

Wenn man die Gleichungen zusammenfaßt, erhält man die Matrix-Form

$$\begin{pmatrix} s_{00} & s_{01} & s_{02} & \cdots & s_{0n} \\ s_{10} & s_{11} & s_{12} & \cdots & s_{1n} \\ s_{20} & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ s_{n0} & s_{n1} & \cdots & \cdots & s_{nn} \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} \tag{2.60}$$

wobei nun

$$s_{ij} = \sum_{k=1}^m w_k \phi_i(x_k) \phi_j(x_k) \quad \text{und} \quad c_j = \sum_{k=1}^m w_k \phi_j(x_k) f_k. \tag{2.61}$$

Da hier die Querdiagonalen nicht notwendigerweise konstant sind, ist die Matrix i.a. keine Hankel-Matrix, sondern nur noch eine sogenannte *Gramsche Matrix*. Damit (2.60) eindeutig lösbar ist, muß man bestimmte Forderungen an das System von Ansatzfunktionen $\{\phi_i(x)\}$ stellen.⁵

Die Matrizen in den Normalgleichungen (2.53) und (2.60) können in der Regel für kleine Werte von n gut gelöst werden. Wenn n jedoch zu groß wird, kann es sein, daß die Matrix schlecht konditioniert ist (siehe Kap. 5.1.4 und (5.38)), was zu numerischen Problemen führen kann. Eine Möglichkeit der Abhilfe besteht darin, die Approximation durch ein System von Polynomen $\{q_k(x)\}$ vorzunehmen, wobei man die Anzahl der Polynome sukzessive erhöht (mit aufsteigender Ordnung der Polynome), bis man eine hinreichend gute Approximation erhält. Diese Prozedur wird numerisch sehr vereinfacht, wenn man *orthogonale Polynome* verwendet. Wie man diese konstruieren kann, ist z.B. in Golub & Ortega (1996) beschrieben.⁶



Jorgen Pedersen Gram
1850–1916

2.5 Bestimmung von Nullstellen

Die Bestimmung von Nullstellen ist ein Problem, daß sehr häufig auftritt. Schon das anscheinend einfache mechanische Problem, die Gleichgewichtslage eines Systems aus zwei elastischen Stäben unter einer Last (Abb. 2.12) (Zweischlag) zu berechnen, führt auf nichtlineare Gleichungen, die nicht mehr in geschlossener Form gelöst werden können. Auch die Navier-Stokes-Gleichung der Strömungsmechanik ist auf-

⁵Man kann (2.60) auch wieder in der Form (2.55) schreiben mit entsprechend modifizierten Matrizen E und W . Jedoch ist $E^T \cdot W \cdot E$ nicht mehr positiv definit. Für eine eindeutige Lösung muß man solche Ansatzfunktionen (und Punkte x_i) wählen, daß positiv definit ist.

⁶Polynome q_0, \dots, q_n vom Grade $0, \dots, n$ heißen *orthogonal* bezüglich der Punkte x_1, \dots, x_m , wenn gilt

$$\sum_{i=1}^m q_k(x_i) q_j(x_i) = 0, \quad \text{für } k, j = 0, 1, \dots, n, k \neq j.$$

Man kann sie sukzessive aus den Daten konstruieren.

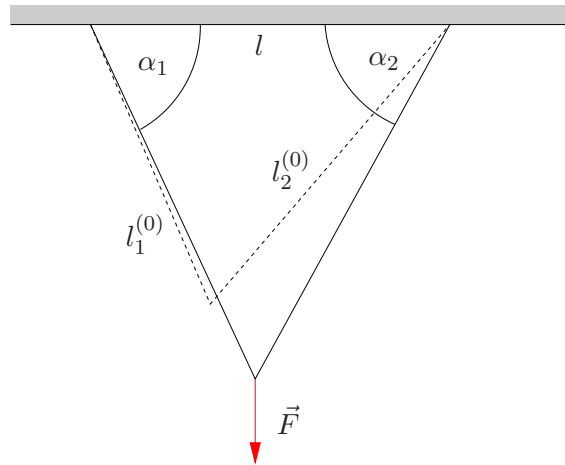


Abbildung 2.12: Gleichgewicht zweier elastischer Stäbe unterschiedlicher Elastizität bei Belastung.

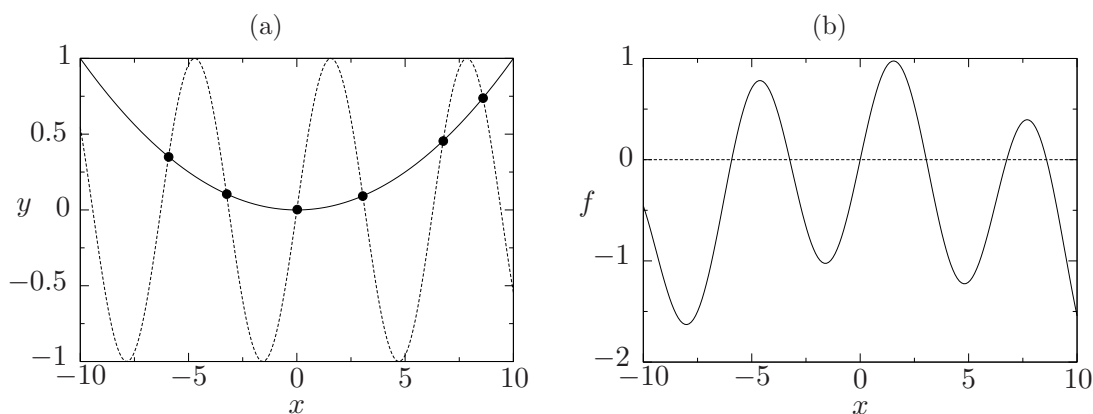


Abbildung 2.13: (a) Graphische Lösung von (2.62) für $\lambda = 0.01$. (b) Entsprechendes Nullstellenproblem.

grund des konvektiven Terms $\vec{u} \cdot \nabla \vec{u}$ nichtlinear. Zur Berechnung von Strömungen muß man daher im allgemeinen ein großes System nichtlinearer Gleichungen für die unbekanntes Geschwindigkeitskomponenten an allen Punkten eines feinen Gitters lösen. Ein einfaches algebraisches Problem könnte lauten: Berechnen Sie die Lösung(en) von

$$\lambda x^2 = \sin(x). \quad (2.62)$$

Die graphische Lösung ist in Abb. 2.13a dargestellt.

2.5.1 Newton-Verfahren

Eine nichtlineare algebraische Gleichung wie (2.62) kann man immer in der Form eines *Nullstellenproblems* schreiben (Abb. 2.13b)

$$f(x) = \sin(x) - \lambda x^2 = 0. \quad (2.63)$$

Wäre $f(x)$ ein Polynom vom Grade n ist, dann hätte f genau n reelle oder komplexe Nullstellen (Wurzeln). Für eine allgemeine Funktion läßt sich nicht *a priori* sagen,

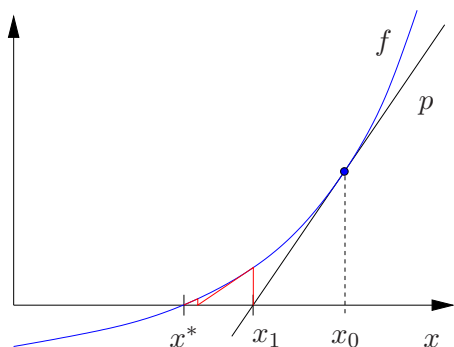


Abbildung 2.14: Lineare Näherung der Funktion f in der Nähe der Nullstelle x^* durch die Tangente p an f bei x_0 . Die roten Linien verdeutlichen die Abfolge der Newton-Iteration anschaulich.

wieviele Nullstellen sie in einem bestimmten Intervall besitzt. Wenn eine Funktion in einem Intervall aber streng monoton ist, dann besitzt sie maximal eine Nullstelle. Gilt für eine stetige Funktion

$$f(a) < 0, \quad \text{und} \quad f(b) > 0, \tag{2.64}$$

dann besitzt $f(x)$ im Intervall $[a, b]$ *mindestens eine* Nullstelle. Um eine Nullstelle x^* numerisch zu bestimmen, wird meistens der Funktionsverlauf zwischen den Punkten a und b durch ein Polynom $p(x)$ niedriger Ordnung approximiert und dessen Nullstelle näherungsweise als Nullstelle von $f(x)$ aufgefaßt.

