Lösung positiver polynomieller Gleichungssysteme

Stefan Kiefer

Technische Universität München

19. Oktober 2009

Positive Polynomielle Gleichungssysteme

Wir betrachten Gleichungssysteme der Form

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}) \, ,$$

dabei ist

- $\boldsymbol{x}$ ein Vektor aus n Variablen,
- $\boldsymbol{f}(\boldsymbol{x})$ ein Vektor aus Polynomen mit positiven Koeffizienten.

Beispiel ($n = 2$)

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} 1.2 \cdot X \cdot Y + 0.3 \cdot Y^2 \\ 0.7 \cdot X^2 + 0.2 \end{pmatrix}$$

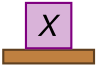
Wir nennen solche Gleichungssysteme
positive polynomielle Gleichungssysteme.

Im Vortrag angenommen: es gibt eine nichtnegative Lösung.
Dann gibt es eine kleinste, genannt $\mu\boldsymbol{f}$.
Wir möchten $\mu\boldsymbol{f}$ berechnen oder approximieren.

Probabilistische Rekursive Programme

Einfaches Modell für probabilistische rekursive Programme:

$$
\begin{array}{lll}
X \xhookrightarrow{0.4} \text{call } Y & Y \xhookrightarrow{0.3} \text{call } X & Z \xhookrightarrow{0.3} \text{call } X \\
X \xhookrightarrow{0.6} \text{return} & Y \xhookrightarrow{0.4} \text{call } Z & Z \xhookrightarrow{0.7} \text{return} \\
 & Y \xhookrightarrow{0.3} \text{return} &
\end{array}
$$



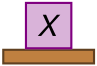
Probabilistische Rekursive Programme

Einfaches Modell für probabilistische rekursive Programme:

$$X \xhookrightarrow{0.4} \text{call } Y \qquad Y \xhookrightarrow{0.3} \text{call } X \qquad Z \xhookrightarrow{0.3} \text{call } X$$

$$X \xhookrightarrow{0.6} \text{return} \qquad Y \xhookrightarrow{0.4} \text{call } Z \qquad Z \xhookrightarrow{0.7} \text{return}$$

$$Y \xhookrightarrow{0.3} \text{return}$$



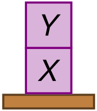
Probabilistische Rekursive Programme

Einfaches Modell für probabilistische rekursive Programme:

$$X \xhookrightarrow{0.4} \text{call } Y \qquad Y \xhookrightarrow{0.3} \text{call } X \qquad Z \xhookrightarrow{0.3} \text{call } X$$

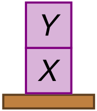
$$X \xhookrightarrow{0.6} \text{return} \qquad Y \xhookrightarrow{0.4} \text{call } Z \qquad Z \xhookrightarrow{0.7} \text{return}$$

$$Y \xhookrightarrow{0.3} \text{return}$$



Einfaches Modell für probabilistische rekursive Programme:

$X \xhookrightarrow{0.4} \text{call } Y$	$Y \xhookrightarrow{0.3} \text{call } X$	$Z \xhookrightarrow{0.3} \text{call } X$
$X \xhookrightarrow{0.6} \text{return}$	$Y \xhookrightarrow{0.4} \text{call } Z$	$Z \xhookrightarrow{0.7} \text{return}$
	$Y \xhookrightarrow{0.3} \text{return}$	

