

Newton-Verfahren als Fixpunktiteration:

Dazugehörige Iterationsfunktion ist

$$\Phi(x) = x - \frac{f(x)}{f'(x)}.$$

Gesuchte Nullstelle \bar{x} von $f(x)$ ist gleichzeitig Fixpunkt von $\Phi(x)$ (falls $f'(\bar{x}) \neq 0$):

$$x = \Phi(x) = x - \frac{f(x)}{f'(x)} \Leftrightarrow \frac{f(x)}{f'(x)} = 0$$

Damit sind die Resultate über Iterationsfunktionen und deren Fixpunkte anwendbar (Fixpunktsatz, $L < 1$, usw.):

$$\Phi'(x) = 1 - \frac{f'^2(x) - f(x)f''(x)}{f'^2(x)} = \frac{f(x)f''(x)}{f'^2(x)}$$

und (falls \bar{x} einfache Nullstelle von f) gilt:

$$|\Phi'(\bar{x})| = 0 < 1 .$$

Daher ist die gesuchte Nullstelle \bar{x} von f dann ein anziehender Fixpunkt von Φ !

6.2.2. Satz:

Das Newtonverfahren für eine stetig diff'bare Funktion f mit einfacher Nullstelle \bar{x} ist lokal quadratisch konvergent.

Beweis:

Lokal konvergent nach Banach'schem Fixpunktsatz!

Startwert nahe genug bei Fixpunkt \rightarrow lineare Konvergenz im

Intervall $\mathbf{U = [\bar{x}-h, \bar{x}+h]}$

Zum Beweis der quadratische Konvergenz:

$$0 = f(\bar{x}) = f(x_k) + f'(x_k)(\bar{x} - x_k) + \frac{1}{2} f''(z)(\bar{x} - x_k)^2$$

mit Zwischenstelle z (Taylorentwicklung).

Umformung:

$$\bar{x} - \boxed{x_k + \frac{f(x_k)}{f'(x_k)}} = -\frac{1}{2} \cdot \frac{f''(z)}{f'(x_k)} (\bar{x} - x_k)^2$$

$$\left| \bar{x} - \boxed{x_{k+1}} \right| = \left| \frac{f''(z)}{2f'(x_k)} \right| \cdot |\bar{x} - x_k|^2 = C(x_k, \bar{x}) \cdot |\bar{x} - x_k|^2$$

Ist $f'(\bar{x}) \neq 0$, so kann C lokal durch eine Konstante L nach oben beschränkt werden.

Daher folgt:
$$|\bar{x} - x_{k+1}| \leq L \cdot |\bar{x} - x_k|^2$$

Der Abstand von der gesuchten Lösung verringert sich in jedem Schritt quadratisch (wenn man nahe genug an der Lösung ist, so dass $|\bar{x}-x_k| \ll 1$).

z.B. Abstand von der Lösung in jedem Schritt $|\bar{x}-x_k|$ z.B.

wie 10^{-1} , 10^{-2} , 10^{-4} , 10^{-8} , 10^{-16}

Also schnelle Konvergenz!

Voraussetzungen für quadratische Konvergenz:

- \bar{x} einfache Nullstelle, d.h.
 $f(\bar{x}) = 0$, aber $f'(\bar{x}) \neq 0$, oder
 $f(x) = (\bar{x}-x)g(x)$ mit $g(\bar{x}) \neq 0$
- Startwert x_0 nahe bei \bar{x}

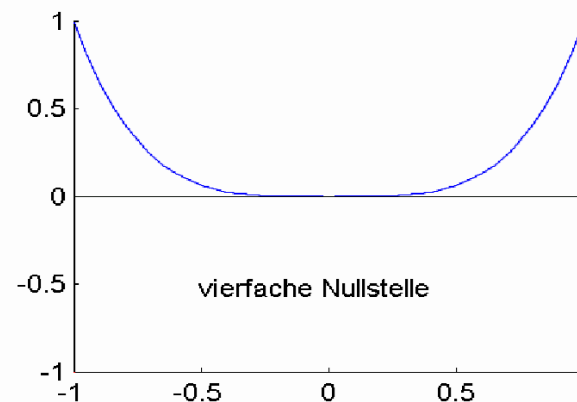
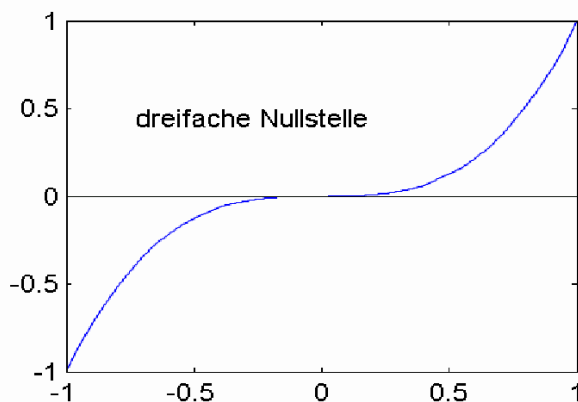
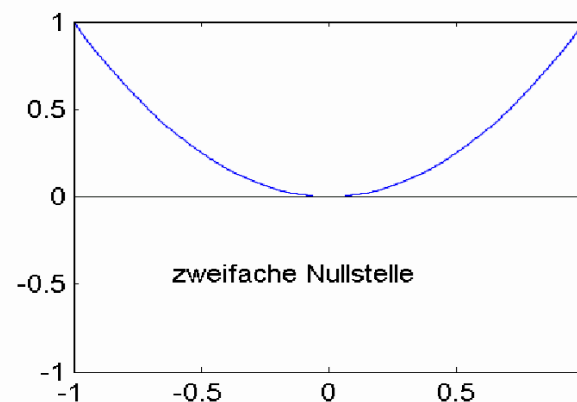
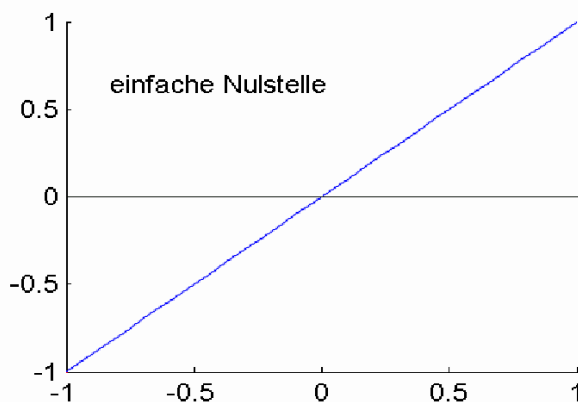
a heißt **m-fache Nullstelle**, wenn:

$$f^{(j)}(a) = 0 \text{ für } j=0, \dots, m-1, \text{ aber } f^{(m)}(a) \neq 0$$

oder

$$f(x) = (x-a)^m \cdot g(x) \text{ mit } g(a) \neq 0$$

Graphisch:



6.2.3. Definition der Konvergenzordnung:



Linear konvergent: $|\bar{x} - x_{k+1}| \leq L \cdot |\bar{x} - x_k|$ und $L < 1$

Konvergent von Ordnung $p > 1$: $|\bar{x} - x_{k+1}| \leq L \cdot |\bar{x} - x_k|^p$

Falls f an der Stelle \bar{x} eine m -fache Nullstelle hat mit $m > 1$, so ist das Newtonverfahren nur noch **lokal linear** konvergent:

Ist nämlich \bar{x} eine m -fache Nullstelle, so folgt

$$f(x) = (x - \bar{x})^m g(x) \quad \text{mit} \quad g(\bar{x}) \neq 0$$

und die Iterationsfunktion lautet

$$\Phi(x) = x - \frac{f(x)}{f'(x)} = x - \frac{(x - \bar{x})^m g(x)}{m(x - \bar{x})^{m-1} g(x) + (x - \bar{x})^m g'(x)}$$

Die Ableitung der Iterationsfunktion ist daher

$$\Phi'(\bar{x}) = 1 - \frac{g(\bar{x})}{m \cdot g(\bar{x})} = 1 - \frac{1}{m} < 1$$

Nach dem Banach'schen Fixpunktsatz folgt:

Φ ist lokal kontrahierend und es liegt *lineare* Konvergenz vor!