Sei x_0 ein Punkt in der Nähe der Nullstelle. Für die Taylor-Entwicklung von f um den Punkt x_0 gilt

$$f(x) = \underbrace{f(x_0) + (x - x_0)f'(x_0)}_{:=p(x)} + \frac{1}{2}f''(z)(x - x_0)^2. \tag{2.65}$$

Die Taylor-Entwicklung bis zur ersten Ordnung können wir als Polynomapproximation erster Ordnung auffassen. Indem wir f durch p ersetzen, haben wir f linearisiert. Dabei wird die Funktion f lokal durch die Tangente im Punkt x_0 ersetzt (Abb. 2.14). Aus (2.65) ergibt sich die Approximation x_1 der Nullstelle von f durch $p(x_1) = 0$ aus

$$p(x_1) = f(x_0) + (x_1 - x_0)f'(x_0) = 0, \tag{2.66}$$

also

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}. \tag{2.67}$$

Wenn man nun vom Punkt x_1 ausgeht und auf dieselbe Art eine zweite Näherung x_2 berechnet, und so weiter, dann erhält man die *Newton-Iteration*

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, \dots \tag{2.68}$$

Diese Iterationsvorschrift wird auch *Newton-Raphson-Verfahren*⁷ genannt.

⁷Joseph Raphson, 1648–1715: Raphson war einer der wenigen, denen es Isaac Newton erlaubte,

Tabelle 2.1: Beispiel für die Newton-Iteration von (2.63) mit $\lambda = 0.01$ und Startwert $x_1 = 4$. Man erkennt die quadratische Konvergenz.

n	x_n
1	4
2	2.750343532969441
3	3.062460099178964
4	3.048532919044707
5	3.048523403179332
6	3.048523403174493
7	3.048523403174493

Um die Konvergenz der Newton-Iteration zu untersuchen, betrachten wir den Fehler $x^* - x_{n+1}$, den wir beim $(n+1)$ -ten Iterationsschritt machen. Dazu verwenden wir das Lagrangesche Restglied und finden für $x = x^*$ aus (2.65) ($x_0 \rightarrow x_n$)

$$f(x^*) - p(x^*) = -f(x_n) - (x^* - x_n)f'(x_n) = \frac{1}{2}f''(z)(x^* - x_n)^2. \quad (2.69)$$

Mit Hilfe eines Newton-Schrittes (2.68) können wir $x_n f'(x_n) - f(x_n) = x_{n+1} f'(x_n)$ ersetzen und erhalten so

$$x_{n+1} f'(x_n) - x^* f'(x_n) = \frac{1}{2} f''(z) (x^* - x_n)^2, \quad (2.70)$$

und mit $f'(x_n) \neq 0$

$$x^* - x_{n+1} = \underbrace{-\frac{1}{2} \frac{f''(z)}{f'(x_n)}}_{c_n} (x^* - x_n)^2 = c_n (x^* - x_n)^2. \quad (2.71)$$

Ist nun in einem Intervall um die Nullstelle $|f''(x)| \leq M$ betragsmäßig beschränkt und $|f'(x)| \geq m > 0$, dann kann man $|c_n| \leq M/(2m) := c$ nach oben hin abschätzen. Aus der Beschränktheit des Koeffizienten folgt, daß sich der Fehler in jedem Schritt quadratisch verringert. Ist zum Beispiel der Fehler schon $|x^* - x_n| = 10^{-4}$, dann wird er im nächsten Schritt $< c \times 10^{-8}$ sein. Bei jeder Iteration verdoppelt sich ungefähr die Anzahl der korrekten Dezimalen der Approximation der Nullstelle (siehe Tab. 2.1). Typischerweise wählt man als Abbruchkriterium der Newton-Iteration die Bedingung $|x_{n+1} - x_n| < \epsilon$.

Die *quadratische Konvergenz* der Newton-Iteration ist so ziemlich das schnellste, was man erreichen kann. Die rapide Konvergenz hat ihren Preis, denn die Newton-Iteration konvergiert nur, wenn der Anfangswert x_0 schon hinreichend nahe bei der wirklichen Nullstelle x^* liegt. Dieses Verhalten wird auch als *lokale Konvergenz* bezeichnet. Für das Beispiel (2.63) ist die schnelle Konvergenz in Abb. 2.15 illustriert.

vorab seine mathematischen Werke zu sehen. 1690 veröffentlichte er sein Buch *Analysis aequationum universalis*, welches das Newton-Raphson-Verfahren zur numerischen Lösung von nichtlinearen Gleichungen enthält. Hierfür wurde er 1691 zum Mitglied der Royal Society ernannt. Außerdem schrieb er das Buch *History of Fluxions*, was aber erst nach seinem Tod erschien.

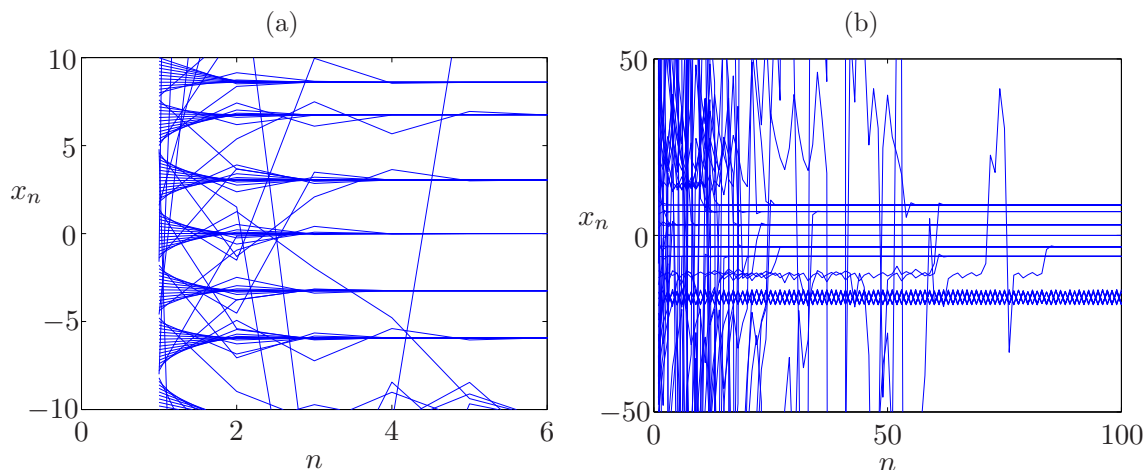


Abbildung 2.15: Newton-Iteration von (2.63) mit $\lambda = 0.01$ für äquidistanten Anfangswerte $x_1 \in [-10, 10]$ mit $\Delta x_1 = 0.2$ (a) und mit $x_1 \in [-50, 50]$ mit $\Delta x_1 = 1$ (b). Die Iteration konvergiert sehr schnell für die meisten Anfangsbedingungen nahe einer der sechs Nullstellen (a). Für andere Anfangsbedingungen sehr großen Schwankungen oder periodischem oder quasiperiodischem Verhalten (b).