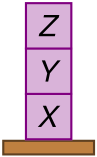


Einfaches Modell für probabilistische rekursive Programme:

$$X \xhookrightarrow{0.4} \text{call } Y \qquad Y \xhookrightarrow{0.3} \text{call } X \qquad Z \xhookrightarrow{0.3} \text{call } X$$

$$X \xhookrightarrow{0.6} \text{return} \qquad Y \xhookrightarrow{0.4} \text{call } Z \qquad Z \xhookrightarrow{0.7} \text{return}$$

$$Y \xhookrightarrow{0.3} \text{return}$$



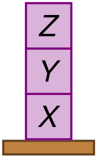
Probabilistische Rekursive Programme

Einfaches Modell für probabilistische rekursive Programme:

$$X \xhookrightarrow{0.4} \text{call } Y \qquad Y \xhookrightarrow{0.3} \text{call } X \qquad Z \xhookrightarrow{0.3} \text{call } X$$

$$X \xhookrightarrow{0.6} \text{return} \qquad Y \xhookrightarrow{0.4} \text{call } Z \qquad \color{red}{Z \xhookrightarrow{0.7} \text{return}}$$

$$Y \xhookrightarrow{0.3} \text{return}$$

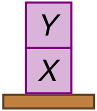


Einfaches Modell für probabilistische rekursive Programme:

$$X \xhookrightarrow{0.4} \text{call } Y \qquad Y \xhookrightarrow{0.3} \text{call } X \qquad Z \xhookrightarrow{0.3} \text{call } X$$

$$X \xhookrightarrow{0.6} \text{return} \qquad Y \xhookrightarrow{0.4} \text{call } Z \qquad Z \xhookrightarrow{0.7} \text{return}$$

$$Y \xhookrightarrow{0.3} \text{return}$$

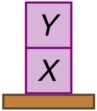


Einfaches Modell für probabilistische rekursive Programme:

$$X \xhookrightarrow{0.4} \text{call } Y \qquad Y \xhookrightarrow{0.3} \text{call } X \qquad Z \xhookrightarrow{0.3} \text{call } X$$

$$X \xhookrightarrow{0.6} \text{return} \qquad Y \xhookrightarrow{0.4} \text{call } Z \qquad Z \xhookrightarrow{0.7} \text{return}$$

$$Y \xhookrightarrow{0.3} \text{return}$$



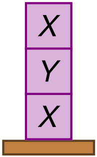
Probabilistische Rekursive Programme

Einfaches Modell für probabilistische rekursive Programme:

$$X \xhookrightarrow{0.4} \text{call } Y \qquad Y \xhookrightarrow{0.3} \text{call } X \qquad Z \xhookrightarrow{0.3} \text{call } X$$

$$X \xhookrightarrow{0.6} \text{return} \qquad Y \xhookrightarrow{0.4} \text{call } Z \qquad Z \xhookrightarrow{0.7} \text{return}$$

$$Y \xhookrightarrow{0.3} \text{return}$$

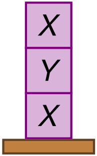


Einfaches Modell für probabilistische rekursive Programme:

$$X \xhookrightarrow{0.4} \text{call } Y \qquad Y \xhookrightarrow{0.3} \text{call } X \qquad Z \xhookrightarrow{0.3} \text{call } X$$

$$X \xhookrightarrow{0.6} \text{return} \qquad Y \xhookrightarrow{0.4} \text{call } Z \qquad Z \xhookrightarrow{0.7} \text{return}$$

$$Y \xhookrightarrow{0.3} \text{return}$$



Terminierungs-Wahrscheinlichkeit?

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4YX + 0.6 \\ 0.3XY + 0.4ZY + 0.3 \\ 0.3XZ + 0.7 \end{pmatrix}$$

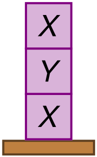
Probabilistische Rekursive Programme

Einfaches Modell für probabilistische rekursive Programme:

$$X \xhookrightarrow{0.4} \text{call } Y \qquad Y \xhookrightarrow{0.3} \text{call } X \qquad Z \xhookrightarrow{0.3} \text{call } X$$

$$X \xhookrightarrow{0.6} \text{return} \qquad Y \xhookrightarrow{0.4} \text{call } Z \qquad Z \xhookrightarrow{0.7} \text{return}$$

$$Y \xhookrightarrow{0.3} \text{return}$$



Terminierungs-Wahrscheinlichkeit?