Variationen bei m-facher Nullstelle, um quadratische Konvergenz zu erreichen:

Wende Newtonverfahren an auf

(m-1) – te Ableitung von f(x)

$f(x)/f'(x)$

m-te Wurzel von $|f(x)|$

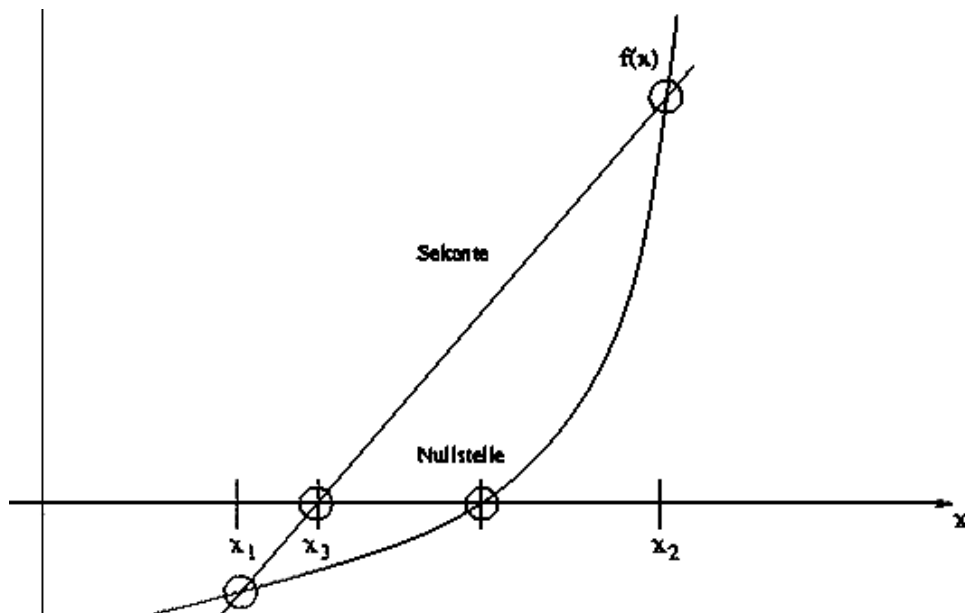
oder modifiziere Newtonformel (bei bekanntem m) zu

$$\Phi(x) = x - m \cdot \frac{f(x)}{f'(x)}$$

6.2.4. Verwandte Verfahren

Sekantenverfahren:

Ersetze Tangente in letztem Punkt durch die Sekante, die die beiden letzten Punkte verbindet; verwende deren Nullstelle!



$$\approx f'(x_k)$$

$$g(x) = f(x_{k-1}) + \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \cdot (x - x_{k-1})$$

Nullstelle der Sekante als nächste Iterierte:

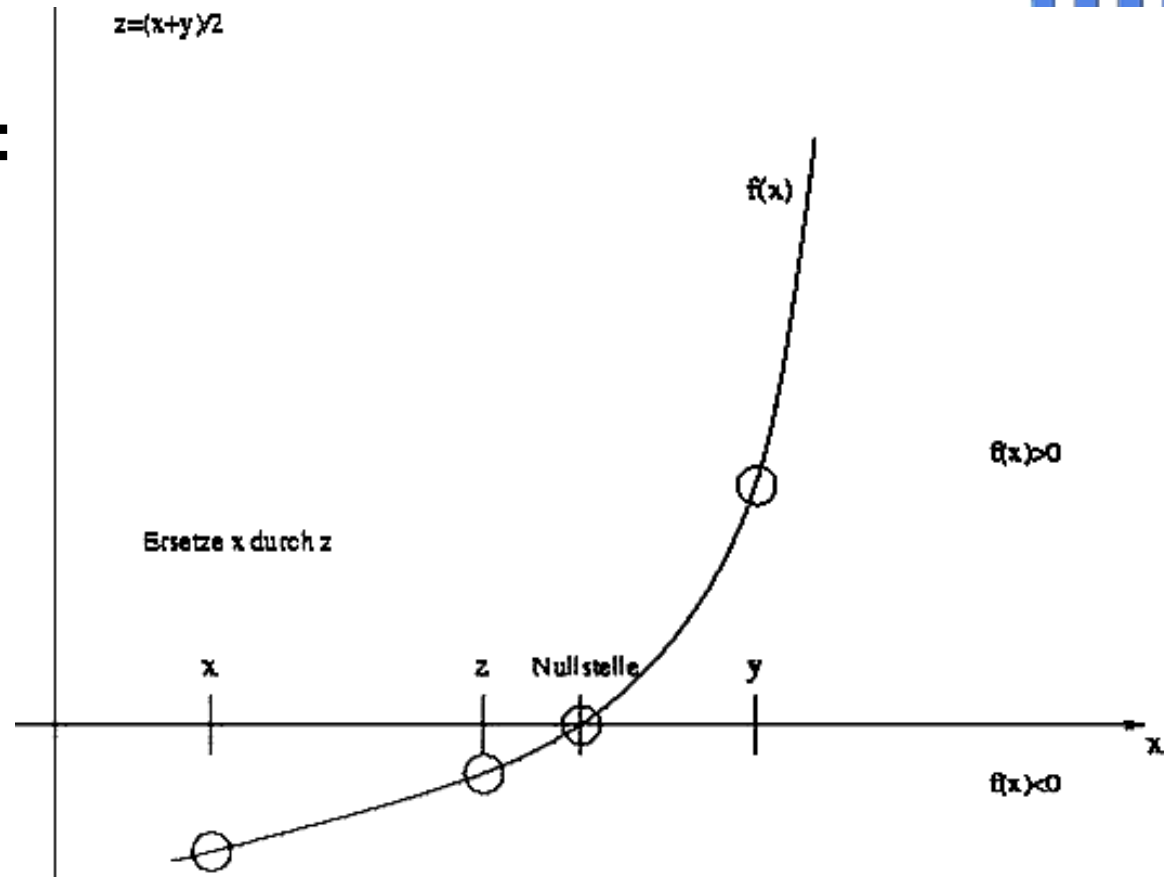
$$x_{k+1} = x_{k-1} - \frac{f(x_{k-1})(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})} = \frac{x_{k-1}f(x_k) - x_k f(x_{k-1})}{f(x_k) - f(x_{k-1})}$$

Liefert lokale Konvergenz wie beim Newtonverfahren.
 Vorteil: keine Ableitungen benötigt, jeweils nur die letzten beiden Funktionswerte! Billiger!
 Aber Problem, falls Nenner gleich Null ist!

Konvergenzordnung p nur mit $1 < p < 2$, aber dafür pro Schritt nur eine neue Funktionsauswertung nötig
 (bei Newton f und f' pro Schritt)



Bisektionsverfahren:



Starte mit zwei Werten x und y , für die gilt, dass $f(x) \cdot f(y) < 0$ ist, für die also f verschiedene Vorzeichen hat.

Daher liegt für stetiges f zwischen x und y garantiert eine Nullstelle!

Setze $z := (x+y)/2$ den Wert genau in der Mitte zwischen x und y :

Berechne $f(z)$.

Ist $f(z) = 0$: fertig

Ist $f(x) \cdot f(z) > 0$, so setze $x := z$, y bleibt

Ist $f(y) \cdot f(z) > 0$, so setze $y := z$, x bleibt

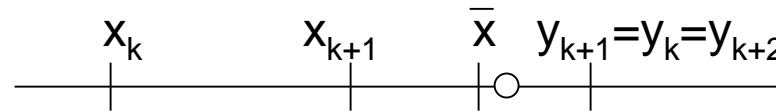
Damit gilt wieder $f(x) \cdot f(y) < 0$, aber der Abstand zwischen x und y hat sich halbiert.