Falls der Startwert x_0 nicht hinreichend nahe an x^* liegt, kann die Iteration auch divergieren oder in chaotischer Weise schwanken. Auch ist ein periodisches Verhalten der Iterierten möglich. In Abb. 2.16 ist dieses Verhalten für ein zweidimensionales System illustriert.

2.5.2 Sekantenmethode

Grundlage der Newton-Iteration war die Taylor-Entwicklung in der Nähe eines Punktes, der hinreichend dicht an der Nullstelle liegt. Anstelle der Taylor-Entwicklung kann man zur Konstruktion einer verbesserten Näherung der Nullstelle auch die Sekante verwenden, die durch die Punkte $f(x_0)$ und $f(x_1)$ definiert ist, wobei x_0 und x_1 die beiden vorhergehenden Iterierten sind (Abb. 2.17).

Die Gleichung für die Sekante mit Stützpunkten x_0 und x_1 lautet

$$s = f_1 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_1) \tag{2.72}$$

Für die Nullstelle x_2 der Sekante gilt damit

$$0 = f_1 + \frac{f_1 - f_0}{x_1 - x_0}(x_2 - x_1) \Rightarrow x_2 = x_1 - f_1 \frac{x_1 - x_0}{f_1 - f_0}. \tag{2.73}$$

Wenn man diese Methode systematisch fortsetzt, kommt man auf die Iteration (*Sekantenmethode*)

$$x_{n+1} = x_n - f_n \frac{x_n - x_{n-1}}{f_n - f_{n-1}}, \quad n = 1, 2, \dots \tag{2.74}$$

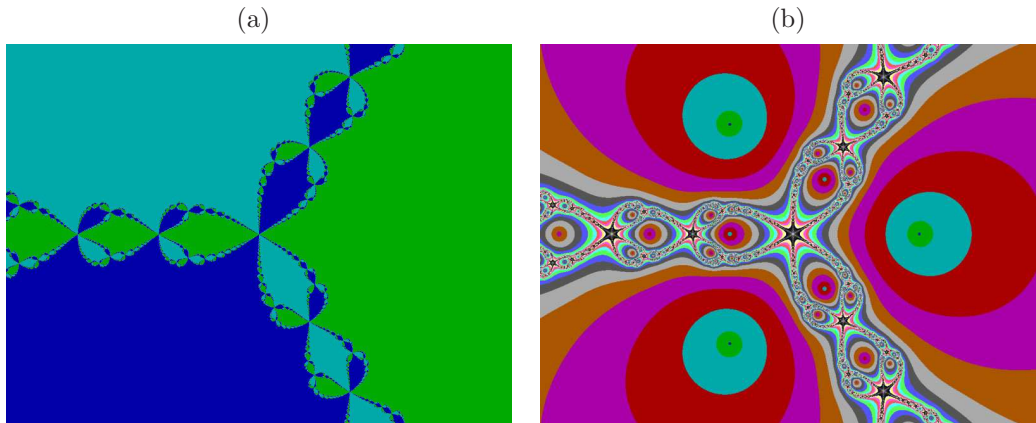
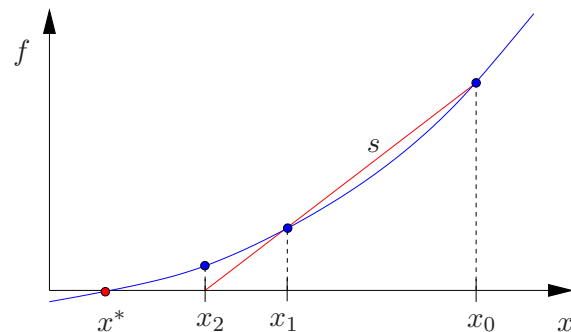


Abbildung 2.16: Newton-Fraktal für die Newton-Iteration der komplexen Funktion $p(z) = z^3 - 1$. Die Nullstellen sind die drei Einheitswurzeln z_i^* , $i = 1, 2, 3$, in der komplexen Ebene. Es sind dies $z_1^* = e^0$, $z_2^* = e^{2i\pi/3}$ und $z_3^* = e^{4i\pi/3}$. Gezeigt ist die komplexe Ebene der Startwerte z_0 . In (a) zeigt die Farbe an, zu welcher Nullstelle die Newton-Iteration konvergiert. Die Grenzen der Attraktionsbereiche der einzelnen Nullstellen sind fraktal. In (b) ist die Farbe ein Maß für die Anzahl der Iterationen, die erforderlich sind, um von einem Startwert z_0 eine bestimmte Genauigkeit $\|z_n - z_i^*\| < \epsilon$ zu erreichen. Die Abbildungen wurden mit dem Programm **FRACTINT** erstellt.

Abbildung 2.17: Approximation der Nullstelle von f durch die Nullstelle der Sekante s .



Die Sekantenmethode hat eine ähnliche Struktur wie die Newton-Methode. Der Unterschied besteht darin, daß bei der Sekantenmethode die Ableitung diskret gebildet wird, und zwar mit Hilfe der beiden vorhergehenden Iterationspunkte

$$f'_n \approx \frac{f_n - f_{n-1}}{x_n - x_{n-1}}. \quad (2.75)$$

Die Konvergenzrate der Sekanten-Methode ist im allgemeinen geringer als die der Newton-Methode. Man kann zeigen, daß im Limes $x_i \rightarrow x^*$ die Konvergenzordnung $(1 + \sqrt{5})/2 \approx 1.62$ beträgt, daß also gilt

$$|x^* - x_{i+1}| \sim c |x^* - x_i|^{1.62} \quad (2.76)$$

Dieser Nachteil wird aber unter Umständen dadurch kompensiert, daß man für

die Sekanten-Methode pro Iteration nur einen Funktionswert f berechnen muß, während man bei der Newton-Iteration sowohl f als auch f' bestimmen muß.

2.5.3 Bisektion und Regula Falsi

Die einfachste Möglichkeit, eine Nullstelle zu bestimmen, besteht in der Halbierung des Intervalls (*Bisektion*). Angenommen, in dem Intervall $[a, b]$ mit $f(a)f(b) < 0$ existiert nur eine Nullstelle von $f(x)$, dann berechnet man $f(x_1)$ an der Stelle $x_1 = (a + b)/2$. Das neue Intervall, in dem sich die Nullstelle befindet ist dann

$$\begin{aligned} [a, x_1], & \quad \text{falls } f(a)f(x_1) < 0, \\ [x_1, b], & \quad \text{falls } f(x_1)f(b) < 0. \end{aligned} \tag{2.77}$$

Diese Intervallhalbierung wird nun iteriert. Bei der Intervallhalbierung kann man den Fehler mit Hilfe der ursprünglichen Intervalllänge $b - a$ abschätzen. Nach m Bisektionen beträgt er

$$|x^* - x_m| \leq \frac{b - a}{2^m}. \tag{2.78}$$

Um den Fehler um 6 Zehnerpotenzen zu verringern ($2^m = 10^6$), benötigt man daher $m = 6 \log_2(10) \approx 20$ Iterationen. Die Konvergenz ist also sehr langsam.

Um eine Verbesserung der Konvergenzrate zu erreichen, kann man anstelle der Intervallhalbierung die neue Intervallgrenze x_1 auch mit Hilfe der Sekanten-Methode (2.74) berechnen. Das resultierende Verfahren heißt dann *regula falsi*.