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4YX + 0.6 \\ 0.3XY + 0.4ZY + 0.3 \\ 0.3XZ + 0.7 \end{pmatrix}$$

Die Terminierungs-W'keit ist die kleinste Lösung.

Nicht unbedingt $(1, 1, 1)$!

Weitere Motivation

- Verifikation probabilistischer rekursiver Programme
  [Esparza, Kučera, Mayr, LICS'04],
  [Etessami, Yannakakis, STACS'05]

  (Terminierungsw'keiten und mehr)
- Analyse stochastischer kontext-freier Grammatiken
  – um die Sekundärstruktur von RNA zu beschreiben
  [Knudsen, Hein, Bioinformatics'99]
  – in der Verarbeitung natürlicher Sprache
  [Wojtczak, Etessami, TACAS'07]
- Stochastische Analyse von Surf-Verhalten
  [Fagin et al., STOC'00]
- Reputationssysteme
  [Bouajjani et al., TACAS'08]

$\Longrightarrow$ All das führt auf das Bestimmen der kleinsten Lösung von $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$.

Approximation ist nötig.

Im Allgemeinen kann $\boldsymbol{\mu f}$ nicht exakt berechnet werden.

Beispiel

Sei $f(X) = \frac{1}{6}X^6 + \frac{1}{2}X^5 + \frac{1}{3}$.
μf kann nicht mit Wurzeln ausgedrückt werden.

$\Longrightarrow$ Approximation nötig
$(0.3357037075 < \mu f < 0.3357037076)$

Ein einfaches Approximationsverfahren

Proposition (Fixpunktsatz von Kleene)

Die Kleene-Folge $\mathbf{0}, \boldsymbol{f}(\mathbf{0}), \boldsymbol{f}(\boldsymbol{f}(\mathbf{0})), \ldots$ *konvergiert zu* $\mu\boldsymbol{f}$.

Ein einfaches Approximationsverfahren

Proposition (Fixpunktsatz von Kleene)

Die Kleene-Folge $\mathbf{0}, \boldsymbol{f}(\mathbf{0}), \boldsymbol{f}(\boldsymbol{f}(\mathbf{0})), \ldots$ *konvergiert zu* $\boldsymbol{\mu f}$.

Aber sehr langsam!

Beispiel

Sei $f(X) = 0.5X^2 + 0.5$.

Dann $\mu f = 1$, aber $f^k(0) \leq 1 - \frac{1}{k+1}$,

$\Longrightarrow$ Man braucht 2^i Iterationen,

um μf auf i Bits genau zu berechnen.

Ein einfaches Approximationsverfahren

Proposition (Fixpunktsatz von Kleene)

Die Kleene-Folge $\mathbf{0}, \boldsymbol{f}(\mathbf{0}), \boldsymbol{f}(\boldsymbol{f}(\mathbf{0})), \ldots$ *konvergiert zu* $\boldsymbol{\mu f}$*.*

Aber sehr langsam!

Beispiel

Sei $f(X) = 0.5X^2 + 0.5$.

Dann $\mu f = 1$, aber $f^k(0) \leq 1 - \frac{1}{k+1}$,

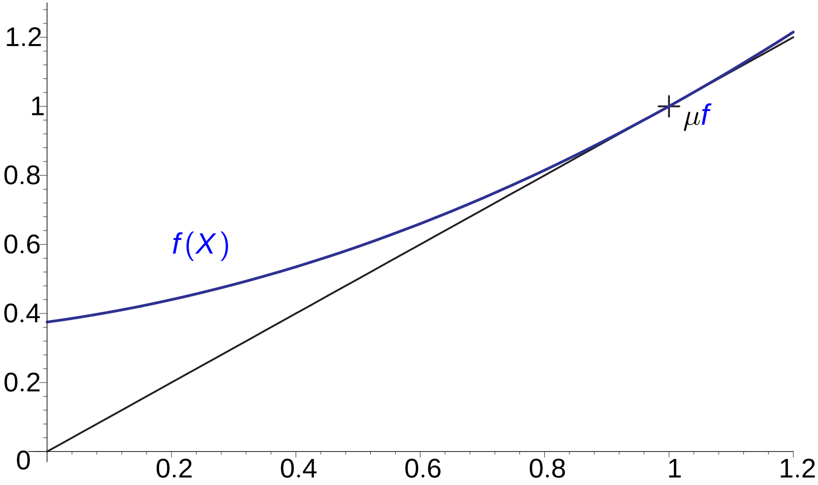
$\Longrightarrow$ Man braucht 2^i Iterationen,