Daher liegt die gesuchte Nullstelle nach k Schritten garantiert in einem kleineren Intervall der Größe $\text{const}/2^k$, und es gilt:
(Bezeichnung: $x_k, y_k \rightarrow z_k$ usw.)

$$\begin{aligned}
 \underline{|\bar{x} - x_{k+1}| + |y_{k+1} - \bar{x}|} &= |y_{k+1} - x_{k+1}| = \\
 &= 0.5 \cdot |y_k - x_k| = 0.5 \cdot \left(\underline{|\bar{x} - x_k| + |y_k - \bar{x}|} \right) \\
 &= |y_k - x_k| / 2
 \end{aligned}$$

Daher schrumpft der „**Abstand**“ der Nullstelle zu den beiden Intervallgrenzen linear in jedem Schritt um den Faktor 0.5

$$\{|x_{k+2} - \bar{x}| + |y_{k+2} - \bar{x}|\} = |x_{k+2} - y_{k+2}| = |x_{k+1} - y_{k+1}| / 2 = \{|x_{k+1} - \bar{x}| + |y_{k+1} - \bar{x}|\} / 2$$



Regula falsi durch Verbindung von Bisektion und Sekantenverf.:

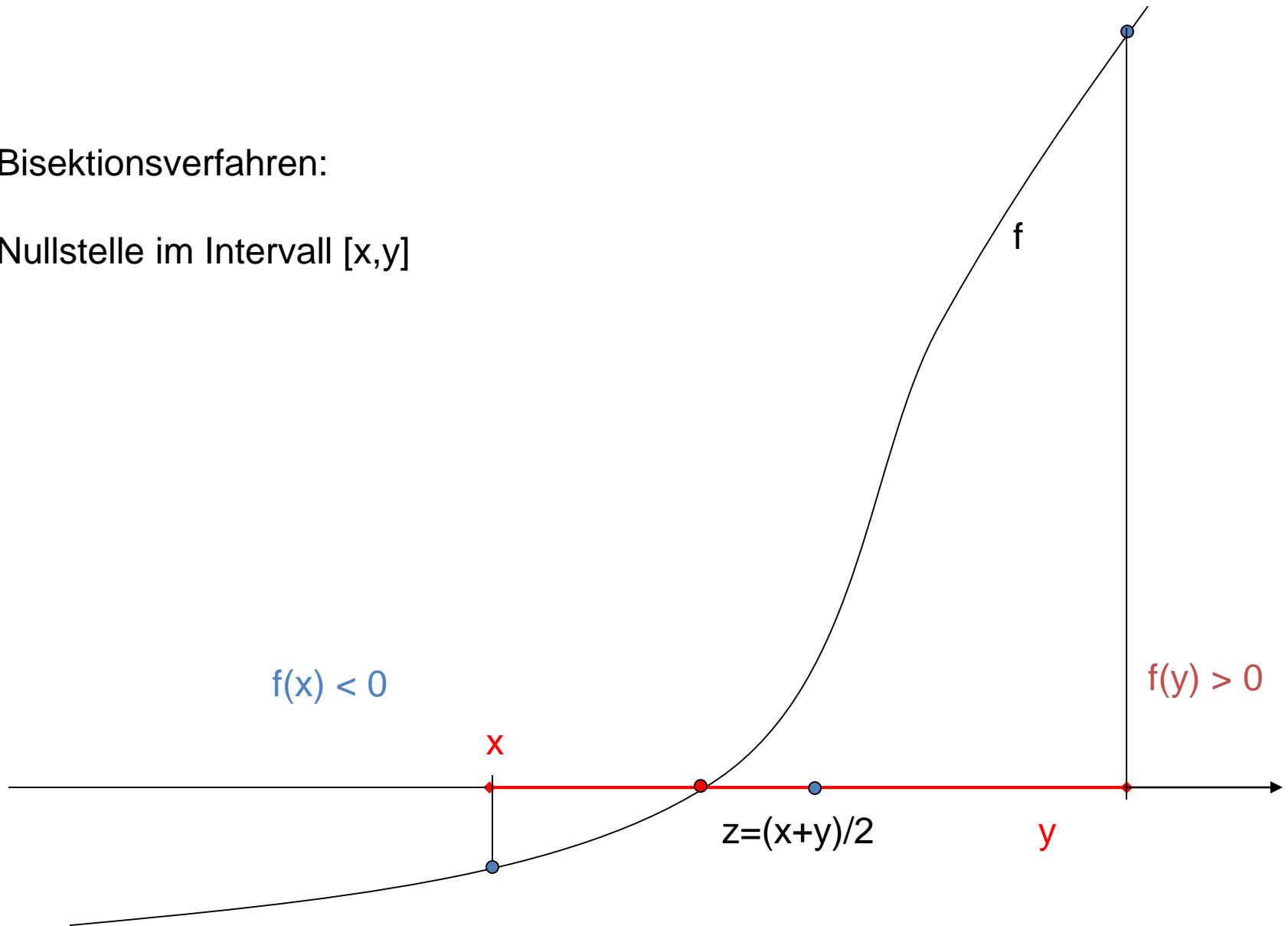
Iteriere wieder Intervall-Einschließungsgrenzen.

Neuer Kandidat für Ober/Untergrenze ist jetzt nicht der Punkt in der Mitte, sondern die Nullstelle der Sekante.

Ersetze eine der beiden alten Grenzen durch diesen neuen Kandidaten, so dass die gesuchte Nullstelle wieder garantiert in dem neuen Intervall liegt!

Bisektionsverfahren:

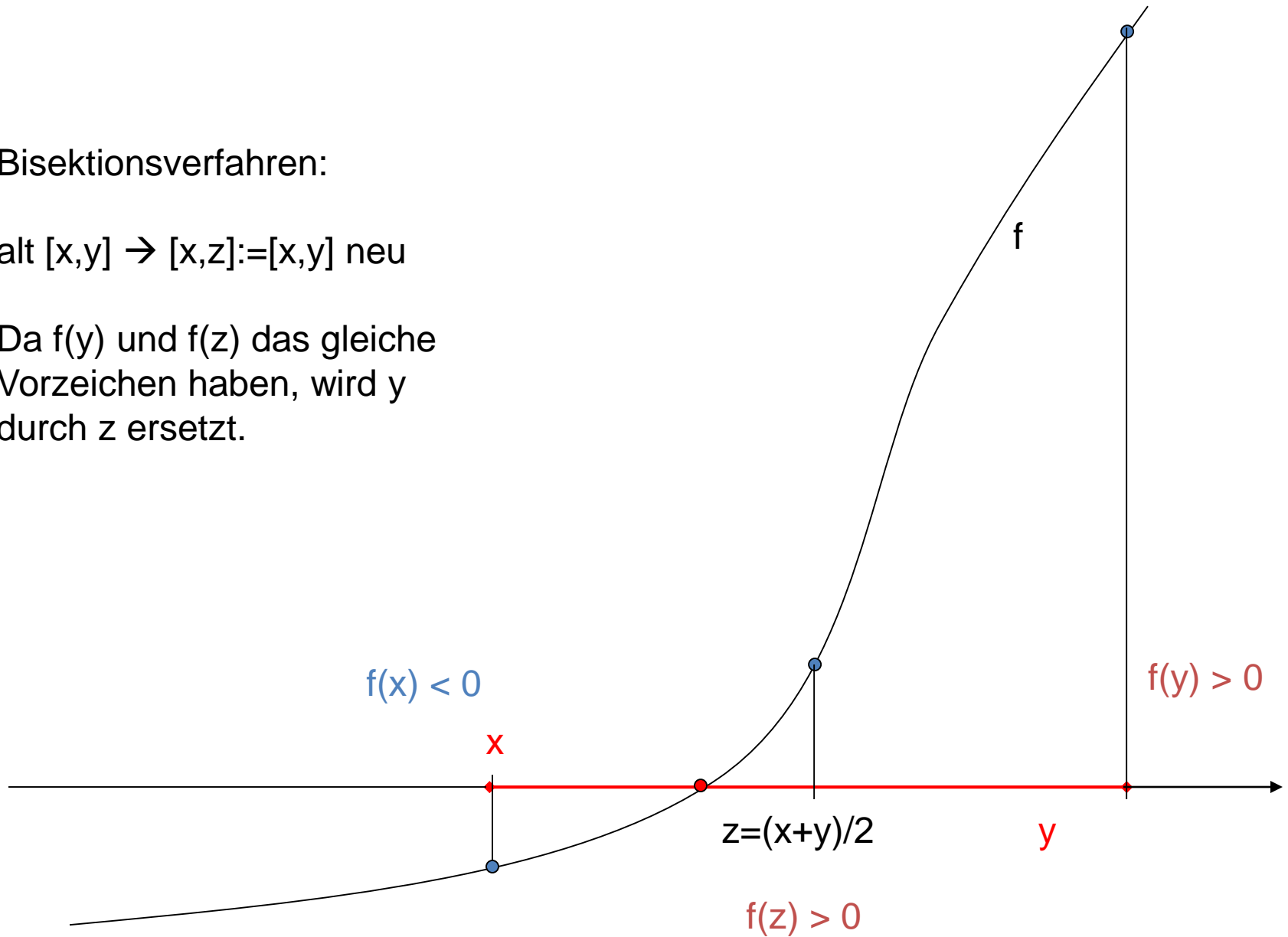
Nullstelle im Intervall $[x,y]$

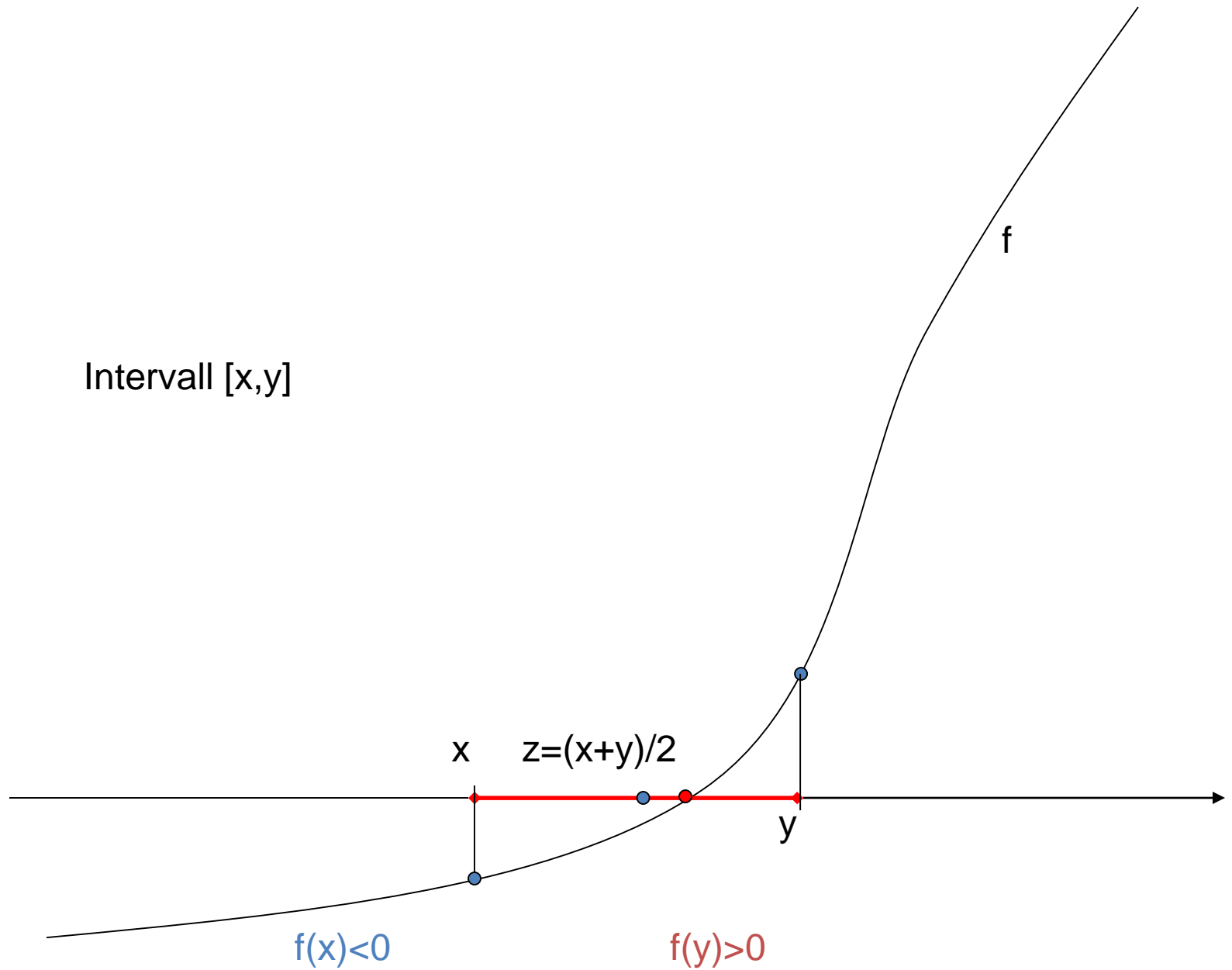


Bisektionsverfahren:

alt $[x,y] \rightarrow [x,z]:=[x,y]$ neu

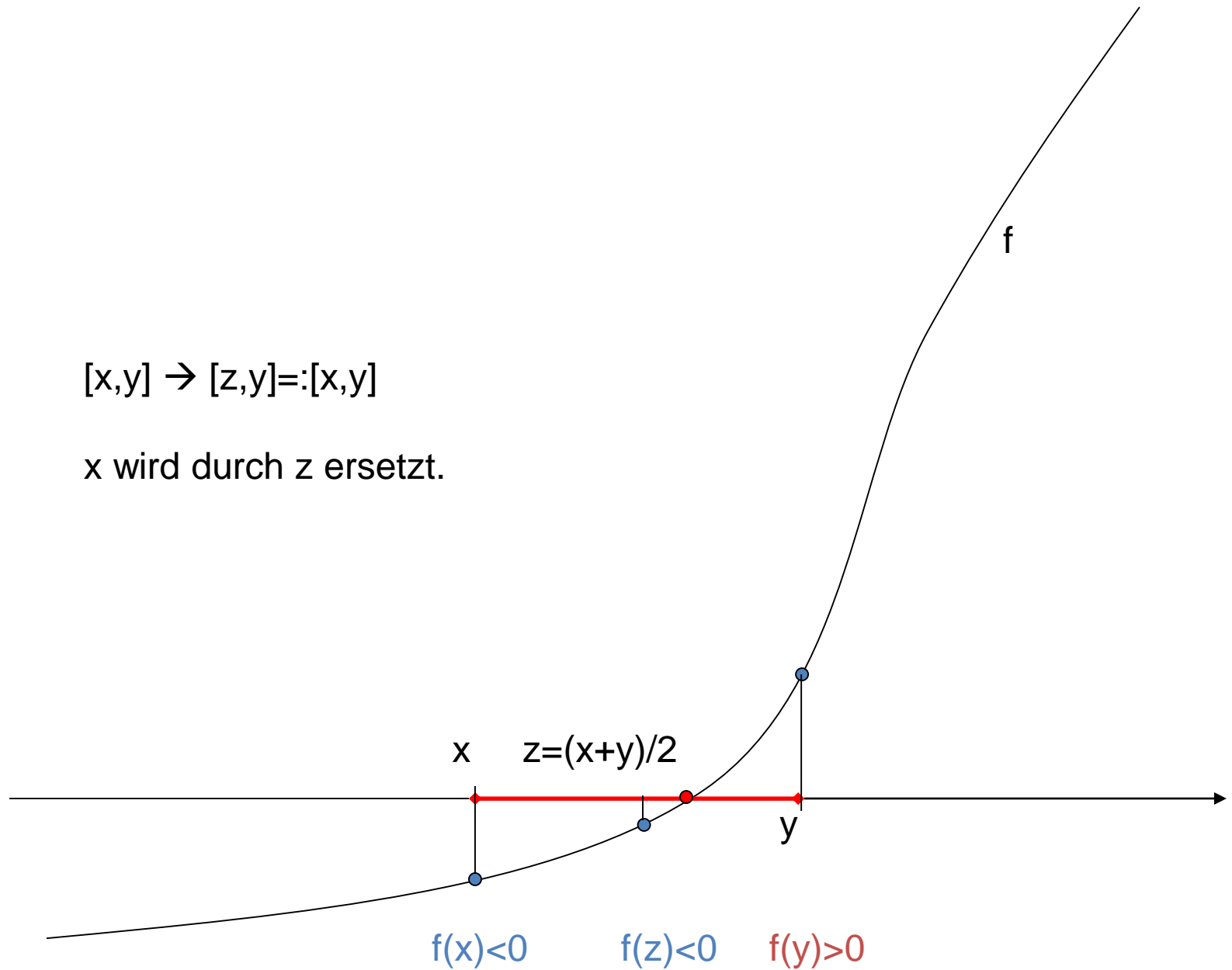
Da $f(y)$ und $f(z)$ das gleiche Vorzeichen haben, wird y durch z ersetzt.



