

Nichtlineare Gleichungssysteme

Funktionen mit mehreren Variablen

auch multivariat genannt, hier nur skalarwertige Funktionen (keine Komplexen Zahlen).  
Definition:

Explizite Darstellung Funktionsgleichung ist nach einer variablen aufgelöst.

$y = f(x_1, x_2, \dots, x_n)$

Implizite Darstellung nicht nach einer Variablen aufgelöst(nur n-1 unabhängige Variablen)

$F(x_1, x_2, \dots, x_n) = 0$

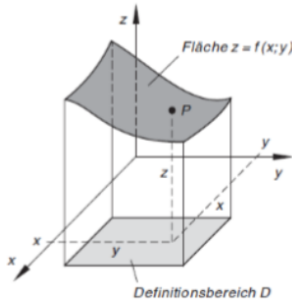
für vektorwertige funktionen

$\vec{f}(x_1, \dots, x_n) = \begin{pmatrix} y_1 = f_1(x_1, \dots, x_n) \\ \vdots \\ y_m = f_m(x_1, \dots, x_n) \end{pmatrix}$

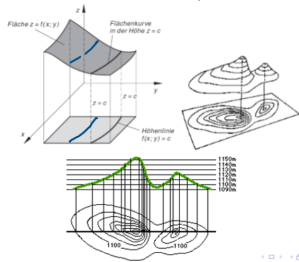
Graphische Darstellungsformen

Funktionen mit 2 Variablen können 3D dargestellt werden.  
Interpretieren als  $z = f(x, y)$

Fläche Punkte  $(x, y, f(x, y))$

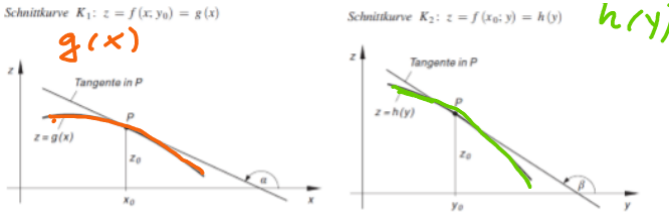
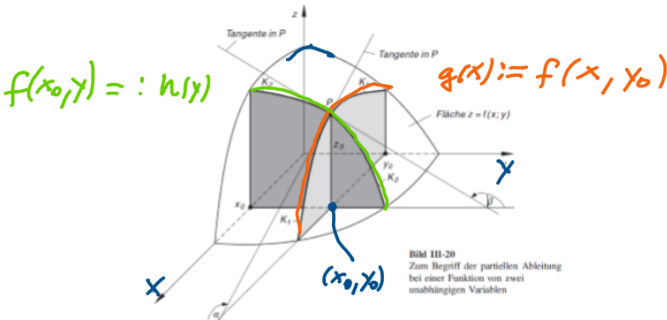


Schnittkurve bei konstanter Höhe  $z$ , auch Höhen-/Contour-Plot



Partielle Ableitungen

Nur eine der Variablen wird abgeleitet, der Rest als Konstante behandelt. Visuell entspricht dies der Steigung an einer Flächentangente.



Partielle Ableitungen 1. Ordnung

Beispiel nach x abgeleitet(normale Ableitungsregeln, andere Variablen als Konstanten betrachten):

$\frac{\partial f}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$

$f(x, y) = 3xy^3 + 10x^2y + 5y + 3y * \sin(5xy)$

$\frac{\partial f}{\partial x} = 3 * 1 * y^3 + 10 * 2x + 0 + 3y * \cos(5xy) * 5 * 1 * y$

Linearisierung

Repetition Tangentengleichung

Dient als Annäherung für eindimensionale  $f(x)$  in der Nähe von  $x_0$  (Linearisierung):  
 $g(x) = f(x_0) + f'(x_0)(x - x_0)$

Jacobi-Matrix

Sozusagen wie Tangentengleichung aber für mehrere Variablen

$$\vec{y} = \vec{f}(\vec{x}) = \begin{pmatrix} y_1 = f_1(x_1, \dots, x_n) \\ \vdots \\ y_m = f_m(x_1, \dots, x_n) \end{pmatrix}$$

