeffizienten ergeben sich aus den $x^{(k-2)}, x^{(k-1)}, x^{(k)}$ und ihren Funktionswerten. Es handelt sich also um ein **dreistufiges Interpolationsverfahren**.
Iterationsvorschrift: $x^{(k+1)} = F(x^{(k-2)}, x^{(k-1)}, x^{(k)})$
Konvergenzordnung: $p=1,839$
Reelle Startwerte konvergieren auch gegen komplexe Nullstellen.

Nullstellen von Polynomen $P_n(x)=0$ $P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \neq 0$

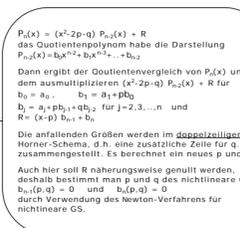
Divisionsalgorithmus
Idee: Division mit Rest durch den Linearfaktor $(x-p)$ führt zu $P_n(x) = (x-p)P_{n-1}(x) + R$ das Quotientenpolynom habe die Darstellung $P_{n-1}(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \dots + b_1x + b_0$.
Dann ergibt der Divisionsalgorithmus von $P_n(x)$ und dem ausmultiplizieren $(x-p)P_{n-1}(x)$ die Restglieder $R = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 - (x-p)(b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \dots + b_1x + b_0)$.
Wir verwenden das Newton-Verfahren um die Nullstelle von P_n zu approximieren. Startwert $x^{(0)}$.
 $x^{(1)} = x^{(0)} - P_n(x^{(0)}) / P_n'(x^{(0)})$ mit $x^{(0)} = p$.
wobei $P_n(p) = b_n$ und $P_n'(p) = P_{n-1}(p)$ ist.
Die anfallenden Größen werden im Horner-Schema zusammengestellt. Es berechnet ein neues p , so dass die mehrmalige Durchführung $P_n(p)$ ungefähr 0 ist.
(Es lassen sich also mit dem Divisionsalgo. auch höhere Ableitungen berechnen.)



Sukzessive Deflation von Nullstellen
Durch den Divisionsalgo. können nach und nach die Nullstellen berechnet werden, jeweils durch Anwendung des Algo. und anschließender Abspaltung des Linearfaktors durch Division.
Diese Vorgehensweise kann problematisch sein, weil die berechneten Nullstellen nur Näherungen sind und somit sich die Fehler anhäufen, so dass die zuletzt berechneten Nullstellen sehr schief getrennt werden. Eine Empfindlichkeitsanalyse ist deshalb angebracht.



Das Newton-Verfahren ist auch durchführbar für Polynome mit komplexen Koeffizienten oder trotz reeller Koeffizienten mit komplexen Nullstellen. Bei komplexen Nullstellen mußte allerdings mit komplexen Startwerten gerechnet werden.
Wir nutzen die Tatsache, das **komplexe Nullstellen von reellen Polynomen paarweise konjugiert** sind, deshalb dividiert man durch einen quadratischen Teiler, dem Produkt der konjugierten Nullstellen, der wieder reell ist:
 $(x-u-iv)(x-u+iv) = x^2 - 2ux + (u^2 + v^2)$
Das Verfahren zur Bestimmung des quadratischen Teilers eines Polynoms mit reellen Koeffizienten nennt man die **Methode von Balzrow**.



Lipschitz-stetig
Eine abgeschlossene Teilmenge A aus dem Banach-Raum B , die F von A in A abbildet, heißt Lipschitz-stetig auf A mit der Lipschitzkonstante $0 < L < \infty$, wenn gilt
 $\|F(x) - F(y)\| \leq L \|x - y\|$ für alle x, y aus A
mit der Norm $\| \cdot \|$ ($\| \cdot \| = \max_i |x_i|$)
 F heißt **kontrahierend** auf A , wenn $L < 1$ gilt.
Praktische Bestimmung von L durch $L = \max_{x \in A} \|F'(x)\|$

Banach-Raum
Allen folgenden Betrachtungen wird ein Banach-Raum B zugrunde gelegt, ein reeller/komplexer Vektorraum auf dem eine Norm definiert ist, s. d. in ihm jede **Cauchy-Folge** konvergiert und ihr Grenzwert in ihm liegt.

Cauchy-Folge
Jede konvergente Folge reeller Zahlen ist eine C.-Folge. Unter einer C.-Folge $\{x_n\}_{n \in \mathbb{N}}$ versteht man: Für alle $\epsilon > 0$ existiert ein $N \in \mathbb{N}$, s. d. $|x_n - x_m| < \epsilon$ für alle $n, m \geq N$, d. h. die Folgeglieder weichen beliebig wenig voneinander ab, wenn der Index nur genügend groß ist.

Aitkens Δ^2 -Prozeß
Aus einer linear konvergenten Folge kann eine schneller konvergierende Folge konstruiert werden:
 $e^{(k+1)} = e^{(k-2)} e^{(k)}$ wird nach s aufgelöst (und in $z^{(k)}$ umbenannt).