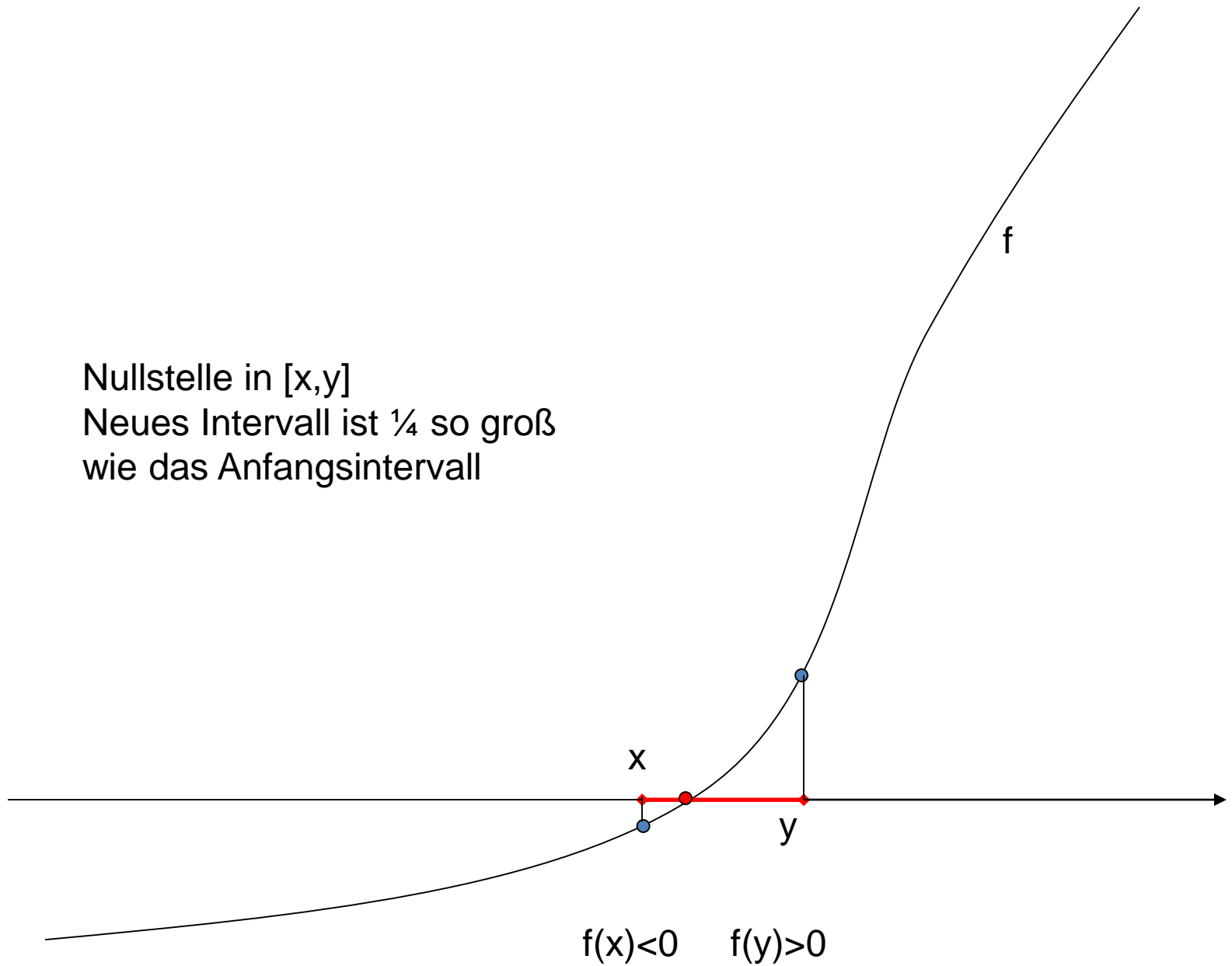


$[x,y] \rightarrow [z,y]=:[x,y]$

x wird durch z ersetzt.



Nullstelle in $[x,y]$
Neues Intervall ist $\frac{1}{4}$ so groß
wie das Anfangsintervall



In der Praxis wird häufig eine Kombination von Bisektion und Newton eingesetzt:

Starte mit ‚sicherer‘ Bisektion, bis der ‚Einzugsbereich‘ der quadratischen Konvergenz des Newtonverfahrens erreicht ist.

Dazu nötig sind Werte x und y mit $f(x)f(y) < 0$;
solche Werte erhält man z.B. durch Auswertung von $f(x)$ an
Zufallszahlen oder an äquidistanten Stellen.

Falls solche x und y nicht auffindbar sind, liegt ev. keine
Nullstelle vor oder eine Nullstelle gerader Ordnung!
Im letzteren Fall kann das Newtonverfahren auf $f'(x)$
angewendet werden.

6.2.5. Newtonverfahren und Minimierung:

Nullstelle einer Funktion $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$,

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

Jacobi-Matrix der Ableitungen von f : $J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$

Newtonverfahren im \mathbb{R}^n : $x^{k+1} := x^k - \text{inv}(J) \cdot f(x^k)$

Vektor = Vektor – Matrix * Vektor

In jedem Schritt ist ein neues Gleichungss. in $J(x^k)$ zu lösen!

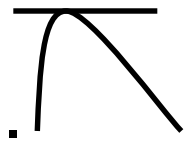
6.2.6. Minimierungsproblem:



Gegeben: $F: \mathbb{R}^n \rightarrow \mathbb{R}$, $F(x_1, \dots, x_n) \in \mathbb{R}$

Gesucht: (relatives) Minimum / Maximum

Bestimme dazu Nullstelle der Ableitung
(entspricht waagrechter Tangente, also Extremwert).



Gradientenvektor von F :

$$\nabla F = f(x_1, \dots, x_n) = \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_n(x_1, \dots, x_n) \end{pmatrix} = \begin{pmatrix} \frac{\partial F}{\partial x_1} \\ \vdots \\ \frac{\partial F}{\partial x_n} \end{pmatrix} \stackrel{!}{=} 0$$

Jacobi-matrix J der ersten Ableitungen von f entspricht der sog. Hesse-matrix H der zweiten Ableitungen von F :

$$H(F) = J(f) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} = \left(\frac{\partial^2 F}{\partial x_i \partial x_j} \right)_{i,j=1}^n$$

Newtonverfahren zur Bestimmung der Nullstelle des Gradienten ergibt:

$$x^{k+1} = x^k - \text{inv}(H) \cdot \nabla F(x^k)$$

Löse dazu in jedem Schritt ein lineares Gleichungssystem mit sog. Hesse-matrix $H(x^k)$!

Problem, falls H an der Stelle x^k singulär.

Billiger: Quasi-Newton-Verfahren, ersetze H durch billiges B .

6.2.7. Nichtlineare Ausgleichsrechnung: Gauss-Newton-Verfahren

$$\min_x \left\| \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_m(x_1, \dots, x_n) \end{pmatrix} - b \right\|_2, \quad (\text{vgl. } \min \|Ax-b\|)$$

Erster Schritt:

Linearisierung mittels Jacobi-matrix und Taylorreihe

$$f(x) = f(x^k) + J(x - x^k) + \cancel{O(\|h\|^2)}$$

Damit ergibt sich neue Iterierte x^{k+1} aus der Lösung des linearen Ausgleichsproblems

$$\min_x \left\| \left(f(x^k) + J^k (x - x^k) \right) - b \right\|_2$$

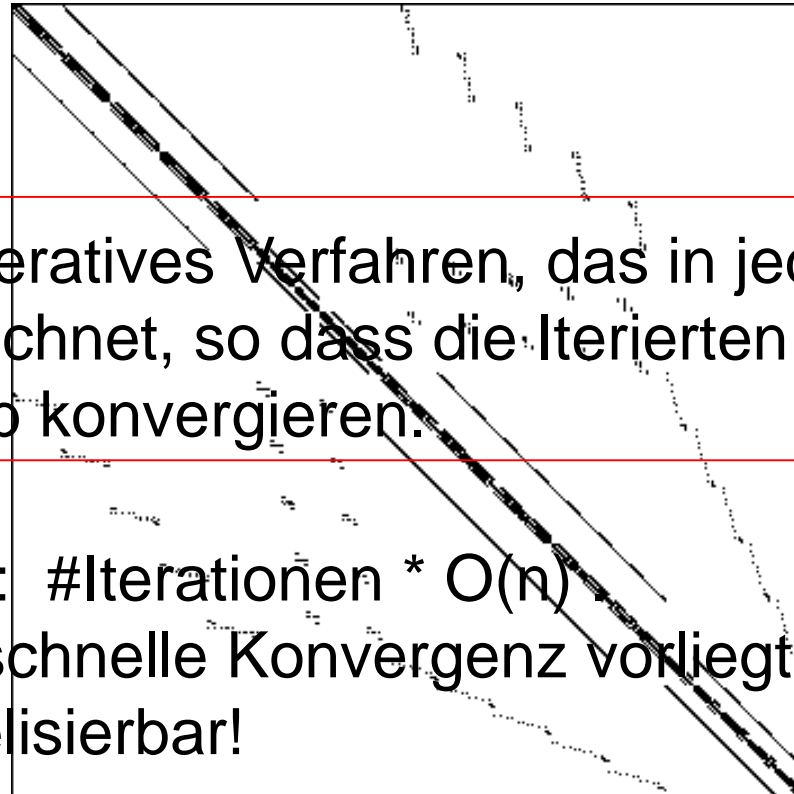
Neue Linearisierung an der Stelle x^{k+1} ergibt neues lineares Ausgleichsproblem mit neuer Matrix J^{k+1} .