2.5.4 Fehler und Kondition

Numerisch kann man die Funktion, deren Nullstelle zu bestimmen ist, nur mit endlicher Genauigkeit auswerten. Da die Funktionswerte in der Nähe der Nullstelle sehr klein werden, kann es deshalb sein, daß bei einer bestimmten Iteration der Bisektion das Vorzeichen von f falsch ist. Weitere Iterationen machen dann keinen Sinn mehr und die Fehlerabschätzung (2.78) verliert ihre Gültigkeit. Dieses Problem äußert sich dann in einem irregulären Verhalten der Iteration. Dies trifft auch für die Newton-Iteration und die Sekanten-Methode zu, denn auch bei diesen Verfahren muß die kleine Größe f in der Nähe der Nullstelle berechnet werden.

Das Problem ist offenbar umso größer, je flacher die Funktion f durch die Nullstelle geht. Am Beispiel einer linearen Funktion ist dieser Sachverhalt in Abb. 2.18 dargestellt. Ein bekanntes Beispiel sind die n entarteten Nullstellen $x_j = 0$ von $x^n = 0$. Falls nun die rechte Seite nicht Null ist, sondern nur ein wenig davon abweicht, erhält man

$$x^n = \epsilon, \tag{2.79}$$

mit den Lösungen

$$x_j = \epsilon^{1/n} \exp \left\{ 2\pi i \frac{j}{n} \right\}. \tag{2.80}$$

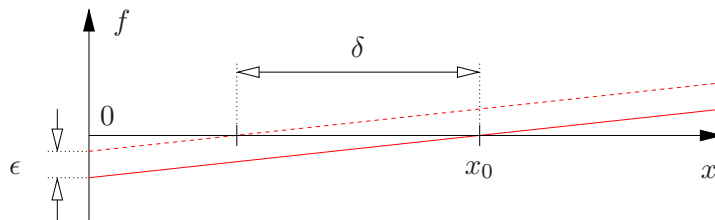


Abbildung 2.18: Ungenauigkeit der Nullstellenbestimmung bei schlechter Kondition: Bei geringer Steigung $f'(x_0)$ hat ein kleiner additiver Fehler ϵ eine große Auswirkung auf die Verschiebung δ der Nullstelle.

Falls $n = 10$ ist und $\epsilon = 10^{-10}$, dann sind die Wurzeln betragsmäßig gleich $\epsilon^{1/n} = (10^{-10})^{1/10} = 10^{-1}$. In diesem Fall hat eine Änderung der Eingangsdaten um 10^{-10} eine Änderung der Ausgangsdaten von 10^{-1} bewirkt. Der Fehler wurde demnach um einen Faktor 10^9 verstärkt!

Ganz allgemein bezeichnet man ein Problem, bei dem kleine Ungenauigkeiten in den Eingangsdaten (Parametern) zu sehr großen Ungenauigkeiten der Ausgangsdaten (Lösungen) führt, als schlecht konditioniert. Man sagt auch: die *Kondition eines Problems* ist schlecht. Diesen Sachverhalt kann man noch präzisieren und durch Zuweisung einer *Konditionszahl* (5.38) quantifizieren (Kap. 5.1.4).

2.5.5 Fixpunktiteration

Die Nullstelle einer Funktion wie (2.63) kann man auch mittels *Fixpunktiteration* finden. Dazu addiert man x auf beiden Seiten der Gleichung. In unserem Beispiel erhält man so

$$x = \sin(x) - \lambda x^2 + x = g(x). \quad (2.81)$$

Um zu einer Iteration zu kommen, wird die linke Seite bei $i + 1$ und die rechte bei i ausgewertet. So erhält man

$$x_{i+1} = g(x_i), \quad (2.82)$$

wobei in unserem Beispiel $g(x_i) = \sin(x_i) - \lambda x_i^2 + x_i$. Die beiden Seiten der Gleichung sind in Abb. 2.19 dargestellt. Die Schnittpunkte beider Graphen heißen *Fixpunkte*. Wenn wir einen Fixpunkt mit x^* bezeichnen, gilt offenbar $x_{i+1}^* = x_i^*$. Es gibt stabile und instabile Fixpunkte. Ob ein Fixpunkt *stabil* oder *instabil* ist, hängt davon ab, ob die Iteration eines Startwertes in der Nähe des Fixpunktes auf den Fixpunkt konvergiert, oder ob sie sich von ihm entfernt. Der linke Fixpunkt in Abb. 2.19b ist instabil, während der rechte stabil ist. Für einen stabilen Fixpunkt muß in einer Umgebung des Fixpunktes gelten $|g'(x)| < 1$. Allgemein gilt:⁸

⁸Die dritte Bedingung nennt man auch *Lipschitz-Bedingung* mit *Lipschitz-Konstante* K .

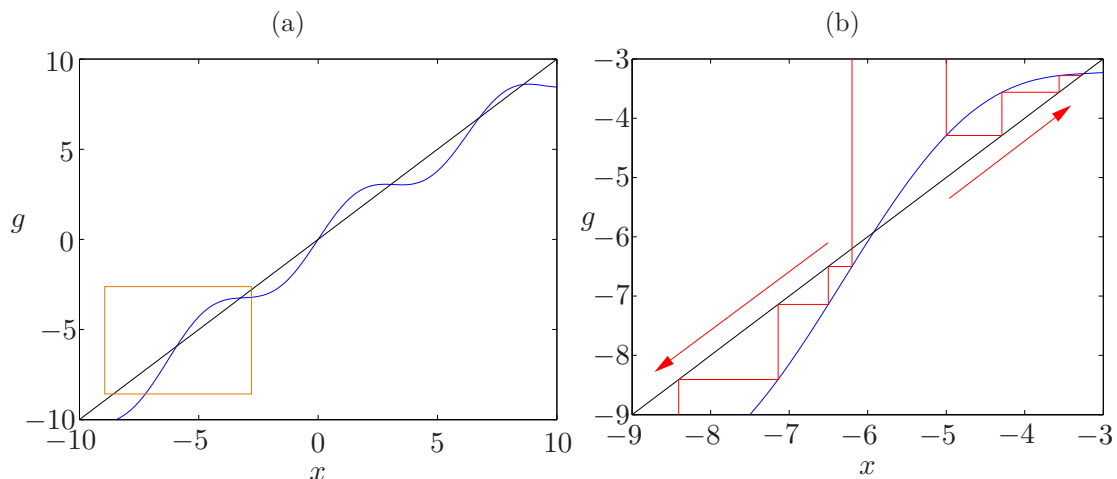


Abbildung 2.19: (a) Die beiden Seiten x (schwarz) und $g(x)$ (blau) aus (2.82) mit $\lambda = 0.01$. (b) Zoom in das Gebiet der beiden linken Schnittpunkte (Fixpunkte) in (a). Die roten Linien zeigen den Verlauf der Fixpunktiteration für die Startwerte $x_0 = -5$ und $x_0 = -6.2$.

Falls

1. $\forall_{x \in [a,b]} \quad g(x) \in [a, b]$,
2. $g(x)$ ist in $[a, b]$ diffbar,
3. $\exists_{K < 1} \quad \forall_{x \in [a,b]} \quad |g'(x)| \leq K$,

dann besitzt $g(x)$ genau einen Fixpunkt in $[a, b]$ und für alle Anfangswerte x_0 konvergiert die Iteration (2.82) gegen den Fixpunkt x^* , wobei

$$\lim_{i \rightarrow \infty} \frac{x_{i+1} - x^*}{x_i - x^*} = g'(x^*). \tag{2.83}$$

Die babylonische Methode die Wurzel aus einer beliebigen positiven reellen Zahl a zu ziehen, bestand aus der Fixpunktiteration

$$x_{i+1} = \frac{1}{2} \left(\frac{a}{x_i} + x_i \right), \tag{2.84}$$

denn diesen Ausdruck kann man für $i \rightarrow \infty$ in die Gleichung $x^{*2} = a$ umformen.