um μf auf i Bits genau zu berechnen.

Lösung: Newton-Verfahren

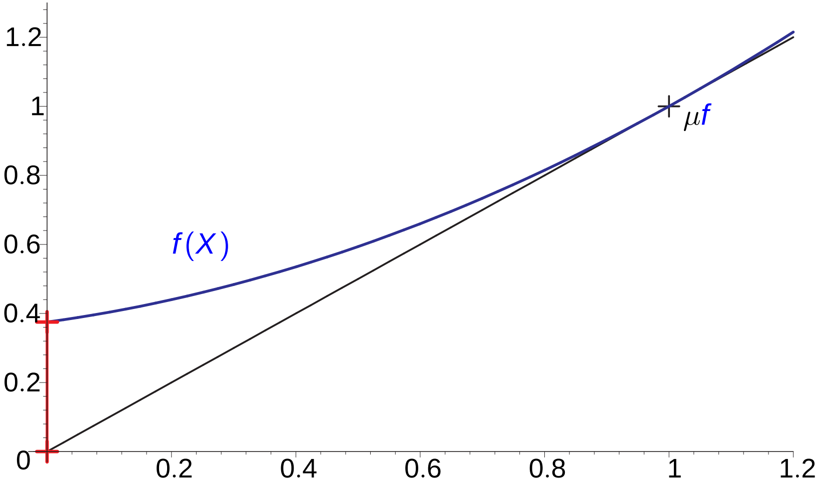
Kleene-Iteration (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



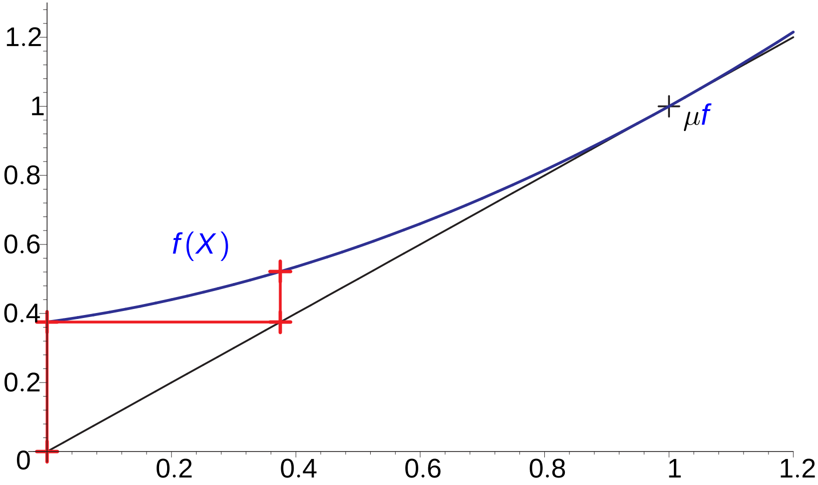
Kleene-Iteration (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