6.3. Iterative Lösung linearer Gleichungssysteme

Großes lineares dünnbesetztes Gleichungssystem $Ax = b$

Gauss-Elimination nutzt in der Regel die Dünnbesetztheit nicht aus und führt meist auf Kosten $O(n^3)$;

Im Gegensatz dazu ist oft nur $O(n)$ Speicherbedarf für Matrix A



Idee: Formuliere iteratives Verfahren, das in jedem Schritt nur $\text{Matrix} \cdot \text{Vektor}$ berechnet, so dass die Iterierten gegen die Lösung von $Ax = b$ konvergieren.

Gesamtkosten: $\# \text{Iterationen} * O(n)$
Effizient, falls schnelle Konvergenz vorliegt!
Einfach parallelisierbar!

6.3.1. Stationäre Methoden:

Richardson-Verfahren:

Formulierung eines passenden Fixpunktproblems

$$b = Ax = (A - I)x + x \Leftrightarrow x = b + (I - A)x =: \Phi(x)$$

Daraus ergibt sich die Iteration:

$$x_{k+1} = \Phi(x_k) = b + (I - A)x_k = x_k + (b - Ax_k) = x_k + r_k$$

mit Residuum r_k .

Frage:

Wann konvergiert die ausgehend von einem x_0 definierte Folge x_k gegen die gesuchte Lösung $\bar{x} = A^{-1}b$?

Betrachte dazu den Fehler im k-ten Schritt:

$$\bar{x} - x_{k+1} = \bar{x} - x_k - (b - Ax_k) = \bar{x} - x_k - (A\bar{x} - Ax_k) = (I - A)(\bar{x} - x_k)$$

Ergibt in der Norm $\|\bar{x} - x_{k+1}\|_2 = \|(I - A)(\bar{x} - x_k)\|_2 \leq \|I - A\|_2 \cdot \|\bar{x} - x_k\|_2$

und daher $\|\bar{x} - x_k\|_2 \leq \|I - A\|_2^k \cdot \|\bar{x} - x_0\|_2$

Daher liegt Konvergenz vor für $\|I - A\|_2 < 1$.

Dies entspricht der Kontraktionsbedingung für die Iterationsfunktion $\Phi(x)$:

$$\|\Phi(x) - \Phi(y)\|_2 \leq \|I - A\|_2 \cdot \|x - y\|_2$$

Richardsonverfahren ist daher nur sinnvoll, wenn $A \approx I$!

Wende das Verfahren auf das modifizierte Problem an:

$$D = \text{diag}(A); \quad \boxed{(D^{-1}A)x = (D^{-1}b)} \quad \Leftrightarrow \quad Ax = b$$

Die Bedingung $D^{-1}A \approx I$ ist besser erfüllt!

Bezeichnung: $\tilde{A} = D^{-1}A$ und $\tilde{b} = D^{-1}b$

ergibt neues Gleichungssystem $\tilde{A}x = \tilde{b}$

Richardson für das tilde-System liefert dann:

$$x_{k+1} = x_k + (\tilde{b} - \tilde{A}x_k) = x_k + D^{-1}(b - Ax_k) \quad \text{oder}$$

$$Dx_{k+1} = Dx_k + (b - Ax_k) = b + (D - A)x_k$$

Dies ist das Jacobiverfahren zur iterativen Lösung von $Ax = b$

Wesentliche Kosten in jedem Schritt: Ax_k

Konvergent, falls $\|I - D^{-1}A\|_2 < 1$

Notwendig: Diagonalmatrix D regulär!

Allgemeinere Idee zur Formulierung einer Iterationsfunktion:

Matrix-splitting

L : Unterdiagonalteil von A

U : Oberdiagonalteil von A

$$b = Ax = (L + D + U)x = (L + D)x + Ux$$

führt auf Iteration

$$\begin{aligned} (L + D)x_{k+1} &= b - Ux_k = b - (A - L - D)x_k \\ &= (L + D)x_k + (b - Ax_k) \end{aligned}$$

Dies ist das Gauss-Seidel-Verfahren

$$\begin{aligned}
 x_{k+1} &= x_k + (L + D)^{-1} (b - Ax_k) \\
 &= (L + D)^{-1} b + \left(I - (L + D)^{-1} A \right) x_k \\
 &= (L + D)^{-1} b - (L + D)^{-1} Ux
 \end{aligned}$$

In jedem Schritt ist dabei nur ein Dreiecksgleichungssystem zu lösen.

Das Verfahren ist konvergent, falls $\|I - (L + D)^{-1} A\|_2 < 1$

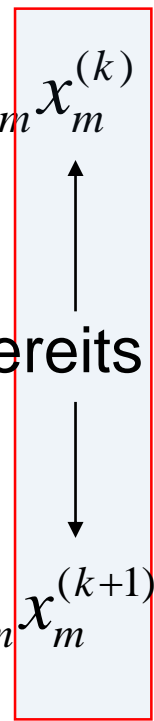
Gauss-Seidel ist äquivalent zu Richardson angewendet auf

$$\left((L + D)^{-1} A \right) x = \left((L + D)^{-1} b \right) \quad \Leftrightarrow \quad Ax = b$$

Gauss-Seidel-Iteration

Versuche immer, die neueste Information zu verwenden!

Jacobi Iteration:

$$a_{jj}x_j^{(k+1)} = b_j - \sum_{m=1}^{j-1} a_{jm}x_m^{(k)} - \sum_{m=j+1}^n a_{jm}x_m^{(k)}$$


Gauss-Seidel Iteration: bereits berechnet

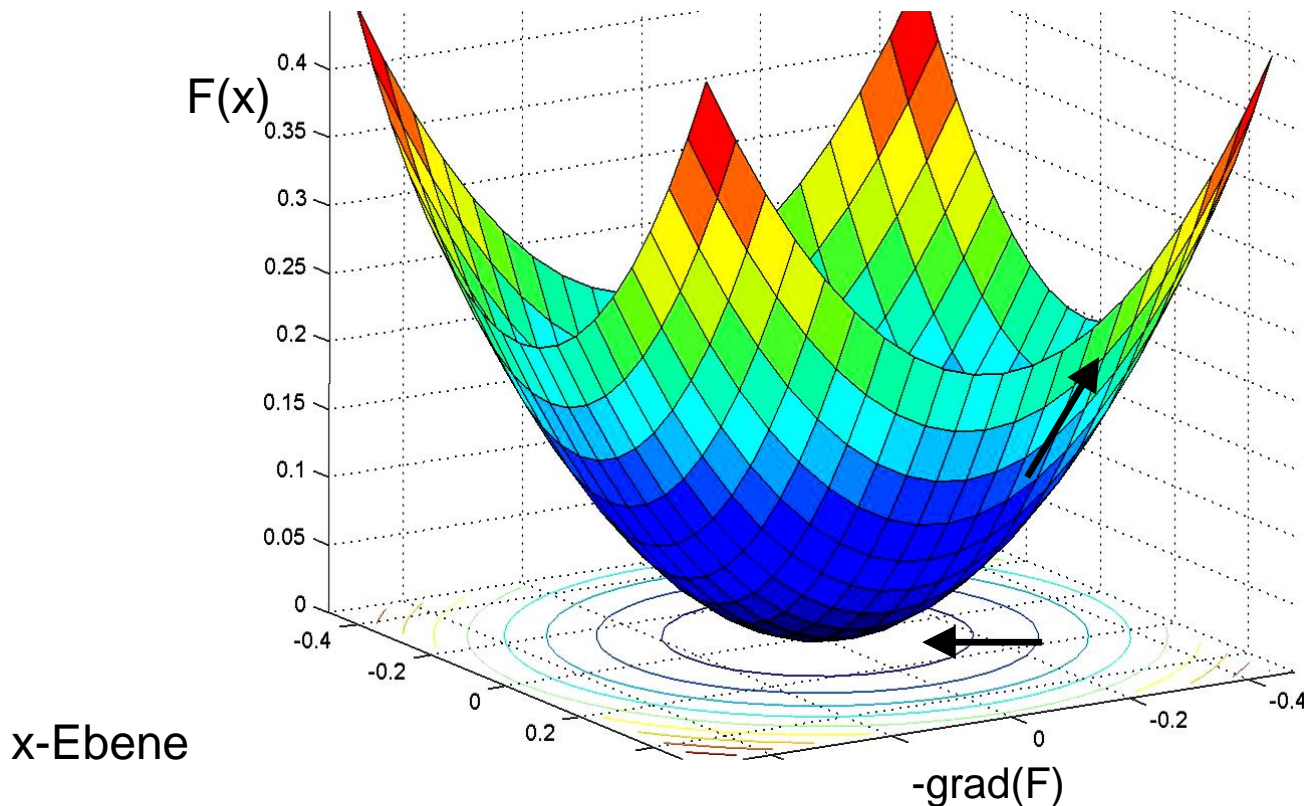
$$a_{jj}x_j^{(k+1)} = b_j - \sum_{m=1}^{j-1} a_{jm}x_m^{(k+1)} - \sum_{m=j+1}^n a_{jm}x_m^{(k)}$$