Die Jacobi-Matrix enthält sämtliche Partielle Ableitungen 1. Ordnung von  $\vec{f}$ .  
Auf jeder Spalte bleibt die funktion  $f_j$  die gleiche und in den Zeilen  $x_i \rightarrow \frac{\partial f_j}{\partial x_i}$

$$D\vec{f}(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \dots & \frac{\partial f_1}{\partial x_n}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) & \dots & \frac{\partial f_2}{\partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(x) & \frac{\partial f_m}{\partial x_2}(x) & \dots & \frac{\partial f_m}{\partial x_n}(x) \end{pmatrix}$$

verallgemeinerte Tangentengleichung

$$\vec{g}(\vec{x}) = \vec{f}(\vec{x}^{(0)}) + D\vec{f}(\vec{x}^{(0)}) * (\vec{x} - \vec{x}^{(0)})$$

ist eine lineare Funktion und für  $\vec{x}$  in der Umgebung von  $\vec{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$  gilt  $\vec{f}(\vec{x}) \approx \vec{g}(\vec{x})$

$Df$  entspricht der obigen Funktion zur Erzeugung einer Jacobi-Matrix.

Hochgestellte Zahlen in Klammern  $(x^{(n)})$  stehen wie zuvor für eine Variable nach  $n$  Iterationsschritten.

Tangentialebene

- Für den speziellen Fall  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  mit  $y = f(x_1, x_2)$  und  $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)})^T \in \mathbb{R}^2$  ist die Jacobi-Matrix nur ein Zeilenvektor mit zwei Elementen, nämlich

$$Df(\mathbf{x}^{(0)}) = \left( \frac{\partial f}{\partial x_1}(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}) \quad \frac{\partial f}{\partial x_2}(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}) \right).$$

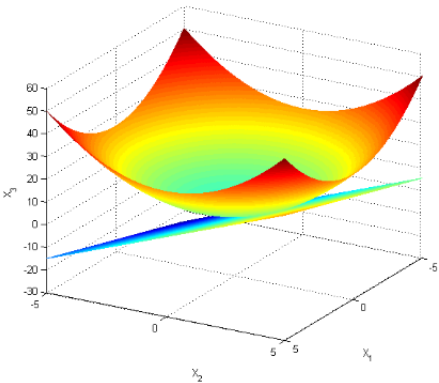
Dann liefert die Linearisierung

$$\begin{aligned} g(x_1, x_2) &= f(x_1^{(0)}, x_2^{(0)}) + \begin{pmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}) & \frac{\partial f}{\partial x_2}(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}) \end{pmatrix} \cdot \begin{pmatrix} x_1 - x_1^{(0)} \\ x_2 - x_2^{(0)} \end{pmatrix} \\ &= f(x_1^{(0)}, x_2^{(0)}) + \frac{\partial f}{\partial x_1}(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}) \cdot (x_1 - x_1^{(0)}) + \frac{\partial f}{\partial x_2}(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}) \cdot (x_2 - x_2^{(0)}) \end{aligned}$$

die Gleichung der **Tangentialebene**.

- Sie enthält sämtliche im Flächenpunkt  $P = (x_1^{(0)}, x_2^{(0)}, f(x_1^{(0)}, x_2^{(0)}))$  an die Bildfläche von  $y = f(x_1, x_2)$  angelegten Tangenten.

Graphische Darstellung der Fläche  $x_3 = f(x_1, x_2) = x_1^2 + x_2^2$  sowie Tangentialebene durch den Flächenpunkt  $(x_1^{(0)} = 1, x_2^{(0)} = 2, f(x^{(0)}) = 5)$



Nullstellenbestimmung für nichtlineare Systeme

- Gegeben:  $n \in \mathbb{N}$  und eine Funktion  $\vec{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$
- Gesucht:  $\vec{x} \in \mathbb{R}^n$  mit  $\vec{f}(\vec{x}) = 0$

$$\vec{f}(x_1, \dots, x_n) = \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_n(x_1, \dots, x_n) \end{pmatrix} = \vec{0}$$

Newton-Verfahren für Systeme

Herleitung

Repetition 1-Dimensional: (nur für  $f : \mathbb{R} \rightarrow \mathbb{R}$ )

Aus der Linearisierung der Funktion  $f$  mittels der Tangente  $g$  an der Stelle  $x_n$

$$f(x) \approx g(x) = f(x_n) + f'(x_n)(x - x_n)$$

folgte die Iteration

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (n = 0, 1, 2, 3, \dots).$$

Mit der Jacobi-Matrix  $Df(x)$  kann das analog für Vektor-wertige Funktionen  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  angewendet werden.