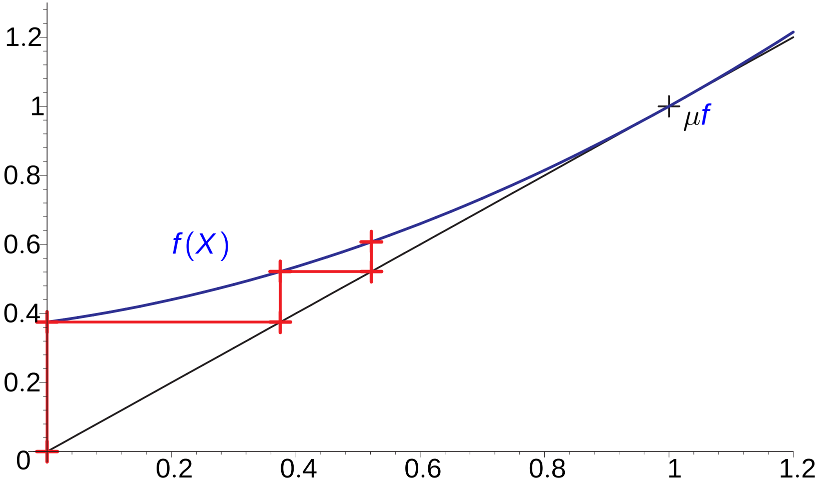
Kleene-Iteration (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



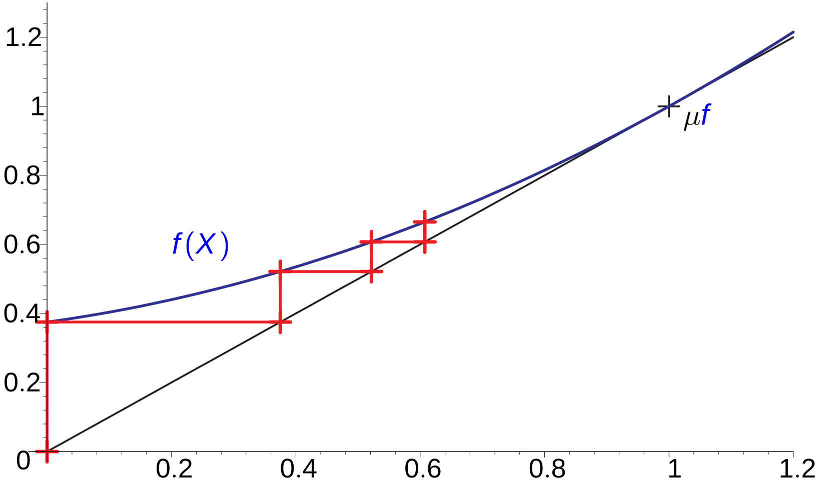
Kleene-Iteration (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



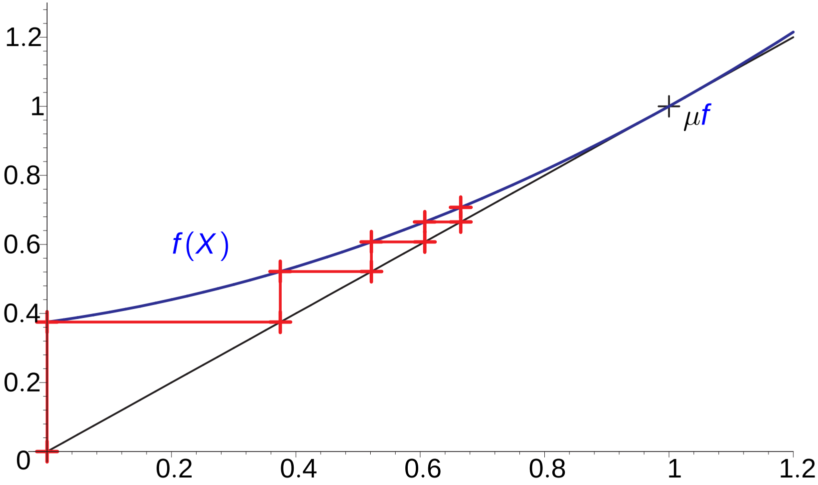
Kleene-Iteration (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