Auch das Newton-Verfahren (2.68) kann als Fixpunktiteration aufgefaßt werden, wenn man die Iterationsfunktion $g(x) := x - f(x)/f'(x)$ definiert. Mit einer Modifikation der Fixpunktiteration (Aitken- oder auch Steffensen-Verfahren) kann man die Konvergenz beschleunigen (siehe Quarteroni & Saleri 2006).

2.5.6 Newton-Verfahren für nichtlineare Gleichungssysteme

Wir wollen nun das Newton-Verfahren zur Bestimmung der Nullstelle einer nichtlinearen Funktion $f(x)$ auf nichtlineare Gleichungssysteme aus N Gleichungen in Abhängigkeit von N Unbekannten x_i übertragen. Diese Art von Problemstellung tritt häufig auf, wenn man z.B. stationäre Lösungen (Gleichgewichtslagen) nichtlinearer dynamischer Systeme wie etwas (4.5) oder (4.56) sucht (siehe auch Abb. 2.12).

Die simultan zu lösenden Gleichungen können man in der Form schreiben

$$f_i(x_1, \dots, x_N) = \vec{f}(\vec{x}) = 0, \quad i = 1, \dots, N. \quad (2.85)$$

Wir suchen nun den Vektor $\vec{x}^* = (x_1^*, \dots, x_N^*)^T$, so daß (2.85) erfüllt ist. Die gesuchte Nullstelle ist also ein Punkt \vec{x}^* im N -dimensionalen Raum.

Wie im eindimensionalen Fall sei hier $\vec{x}^{(0)}$ eine anfängliche Approximation der Nullstelle \vec{x}^* . Dann ergibt die Taylor-Entwicklung von \vec{f} im N -dimensionalen Raum um den Punkt $\vec{x}^{(0)}$

$$\vec{f}(\vec{x}) = \underbrace{\vec{f}(\vec{x}^{(0)}) + (\vec{x} - \vec{x}^{(0)}) \cdot \nabla \vec{f}(\vec{x}^{(0)})}_{:=\vec{p}(\vec{x})} + \text{Terme höherer Ordnung}. \quad (2.86)$$

Die Linearisierung der Vektorfunktion \vec{f} um den Punkt $\vec{x}^{(0)}$ lautet $\vec{f} = \vec{p}$. In zwei Dimensionen wird dann jede Komponente $f_i(x_1, x_2)$ der Funktion $\vec{f}(x_1, x_2)$ durch eine Ebene approximiert, die im Punkt $\vec{x}^{(0)}$ tangential zur Fläche $f_i(x_1, x_2)$. Entsprechend wird in N Dimensionen jede Hyperfläche $f_i(x_1, \dots, x_N)$ durch je eine Hyperebene approximiert, die im Punkt $\vec{x}^{(0)}$ tangential zur Hyperfläche $f_i(x_1, \dots, x_N)$ ist.

Wenn wir nun in (2.86) $\vec{p}(\vec{x}^{(1)}) = 0$ setzen, erhalten wir

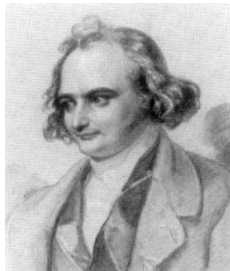
$$\vec{f}^{(0)} + (\vec{x}^{(1)} - \vec{x}^{(0)}) \cdot \nabla \vec{f}^{(0)} = 0. \quad (2.87)$$

Mit der Substitution $x^{(0)} \rightarrow \vec{x}^{(k)}$ und $x^{(1)} \rightarrow \vec{x}^{(k+1)}$ ergibt sich in Komponentenschreibweise

$$-f_i^{(k)} = \underbrace{(x_j^{(k+1)} - x_j^{(k)})}_{\delta_j^{(k)}} \frac{\partial f_i^{(k)}}{\partial x_j} = \left(\frac{\partial f_i^{(k)}}{\partial x_j} \right) \delta_j^{(k)}. \quad (2.88)$$

Wenn wir die *Jacobi-Matrix* definieren

$$J^{(k)} = \left. \frac{\partial f_i}{\partial x_j} \right|_{\vec{x}^{(k)}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\vec{x}^{(k)}) & \cdots & \frac{\partial f_1}{\partial x_N}(\vec{x}^{(k)}) \\ \vdots & & \vdots \\ \frac{\partial f_N}{\partial x_1}(\vec{x}^{(k)}) & \cdots & \frac{\partial f_N}{\partial x_N}(\vec{x}^{(k)}) \end{pmatrix}, \quad (2.89)$$



Carl Gustav Jacob Jacobi
1804–1851

können wir (2.88) auch kompakt vektoriell schreiben als

$$\mathbf{J}^{(k)} \cdot \vec{\delta}^{(k)} = -\vec{f}^{(k)}. \tag{2.90}$$

In der Praxis wird man diese Gleichung nie nach $\vec{x}^{(k+1)}$ auflösen, weil dazu $\mathbf{J}^{(k)}$ invertiert werden müßte. Stattdessen rechnet man mit dem *Korrekturvektor* $\vec{\delta}^{(k)} = \vec{x}^{(k+1)} - \vec{x}^{(k)}$. Der Algorithmus lautet dann:

Newton-Algorithmus

- Berechne $\vec{\delta}^{(k)}$ aus $\mathbf{J}^{(k)} \cdot \vec{\delta}^{(k)} = -\vec{f}^{(k)}$,
- Setze $\vec{x}^{(k+1)} = \vec{x}^{(k)} + \vec{\delta}^{(k)}$

Im Fall der Konvergenz verschwinden sowohl die Korrektur $\vec{\delta} = 0$ als auch das *Residuum* $\vec{f} = 0$.

Man kann folgenden Satz beweisen: Ist $\vec{f}(\vec{x})$ in einer Umgebung der Nullstelle \vec{x}^* , d.h. $\vec{f}(\vec{x}^*) = 0$, zweimal stetig differenzierbar und ist $\mathbf{J} = \nabla \vec{f} \Big|_{\vec{x}^*}$ regulär, dann konvergiert die Newton Iteration (2.90) bzw. (2.88) $\vec{x}^{(k)} \rightarrow \vec{x}^*$ gegen \vec{x}^* , falls $\vec{x}^{(0)}$ genügend nahe bei \vec{x}^* liegt. Die Konvergenz ist quadratisch, d.h.,

$$\|\vec{x}^{(k+1)} - \vec{x}^*\|_2 \leq c \|\vec{x}^{(k)} - \vec{x}^*\|_2^2, \tag{2.91}$$

wobei c eine Konstante ist.

Es ist hierbei wichtig, daß \mathbf{J} regulär ist, ansonsten geht die quadratische Konvergenz verloren, obwohl die Iteration immer noch gegen einen Grenzwert konvergieren kann. Die Bedingung, daß \mathbf{J} regulär ist, entspricht im eindimensionalen Fall der Bedingung $f'(x^*) \neq 0$.

In der Praxis genügen meist nur 1–4 Iterationen (vgl. Tab. 2.1). Ein Nachteil der Newton-Iteration besteht darin, daß alle N^2 Elemente der Jacobi-Matrix für jede Iteration neu berechnet werden müssen. Dies wird besonders dann teuer, wenn die Jacobi-Matrix dicht besetzt ist. Danach muß man auch noch in jedem Schritt ein lineares Gleichungssystem lösen.

Wenn man die Jacobi-Matrix nicht analytisch berechnen kann, muß man sie numerisch bestimmen. Eine Option ist dann, die exakten Ableitungen durch finite Vorwärts-Differenzen zu approximieren

$$\mathbf{J}(\vec{x}) = \frac{\partial f_i(\vec{x})}{\partial x_j} \approx \frac{f_i(\vec{x} + h\vec{e}_j) - f_i(\vec{x})}{h}. \tag{2.92}$$

Hierbei ist h eine kleine Schrittweite und \vec{e}_j der Einheitsvektor in j -Richtung.