$$\vec{x}^{(n+1)} = \vec{x}^{(n)} - (Df(\vec{x}^{(n)}))^{-1} * \vec{f}(\vec{x}^{(n)})$$

Das Inverse der Jacobi-Matrix wird aber nie berechnet sondern die obige Gleichung via Substitution als lineares Gleichungssystem aufgefasst.

$$\delta^{(n)} := - \left( Df(\mathbf{x}^{(n)}) \right)^{-1} \cdot \mathbf{f}(\mathbf{x}^{(n)})$$

als lineares Gleichungssystem auffasst gemäss

$$Df(\mathbf{x}^{(n)})\delta^{(n)} = -\mathbf{f}(\mathbf{x}^{(n)})$$

und so  $\delta^n$  bestimmen und anschliessend

$$\mathbf{x}^{(n+1)} := \mathbf{x}^{(n)} + \delta^{(n)}$$

Quadratisch-konvergentes Newton-Verfahren für Systeme

Gesucht: Nullstellen von  $\vec{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  mit Startvektor  $\vec{x}^{(0)}$  Nahe der Nullstelle.  
es kann passieren, dass das Newton-Verfahren Statt einer Nullstelle ein lokales Minimum  $x_{min} = 0$  findet, in diesem Fall ist  $Df(x_{min})$  aber immer nicht regulär.

für  $n = 0, 1, \dots$ :

1. Berechne  $\delta^{(n)}$  als Lösung des linearen Gleichungsystems

$$D\vec{f}(\vec{x}^{(n)})\delta^{(n)} = -\vec{f}(\vec{x}^{(n)})$$

2. Setze

$$\vec{x}^{(n+1)} = \vec{x}^{(n)} + \delta^{(n)}$$

mögliche Abbruchkriterien

- $n \geq n_{max}, n_{max} \in \mathbb{N}$
- $||x^{(n+1)} - x^{(n)}|| \leq \epsilon \iff ||\delta^{(n)}|| \leq \epsilon$
- $||x^{(n+1)} - x^{(n)}|| \leq \epsilon * ||x^{(n+1)}||$
- $||\vec{f}(x^{(n+1)})|| \leq \epsilon$

Konvergenz

Das Newton-Verfahren konvergiert quadratisch für nahe genug an einer Nullstelle  $\vec{x}$  liegende Startvektoren, wenn  $D\vec{f}(\vec{x})$  regulär und  $\vec{f}$  dreimal stetig differenzierbar ist.

Vereinfachtes Newton-Verfahren für Systeme

Gesucht: Nullstellen von  $\vec{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  mit Startvektor  $\vec{x}^{(0)}$  Nahe der Nullstelle.

Der Unterschied zum Quadratisch-konvergenten Newton-Verfahren liegt darin, dass nur einmal die Jacobi-Matrix für den Startvektor berechnet werden muss (rot)  $\rightarrow$  weniger Aufwand aber konvergiert langsamer (linear).

für  $n = 0, 1, \dots$ :

1. Berechne  $\delta^{(n)}$  als Lösung des linearen Gleichungsystems

$$D\vec{f}(\vec{x}^{(0)})\delta^{(n)} = -\vec{f}(\vec{x}^{(n)})$$

2. Setze

$$\vec{x}^{(n+1)} = \vec{x}^{(n)} + \delta^{(n)}$$

**Gedämpftes Newton-Verfahren**

Die Idee ist bei jedem Iterationsschritt zu überprüfen ob es sich um eine Verbesserung handelt und falls nicht, es stattdessen mit einem gedämpften Schritt zu versuchen.

Die gewählte Dämpfungsgrösse  $k_{max}$  ist stark vom Problem abhängig.  $k_{max} = 4$  ist ein vernünftiger Standard Wert.

für  $n = 0, 1, \dots$ :

1. Berechne  $\delta^{(n)}$  als Lösung des linearen Gleichungssystems

$$D\vec{f}(\vec{x}^{(n)})\delta^{(n)} = -\vec{f}(\vec{x}^{(n)})$$

2. Finde das minimale  $k \in 0, 1, \dots, k_{max}$  mit

$$\|\vec{f}(\vec{x}^{(n)}) + \frac{\delta^{(n)}}{2^k}\|_2 < \|\vec{f}(\vec{x}^{(n)})\|_2$$

3. Setze (falls kein minimales  $k \rightarrow k = 0$ )

$$\vec{x}^{(n+1)} = \vec{x}^{(n)} + \frac{\delta^{(n)}}{2^k}$$