Fall: **System von nichtlinearen Funktionen**
 $F(x) = \begin{pmatrix} f(x,y) \\ g(x,y) \end{pmatrix} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$
zweier Veränderlicher $x = (x,y) \in \mathbb{R}^2$
Für den Fixpunkt $s = (s,t) \in \mathbb{R}^2$
 $s = F(s,t)$ und $t = G(s,t)$

Fall: **System von Funktionen mehrerer Veränderlicher**
Für irgendeine Matrixnorm, z.B. Spektralradius, muß für die Jakobimatrix gelten
 $\|J(x,y)\| = \|F'(x,y)\| < 1$
damit F **kontrahierend** ist.
Der Fehlervektor wird definiert als
 $e^{(k)} = x^{(k)} - s = (e^{(k)}_1, e^{(k)}_2)^T \in \mathbb{R}^2$

Konvergenzordnung $p=1$, wenn
 $J(s,t) \neq 0$
mit $e^{(k+1)} = J(s,t) e^{(k)}$
Konvergenzordnung $p=2$, wenn
 $J(s,t) = 0$ und $H_f \neq 0$ oder $H_g \neq 0$

System stetiger nichtlinearer Gleichungen $F(x)=0$ $f(x) = \begin{pmatrix} f(x,y) \\ g(x,y) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = 0$

Newton-Verfahren für nichtlineare GS
Man arbeitet im k -ten Schritt mit dem Korrekturansatz $x = x^{(k)} + \epsilon^{(k)}$ und $t = y^{(k)} + \eta^{(k)}$, mit welchem das System linearisiert wird.
 $\Phi(x^{(k)}, y^{(k)}) \epsilon^{(k)} + \Gamma(x^{(k)}, y^{(k)}) \eta^{(k)} = 0$ mit der Korrektur $e^{(k)} = (e^{(k)}_1, e^{(k)}_2)^T$ und regulären Funktionalmatrix Φ . Der erste Summand ist die Linearisierung. Der Gauß-Algorithmus löst das entstandene Gleichungssystem und bestimmt somit $e^{(k)}$.
Die Korrektur liefert allerdings nicht die Lösung, sondern eine erneute Näherung $x^{(k+1)} = x^{(k)} + e^{(k)}$ mit der fortgefahren wird. Somit lautet die Iterationsvorschrift:
 $f(x^{(k+1)}) = x^{(k+1)} - \Phi^{-1}(x^{(k)}) f(x^{(k)})$
Konvergenzordnung mindestens $p=2$, wenn f dreimal stetig diff'bar ist.

Vereinfachtes Newtonverfahren
Da sich die 1. Ableitung in den letzten Iterationsschritten kaum ändert, wird sie einmal für einen guten Startwert $x^{(0)}$ berechnet und dann beibehalten.
Iterationsvorschrift: $x^{(k+1)} = x^{(k)} - f(x^{(0)}) / f'(x^{(0)})$
Konvergenzordnung nur $p=1$, allerdings ist die Fehlerkonstante oft nur sehr klein.

Die Berechnung der Funktionalmatrix ist relativ aufwendig (n^2 Ableitungen), deshalb berechnet man $\Phi(x^{(k)}, y^{(k)})$ nur einmal für einen guten Startvektor, die Korrekturen werden dann aus $\Phi(x^{(0)}, y^{(0)}) \epsilon^{(k)} + \Gamma(x^{(k)}, y^{(k)}) \eta^{(k)} = 0$ berechnet.
Weiter muss die LR-Zerlegung des Gauss-Algo. nur einmal erfolgen.
Konvergenzordnung $p=1$

Nichtlineares Einzelschrittverfahren
(= nichtlineare sukzessive Relaxation)
Die Berechnung von Φ wird größtenteils vermieden, indem die Lösung eines Systems von n nichtlinearen Gleichungen in n Unbekannten auf die sukzessive Lösung von nichtlinearen Gleichungen einer Unbekannten zurückgeführt wird.
Voraussetzung: Die i -te Gleichung enthält die i -te Unbekannte und die partielle Ableitung nach ihr ist ungleich null.
Dann soll die i -te Komponente des $(k+1)$ -ten Näherungsvektors $x^{(k+1)}$ als Lösung der i -te Gleichung $f_i(x_1^{(k)}, \dots, x_{i-1}^{(k)}, x_{i+1}^{(k)}, \dots, x_n^{(k)}) = 0$ bestimmt werden. Somit ist nur $x_i^{(k+1)}$ unbekannt und wird (mit dem Newton-Verfahren) berechnet durch
 $f_i(x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x_{i+1}^{(k)}, \dots, x_n^{(k)})$
 $x_i^{(k+1)} = x_i^{(k)} - \frac{d f_i(\dots) / d x_i}{d f_i(\dots) / d x_i}$
Für diese **innere Iteration** werden nur die partielle Ableitung $d f_i(\dots) / d x_i$ benötigt, also n .

Newton'sche Einzelschrittverfahren
Modifikation des Nichtlinearen Einzelschrittverfahrens. Die i -te Gleichung braucht nicht exakt nach $x_i^{(k+1)}$ aufgelöst werden, da dieser Wert ohnehin nur eine Näherung darstellt. Deshalb verzichtet man auf die innere Iteration und führt nur einen **einzigsten Schritt** der Newton-Korrektur aus.
Konvergenzordnung $p=1$

SOR-Newtonverfahren
(= **successive overrelaxation**)
Die lineare Konvergenz des Newton'schen Einzelschrittverfahrens verbessert sich durch Multiplikation der Korrektur der i -ten Komponente mit einem konstanten **Relaxationsfaktor** $\omega \in]0, 2[$
 $f(\dots)$
 $x_i^{(k+1)} = x_i^{(k)} - \omega \frac{d f_i(\dots) / d x_i}{d f_i(\dots) / d x_i}$
Besonders für den Fall geeignet, dass die i -te Gleichung nur wenige der Unbekannten miteinander verknüpft.