6.3.2. Nichtstationäre Verfahren

Das Gradientenverfahren für symmetrisch positiv definite Matrix A:

Betrachte Funktion $F(x): \mathbb{R}^n \rightarrow \mathbb{R}$,
$$F(x) = \frac{1}{2} x^T A x - b^T x$$

F beschreibt einen Paraboloid im \mathbb{R}^n :



Minimum dieser Funktion ist wieder der Punkt mit waagrechter Tangente, also die Stelle mit Gradient gleich Null:

$$\nabla F(x) = Ax - b = 0 \Leftrightarrow Ax = b$$

Stelle x , an der der Paraboloid sein Minimum annimmt ist gleich der gesuchten Lösung des Gleichungssystems!

Betrachte Minimierungsaufgabe!

Von aktueller Stelle x_k aus soll die nächste Iterierte x_{k+1} so gewählt werden, dass sie näher am Minimum liegt.

$$x_{k+1} = x_k + \alpha_k d_k$$

mit Suchrichtung d_k und Schrittweite α_k .

Finde Suchrichtung so, dass Funktionswerte kleiner werden:

Abstiegsrichtung ist gegeben durch Richtung des negativen Gradienten!

Denn Richtungsableitung in Richtung n ist gleich $\nabla F \cdot n$, und wird am betragsgrößten für $n = \nabla F$ nämlich $\|\nabla F\|^2$

Daher ist $n = -\nabla F$ lokal die Richtung des steilsten Abstiegs zum Minimum.

Daher verkleinern sich die Funktionswerte auf jeden Fall, wenn man in diese Abstiegsrichtung geht.

Also wähle

$$x_{k+1} = x_k - \alpha_k \nabla F(x_k) = x_k + \alpha_k (b - Ax_k)$$

Noch zu bestimmen ist optimale Schrittweite α_k , die am nächsten ans Minimum führt!

Betrachte dazu ein-dimensionale Minimierung:

$$\begin{aligned} \min_{\alpha} g(\alpha) &:= \min_{\alpha} \left(F(x_k + \alpha(b - Ax_k)) \right) = \\ \min_{\alpha} &\left(\frac{1}{2} (x_k + \alpha d_k)^T A (x_k + \alpha d_k) - b^T (x_k + \alpha d_k) \right) = \\ \min_{\alpha} &\left(\frac{1}{2} \alpha^2 d_k^T A d_k - \alpha d_k^T d_k + \frac{1}{2} x_k^T A x_k - x_k^T b \right) \end{aligned}$$

mit Lösung $\alpha_k = d_k^T d_k / d_k^T A d_k$, $d_k := b - Ax_k$

Insgesamt:

$$x_{k+1} = x_k + \frac{\|b - Ax_k\|_2^2}{(b - Ax_k)^T A (b - Ax_k)} \cdot (b - Ax_k)$$

Dazugehörige Fixpunktgleichung ist

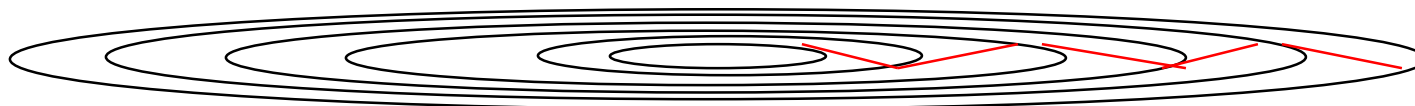
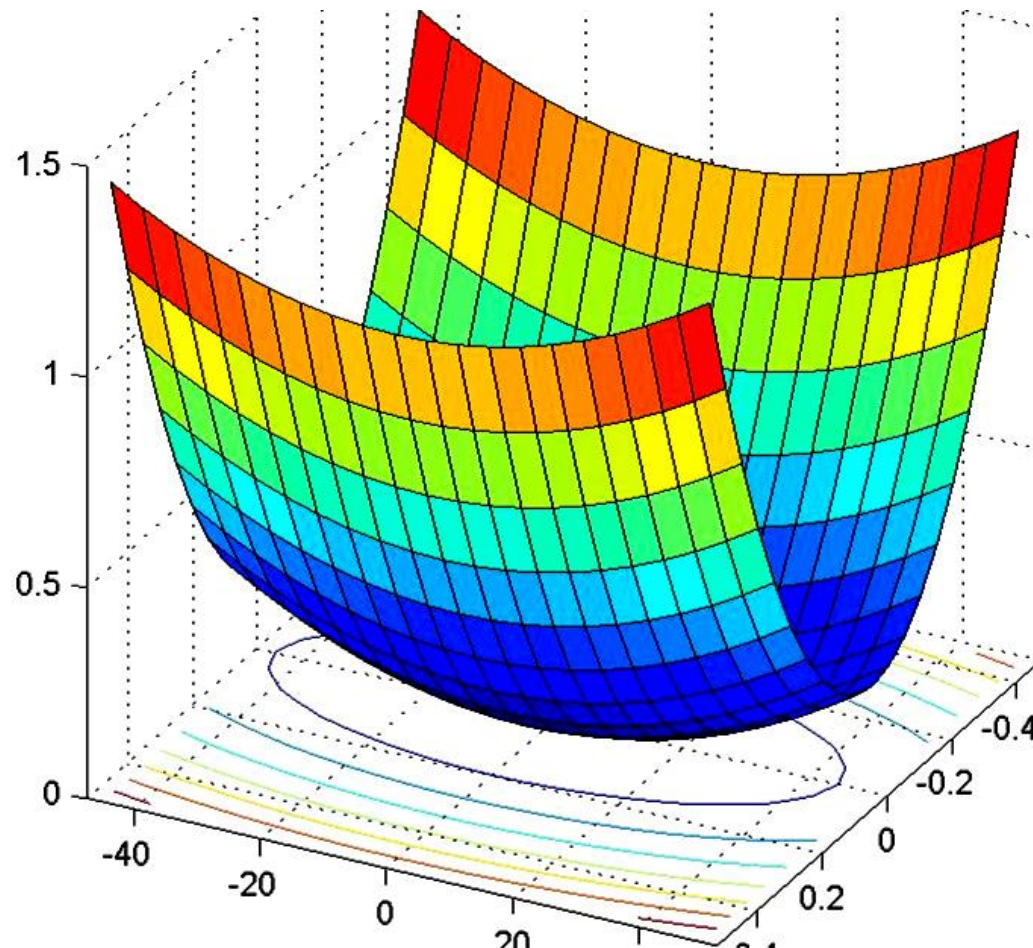
$$x = \Phi(x) := x + \frac{\|b - Ax\|_2^2}{(b - Ax)^T A(b - Ax)} \cdot (b - Ax)$$

mit Fixpunkt $\bar{x} = A^{-1}b$

Ergibt Verfahren des steilsten Abstiegs („steepest descent“) oder Gradientenverfahren

Nachteil:

Bei stark verzerrtem Paraboloiden ergibt sich sehr langsame Konvergenz.



Für $A \approx I$ ist der Paraboloid unverzerrt, die Höhenlinien fast kreisförmig \rightarrow schnelle Konvergenz!