Eine weitere Modifikation des Newton-Verfahrens besteht darin, die Jacobi-Matrix nicht in jeden Iterationsschritt neu zu berechnen, sondern für einige Schritt

mit derselben Jacobi-Matrix zu rechnen. Dies entspricht der sukzessiven Nullstellenapproximation mit einer konstanten Steigung (vgl. Abb. 2.14). Dieses Verfahren verringert jedoch die Konvergenzrate.

Wieder eine andere Modifikationen des Newton-Verfahren sind sogenannte *Quasi-Newton-Verfahren* der Form

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \omega^{(k)} \mathbf{H}^{(k)} \cdot \vec{f}^{(k)}. \quad (2.93)$$

Hierbei wurde die inverse Jacobi-Matrix $[\mathbf{J}^{(k)}]^{-1}$ durch eine einfachere Matrix $\mathbf{H}^{(k)}$ und einen Schrittweitenfaktor $\omega^{(k)}$ ersetzt, so daß sie im Limes der Konvergenz mit der inversen Jacobi-Matrix übereinstimmen (Ortega & Rheinboldt 1970).

2.5.7 Verfolgung der Lösung bei Parametervariation

In vielen Fällen hängt die Funktion noch von mindestens einem Parameter ab. In unserem Beispiel (2.63) ist dies der Parameter λ . Ein nicht-triviales Beispiel ist die Berechnung einer stationären Strömung. Gesucht wird dann das Geschwindigkeitsfeld und seine Abhängigkeit von der Reynoldszahl $\lambda = \text{Re}$. Die Komponenten des unbekanntes Vektors \vec{x} bestehen dann aus allen 3 Geschwindigkeitskomponenten an allen Knotenpunkten eines geeigneten Rechengitters. Die Vektorfunktion \vec{f} , die sich aus der *Navier-Stokes-Gleichung* ergibt, lebt dann in einem sehr hochdimensionalen Raum, der durch die Komponenten von \vec{x} aufgespannt wird. Die stationären Lösungen sind dann durch die Nullstellen $\vec{f} = 0$ im hochdimensionalen Raum \vec{x} gegeben.

Natürliche Parameterfortsetzung

Sei eine Lösung \vec{x}^* für $\lambda = 1$ bekannt, für die $\vec{f}(\vec{x}^*, \lambda = 1) = 0$ gilt. Dann kann es schwierig sein, mit diesem Startwert $\vec{x}^*(\lambda = 1)$ die Lösung von $\vec{f}(\vec{x}^*, \lambda = 1000) = 0$ zu finden, da sich durch die Änderung des Parameters λ (z.B. der Reynoldszahl) die Lösung $\vec{x}^*(\lambda)$ zu weit vom ursprünglichen Wert $\vec{x}^*(1)$ entfernt hat. In diesem Fall kann man λ schrittweise mit hinreichend kleinen Schritten $\Delta\lambda$ erhöhen und in jedem Schritt eine Newton-Iteration durchführen, wobei man als Startwert jeder Newton-Iteration die Lösung der jeweils vorherigen Newton-Iteration verwendet. Mit dieser Strategie kann man die Lösung $\vec{x}^*(\lambda)$ als Funktion des Parameters λ verfolgen.

Diese *natürliche Parameterfortsetzung* funktioniert, wenn sich die Lösung $\vec{x}^*(\lambda)$ kontinuierlich weiterentwickelt und keine Singularitäten auftreten. Es kann jedoch sein, daß sich die Lösungskurve zurückbiegt. Als Beispiel für dieses Verhalten betrachten wir unser eindimensionales Beispiel (2.63). Die Lösung $x^*(\lambda)$ als Funktion des Parameters λ ist in Abb. 2.20a gezeigt.⁹ Wenn man nun einen Lösungsast verfolgt, indem man den Parameter in kleinen Schritten erhöht, wird die Newton-Iteration an dem Punkt, an dem sich der Lösungsast zurückbiegt (*kritischer Punkt*),

⁹Hier ist einfach $\lambda = (x^*)^{-2} \sin(x^*)$.

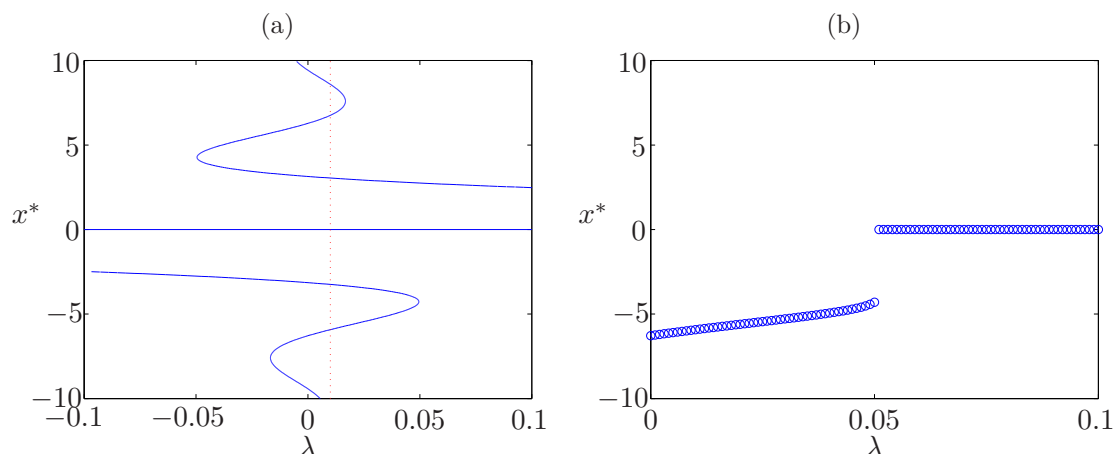


Abbildung 2.20: (a) Nullstellen $x^*(\lambda)$ des Problems (2.63). Die gestrichelte rote Linie entspricht $\lambda = 0.01$. (b) Resultat der Newton-Iteration bei natürlicher Parametervariation mit ansteigende Werten von λ .

im günstigsten Fall auf eine andere Lösung springen. Dies ist in Abb. 2.20b dargestellt. Damit hat man aber den ursprünglichen Lösungsast verloren.

Am kritischen Punkt wird bei einer Parametervariation ein Paar neuer Lösungen kreiert oder vernichtet. Dies nennt man auch *Sattel-Knoten-Verzweigung*. Die Situation ist noch einmal schematisch in Abb. 2.21 gezeigt. In solchen Fällen kommt man mit der natürlichen Parameter-Fortsetzung nicht weiter und man muß andere Methoden der Verfolgung verwenden.

Pseudo-Bogenlängen-Fortsetzung

Eine bessere Möglichkeit der Verfolgung von Lösungen ist die *Pseudo-Bogenlängen-Fortsetzung* (*pseudo-arclength continuation*). Dieses Verfahren wurde zuerst von Keller (1977) vorgeschlagen. Hierbei wird der Parameter λ als zusätzliche Variable aufgefaßt. Im $(N+1)$ -dimensionalen Raum (\vec{x}, λ) definieren dann die N Gleichungen $\vec{f}(\vec{x}, \lambda) = 0$ eine eindimensionale Untermenge: die Lösungskurve $\vec{x}^*(\lambda)$. Um einen Punkt auf dieser Lösungskurve festzulegen, der das Ziel der Newton-Iteration sein soll, brauchen wir eine weitere Gleichung. Diese Gleichung besteht in der Forderung, daß sich der iterierte Punkt $(\vec{x}^{(k)}, \lambda^{(k)})$ im (\vec{x}, λ) -Raum in einem fest vorgegebenen Abstand Δs (pseudo-Bogenlänge) von dem vorherigen, iterierten und konvergierten Punkt (\vec{X}^*, Λ^*) auf der Lösungskurve befindet (Abb. 2.21). Dann gilt für die Werte von $\vec{x}^{(k)}$ und $\lambda^{(k)}$ die Einschränkung

$$\left\| \vec{x} - \vec{X}^* \right\|_2^2 + (\lambda - \Lambda^*)^2 = (\Delta s)^2. \quad (2.94)$$

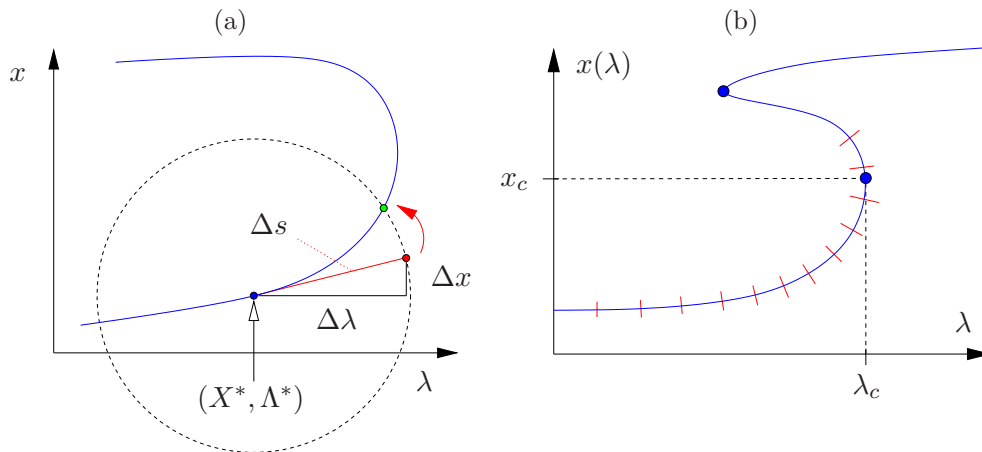


Abbildung 2.21: (a) Ausgehend von einer konvergierten Lösung (X^*, Λ^*) (blauer Punkt) wird von der neuen Newton-Iteration verlangt, daß der Abstand vom alten konvergierten Wert konstant (Δs) bleibt. Dann verläuft die Iteration (angedeutet durch den roten Pfeil) von einer geeigneten ersten Schätzung (roter Punkt) zur neuen Lösung grüner Punkt. Wenn die Schrittweite Δs (Abstand der Schätzung von dem alten konvergierten Wert, rote Linie) klein ist, kann der blaue Lösungsast relativ glatt verfolgt werden. (b) Lösung x von $f(x, \lambda) = 0$ als Funktion des Parameters λ . Mit Hilfe der Pseudo-Bogenlängen-Fortsetzung kann man die Lösung auch über kritische Punkte (Sattel-Knoten-Verzweigungen, blaue Punkte) hinaus verfolgen. Die roten Balken symbolisieren die Schritte mit konstanter quasi-Bogenlänge.

Wir fassen nun \vec{x} und λ als die gesuchten Variablen auf und betrachten das erweiterte Nullstellenproblem im $n + 1$ -dimensionalen Raum

$$\vec{f}(\vec{x}, \lambda) = 0, \quad (2.95a)$$

$$g(\vec{x}, \lambda) := (\lambda - \Lambda^*)^2 + \|\vec{x} - \vec{X}^*\|_2^2 - (\Delta s)^2 = 0. \quad (2.95b)$$

Die Newton-Iteration für dieses erweiterte System können wir dann formulieren als (vgl. den Newton-Algorithmus auf Seite 38)

$$\begin{pmatrix} \frac{\partial f_i}{\partial x_j} & \frac{\partial f_i}{\partial \lambda} \\ \frac{\partial g}{\partial x_j} & \frac{\partial g}{\partial \lambda} \end{pmatrix}^{(k)} \cdot \begin{pmatrix} x_j^{(k+1)} - x_j^{(k)} \\ \lambda^{(k+1)} - \lambda^{(k)} \end{pmatrix} = - \begin{pmatrix} f_i \\ g \end{pmatrix}^{(k)}. \quad (2.96)$$

Wenn wir die partiellen Ableitungen nach λ ausführen (Summenkonvention)

$$\left. \frac{\partial g}{\partial x_j} \right|^{(k)} = 2(x_i - X_i^*) \underbrace{\left. \frac{\partial x_i}{\partial x_j} \right|^{(k)}}_{=\delta_{ij}} = 2(x_j - X_j^*)^{(k)} = 2(x_j^{(k)} - X_j^*), \quad (2.97a)$$

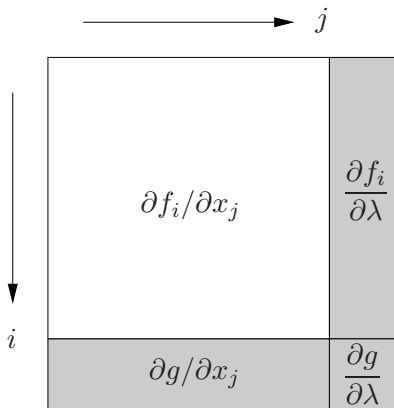


Abbildung 2.22: Struktur der erweiterten Jacobi-Matrix bei Verwendung der Pseudo-Bogenlängen-Fortsetzung.

$$\left. \frac{\partial g}{\partial \lambda} \right|^{(k)} = 2(\lambda - \Lambda^*)^{(k)} = 2(\lambda^{(k)} - \Lambda^*) \quad (2.97b)$$

erhalten wir

$$\begin{bmatrix} \frac{\partial f_i}{\partial x_j} & \frac{\partial f_i}{\partial \lambda} \\ 2(x_j - X_j^*) & 2(\lambda - \Lambda^*) \end{bmatrix}^{(k)} \cdot \begin{pmatrix} x_j^{(k+1)} - x_j^{(k)} \\ \lambda^{(k+1)} - \lambda^{(k)} \end{pmatrix} = - \begin{pmatrix} f_i \\ g \end{pmatrix}^{(k)} \quad (2.98)$$

Die Struktur der erweiterten Jacobi Matrix ist in Abb. 2.22 illustriert. Beachte, daß die reduzierte Jacobi-Matrix $\partial f_i/\partial x_j$ am kritischen Punkt singularär wird. Dies liegt daran, daß sich am kritischen Punkt $\lambda = \lambda_c$ bei Variation von λ zwei Nullstellen gegenseitig vernichten. Dies geschieht in unserem Beispiel bei $\lambda \approx 0.05$ (Abb. 2.20a). Gleichung (2.98) muß nun iteriert werden. Dazu benötigen wir möglichst gute Startwerte.