Daher versucht man, $Ax=b$ zu präkonditionieren:
Ersetze $Ax=b$ durch $M^{-1}Ax=M^{-1}b$ mit $M^{-1}A \approx I$

Bessere Variante eines Gradientenverfahren:
Verfahren der konjugierten Gradienten, kurz cg-Verfahren.

Suchrichtung nicht der negative Gradient selbst, sondern die Projektion des Gradienten, so dass alle Suchrichtungen in gewisser Weise orthogonal zueinander sind

Genauer: Suchrichtungen seien A -konjugiert, d.h.

$$d_k^T A d_j = 0 \quad \text{für } j \neq k$$

Damit ergibt sich iteratives Verfahren, das nach k Schritten jeweils die beste Näherung an die Lösung in einem k -dimensionalen Unterraum liefert, und daher nach n Schritten fertig ist (in exakter Arithmetik).

Conjugate Gradient Method

Bessere Abstiegsrichtung, global optimal:

$x_{k+1} := x_k + \alpha_k p_k$ als neue Suchrichtung an Stelle des Gradienten
Projektion des Gradienten,
so dass A-conjugate (orthogonal) zu allen
vorherigen Suchrichtungen:

$p_k \perp A p_j$ for all $j < k$ oder $p_k \perp_A p_j$ oder $p_k^T A p_j = 0$ für $j < k$

α_k wieder durch 1-dimensionale Minimierung wie vorher
(für verwendetes p_k)

Conjugate Gradient Algorithm

$$x_0=0; r_0 = b - A x_0 ;$$

$$\text{For } k=1,2,\dots: \quad \beta_{k-1} = r_{k-1}^T r_{k-1} / r_{k-2}^T r_{k-2} ; \quad (\beta_0=0)$$

$$p_k = r_{k-1} + \beta_{k-1} p_{k-1};$$

$$a_k = r_{k-1}^T r_{k-1} / p_k^T A p_k ;$$

$$x_k = x_{k-1} + a p_k ;$$

$$r_k = r_{k-1} - a_k A p_k ;$$

if $\| r_k \| < \varepsilon$: STOP

Eigenschaften des cg-Verfahrens:

Bestimme in Unterraum $(b, Ab, A^2b, \dots, A^k b)$ die beste Lösung mit minimalem Fehler: Optimal! Daher nach n Schritten fertig!?

Algorithmus ist billig – in jedem Schritt nur Ax .

In dieser Form nur möglich für symmetrisch positiv definites A .
Für allgemeine Matrix:

GMRES (optimal, aber teurer)

BiCG (nicht optimal, aber billig)

Deep Learning für Neuronale Netze: Gradientenverfahren

Wichtig: Präkonditionierung $P^*Ax = P^*b$, Jacobi, Gauss-Seidel

6.4. Eigenwerte und Vektoriteration



Eigenvektor $v \neq 0$ und Eigenwert λ einer quadratischen Matrix A erfüllen die Gleichung

$$A v = \lambda v$$

Daher ist die durch den Vektor v bestimmte Gerade durch den Nullpunkt eine sog. Fixgerade der durch

$$x \rightarrow A x$$

definierten Abbildung.

Für eine symmetrische Matrix A gilt:

Die Eigenvektoren der n Eigenwerte von A bilden eine Orthonormalbasis des \mathbb{R}^n .

Eigenvektormatrix V liefert eine Diagonalisierung der Matrix A . Die durch A definierte Abbildung wird in dieser Basis trivial:

$$A v = \lambda v \quad \rightarrow \quad A V = V \Lambda,$$

Λ ist Diagonalmatrix, mit den Eigenwerten als Diagonaleinträge

$$V^T A V = \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$$

Ein Eigenwert erfüllt die Gleichung $\det(A - \lambda I) = 0$,
 da $A - \lambda I$ singularär: $(A - \lambda I) v = 0$

Also sind Eigenwerte genau die Nullstellen des *charakteristischen Polynoms* $p(\lambda) := \det(A - \lambda I)$.

Eigenwerte von $A \leftrightarrow$ Nullstellen eines Polynoms $p(x)$

Die Matrix $A :=$
$$\begin{pmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \ddots & \vdots & -a_1 \\ 0 & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & 1 & 0 & -a_{n-2} \\ 0 & \cdots & 0 & 1 & -a_{n-1} \end{pmatrix}$$

hat das charakteristisches Polynom

$$p(x) = (-1)^n (a_0 + a_1 x + \dots + a_{n-1} x^{n-1} + x^n)$$

Daher lassen sich Nullstellen von Polynomen berechnen, indem man die Eigenwerte der obigen Matrix A berechnet!

Eigenwertberechnung ist numerisch stabiler als Nullstellenberechnung bei Polynomen!

Vektoriteration

Vektoriteration ist eine einfache Fixpunktiteration zur Berechnung des betragsgrößten Eigenwerts einer Matrix:

Sei A symmetrisch positiv definit ($x \neq 0 \rightarrow x^T A x > 0$);

$$\text{Start mit } x_0 \neq 0, \quad x_{k+1} := \frac{Ax_k}{\|Ax_k\|};$$

Die Iterationsfunktion: $\Phi(x) = \frac{Ax}{\|Ax\|}$

hat Fixpunkt v , wenn $v = \frac{Av}{\|Av\|}$

d.h. v ist Eigenvektor von A zu Eigenwert $\lambda = \|Av\|$,
denn dann gilt: $Av = \|Av\| \cdot v$.

Offensichtlich ist dann $x_k = \text{const} * A^k x_0$

Außerdem kann x_0 in der Basis der Eigenvektoren dargestellt werden:

$$x_0 = c_1 v_1 + \dots + c_m v_m$$

wobei λ_1 maximaler Eigenwert von A zu Eigenvektor v_1 ist.

Daher gilt $Av_j = \lambda_j v_j$ und

$$A^k x_0 = c_1 A^k v_1 + \dots + c_m A^k v_m =$$

$$= c_1 \lambda_1^k v_1 + \dots + c_m \lambda_m^k v_m =$$

$$= \lambda_1^k (c_1 v_1 + \dots + c_m \underbrace{(\lambda_m / \lambda_1)^k}_{\rightarrow 0} v_m) \xrightarrow{k \rightarrow \infty} \lambda_1^k (c_1 v_1)$$

Normiert man, dann folgt $\frac{A^k x_0}{\|A^k x_0\|} \rightarrow \frac{\lambda_1^k (c_1 v_1)}{\|\lambda_1^k (c_1 v_1)\|} = \frac{c_1}{|c_1|} v_1 = \pm v_1$

x_k ist normierter Vektor zu $A^k x_0$.

Dann bleibt in x_k nur die Komponente zum stärksten Eigenwert übrig, nämlich v_1 .

Also $x_k \rightarrow \text{const} \cdot v_1$ konvergiert gegen Eigenvektor,
und daher auch

$\|Ax_k\| \rightarrow \lambda_1$ konvergiert gegen größten Eigenwert.

Ähnliches gilt für allgemeine Matrizen, solange die Matrix einen eindeutigen betragsgrößten Eigenwert hat

(also z.B. nicht : $\pm\lambda_1$ sind beide Eigenwerte)

Beispiel:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1/2 \end{pmatrix}^k \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2^{-k} \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

ist Eigenvektor zu Eigenwert $\lambda = 1$

Beispiel:

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}^k \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ (-1)^k \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ \pm 1 \end{pmatrix}$$

Im Eigenraum zu Eigenwerten $\lambda = 1$ und -1

Für innere Eigenwerte:

Wende Vektoriteration auf die Matrix $(A-\sigma I)^{-1}$ an.

Liefert deren größten Eigenwert.

Dies ist der Eigenwert von A , der am nächsten bei σ liegt.

Dies ist die sog. Inverse Iteration.

Problem: In jedem Schritt ist ein Gleichungssystem zu lösen
 $(A-\sigma I)$ schlecht konditioniert, ev. singular!

Anwendungen



Resonanzen, siehe Tacoma, London Millenium Bridge

Energieniveaus in der Quantenmechanik

Nullstellen von Polynomen

Biegen eines Balkens - Hauptträgheitsachsen

Stabilität eines Systems, Differentialgleichungen

Modellreduktion, Principal Component Analysis PCA,
Eigenface, finger print

Analyse von Graphen, Pagerank

Unterschiedliche Aufgabestellung:

Alle Eigenwerte/vektoren oder nur einige!

Eigenwerte alleine oder mit Eigenvektoren