Vorhersage-Schritt

Gute Startwerte $(\vec{x}^{(0)}, \lambda^{(0)})$ (rote Punkte in Abb. 2.23) kann man erhalten, indem man von dem konvergenten Punkt (\vec{X}^*, Λ^*) auf dem Lösungsast $\vec{f}(\vec{x}, \lambda) = 0$ ausgeht (grüne Punkte in Abb. 2.23) und einen Schritt der Länge Δs tangential zum Lösungsast $\vec{f}(\vec{x}, \lambda) = 0$ vorwärts geht (blaue Linien in Abb. 2.23). Dazu muß man im $(N+1)$ -dimensionalen Raum (N Variablen \vec{x} und 1 Parameter λ) die tangentielle Richtung an die Lösungskurve $\vec{f} = 0$ im Punkte (\vec{X}^*, Λ^*) berechnen. Sei der Tangentialvektor $\vec{T} = (\vec{t}, t^{(\lambda)})^T$. Diese Richtung ist dadurch gekennzeichnet, daß \vec{f} sich in Richtung von \vec{T} nicht ändert. Die Änderungsrate in \vec{T} -Richtung ist aber gegeben durch die Richtungsableitung $\vec{T} \cdot \tilde{\nabla}$ wobei $\tilde{\nabla} = (\nabla, \partial_\lambda)^T$ der Nabla-Operator im $(N+1)$ -dimensionalen Raum ist. Daher muß für den Tangentenvektor $\vec{T} = (\vec{t}, t^{(\lambda)})^T$ im $(N+1)$ -dimensionalen Raum gelten

$$\vec{T} \cdot \tilde{\nabla} \vec{f} = \vec{t} \cdot \nabla \vec{f} + t^{(\lambda)} \partial_\lambda \vec{f} \stackrel{!}{=} 0. \quad (2.99)$$

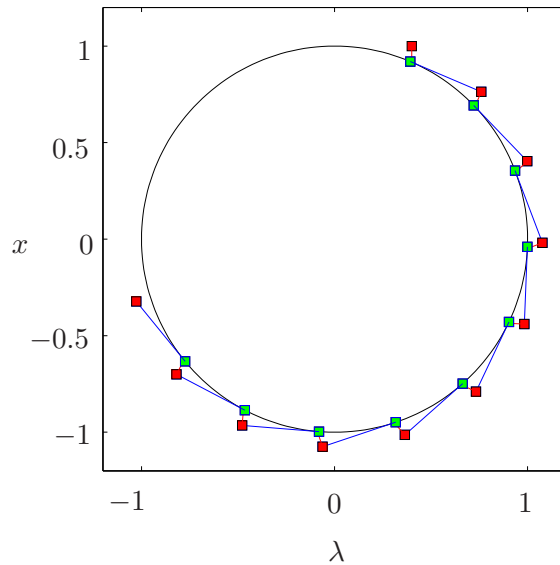


Abbildung 2.23: Quasi-Bogenlängen-Verfolgung am Beispiel der Funktion $f(x, \lambda) = x^2 + \lambda^2 - 1$. Die Nullstelle $f(x, \lambda) = 0$ ist als schwarzer Kreis dargestellt. Die Anfangswerte einer jeden Newton-Iteration sind die roten Punkte. Der Verlauf jeder Newton-Iteration ist als rote Linie gezeigt. Die konvergierten Nullstellen sind mit grünen Punkten symbolisiert. Die erste Iterierte liegt schon innerhalb der Größe grünen Symbole. Als Abbruchkriterium wurde $\epsilon = 10^{-10}$ gewählt, was im Schnitt 5 Newton-Iterationen erforderlich machte.

Dies sind N Bedingungen für die $N + 1$ Komponenten von \vec{T} . Der Tangentenvektor ist durch (2.99) nicht eindeutig bestimmt, da sein Betrag noch frei ist. Deshalb kann man zum Beispiel $T_1 = 1$ setzen und so (2.99) eindeutig lösen. Da wir jedoch um die Länge Δs in \vec{T} -Richtung gehen wollen, setzen wir durch die Normierung

$$\vec{T} \longrightarrow \frac{\vec{T}}{\|\vec{T}\|_2} \Delta s \quad (2.100)$$

die Länge von \vec{T} auf Δs fest. Wenn man $\vec{T} = (\vec{t}, t^{(\lambda)})^T$ so bestimmt hat, wird die nächste Newton-Iteration nach (2.98) mit dem Startwert

$$\begin{pmatrix} \vec{x}^{(0)} \\ \lambda^{(0)} \end{pmatrix} = \begin{pmatrix} \vec{X}^* \\ \Lambda^* \end{pmatrix} + \begin{pmatrix} \vec{t} \\ t^{(\lambda)} \end{pmatrix} \quad (2.101)$$

begonnen.

Beispiel Als Beispiel betrachten wir

$$f(x, \lambda) = x^2 + \lambda^2 - 1. \quad (2.102)$$

Die gesuchte Lösung von $f(x, \lambda) = 0$ ist der schwarze Kreis in Abb. 2.23. Ausgehend von der bekannten Lösung $(X^*, \Lambda^*) = (1, 0)$, einer Pseudo-Bogenlänge $\Delta s = 0.4$

2 Interpolation und Approximation

und einer Anfangsschätzung $(x^{(0)}, \lambda^{(0)}) = (1, 0.4)$ (erster **roter** Punkt in Abb. 2.23) wird nach (2.98) iteriert. Die Matrix aus (2.98) lautet dann

$$\begin{bmatrix} \frac{\partial f_i}{\partial x_j} & \frac{\partial f_i}{\partial \lambda} \\ 2(x_j - X_j^*) & 2(\lambda - \Lambda^*) \end{bmatrix}^{(k)} = \begin{bmatrix} 2x & 2\lambda \\ 2(x - X^*) & 2(\lambda - \Lambda^*) \end{bmatrix}^{(k)}. \quad (2.103)$$

Die Iteration wird als konvergent erachtet, wenn für die Korrektur von x und λ gilt $(x^{(k+1)} - x^{(k)})^2 + (\lambda^{(k+1)} - \lambda^{(k)})^2 \leq \epsilon^2$ (erster **grüner** Punkt in Abb. 2.23). Damit haben wir unsere neue Nullstelle (X^*, Λ^*) berechnet.

Nun benötigen wir einen neuen Vorhersage-Schritt. Dafür betrachten wir die eindimensionale Version von (2.99)

$$t^{(x)} \frac{\partial f}{\partial x} \Big|_{X^*, \Lambda^*} + t^{(\lambda)} \frac{\partial f}{\partial \lambda} \Big|_{X^*, \Lambda^*} = 0. \quad (2.104)$$

Wenn wir $t^{(x)} = 1$ setzen, ergibt sich

$$t^{(\lambda)} = - \left(\frac{\partial f / \partial x}{\partial f / \partial \lambda} \right)_{X^*, \Lambda^*}. \quad (2.105)$$

Die Länge des Tangentialvektors ist dann

$$N = \left\| \vec{T} \right\|_2 = \sqrt{(t^{(x)})^2 + (t^{(\lambda)})^2} = \sqrt{1 + \left(\frac{\partial f / \partial x}{\partial f / \partial \lambda} \right)_{X^*, \Lambda^*}^2}. \quad (2.106)$$

Damit erhalten wir einen Tangentialvektor der Länge Δs mit den Komponenten

$$t^{(x)} = \frac{\Delta s}{N}, \quad (2.107a)$$

$$t^{(\lambda)} = - \left(\frac{\partial f / \partial x}{\partial f / \partial \lambda} \right)_{X^*, \Lambda^*} \frac{\Delta s}{N}. \quad (2.107b)$$

In unserem Beispiel muß man noch $t^{(x)} \rightarrow -\text{sign}(\partial f / \partial \lambda) t^{(x)}$ einfügen, damit die Lösung wie in Abb. 2.23 im Uhrzeigersinn durchlaufen wird.

Man kann nun mit $(x^{(0)}, \lambda^{(0)}) = (X^* + t^{(x)}, \Lambda^* + t^{(\lambda)})$ (zweiter **roter** Punkt in Abb. 2.23) die nächste Newton-Iteration starten. Damit ist es möglich, die gesamte Lösung (Kreis) zu verfolgen, obwohl die Lösung $x^*(\lambda)$ als Funktion von λ nicht eindeutig ist.

Man erkennt leicht, daß die Quasi-Bogenlänge $s = n\Delta s$ der wirklichen Bogenlänge der Lösungskurve umso näher kommt, desto geringer die Schrittweite Δs ist.