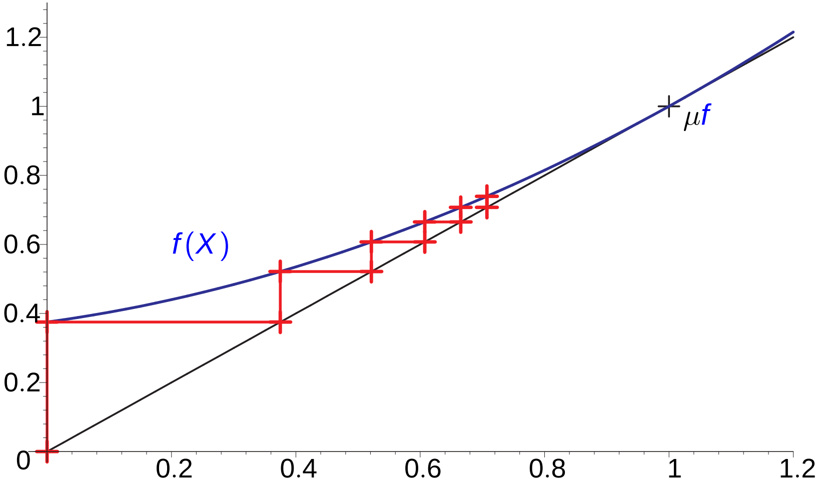
Kleene-Iteration (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



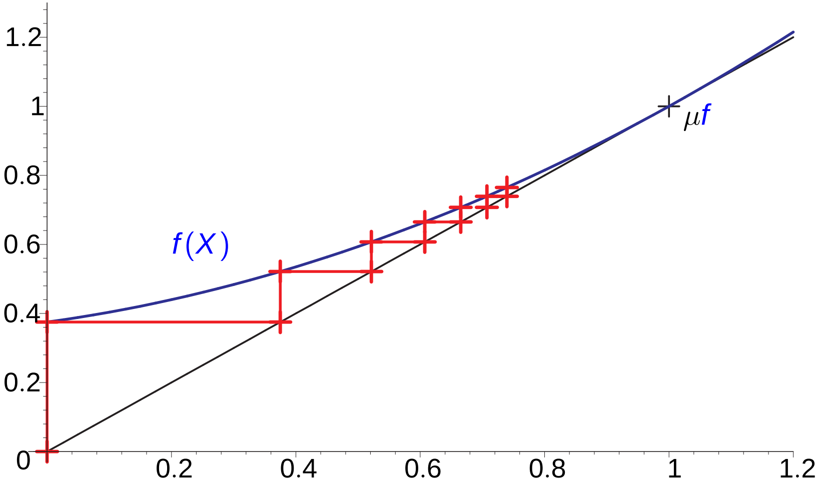
Kleene-Iteration (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



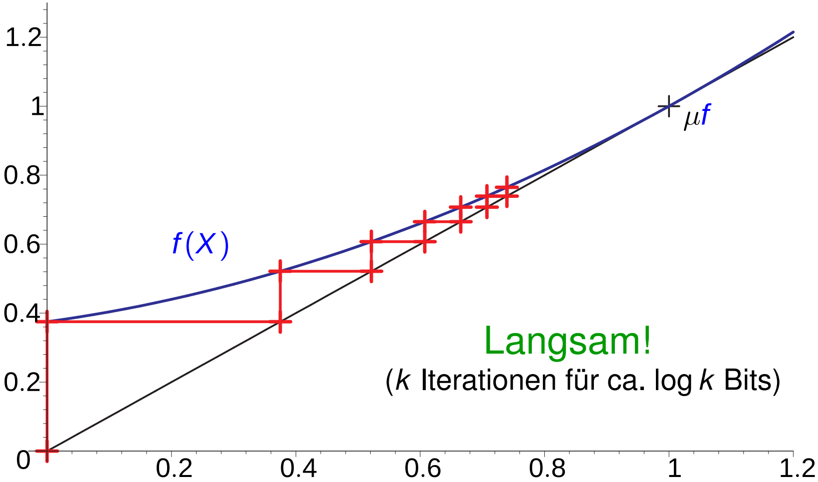
Kleene-Iteration (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



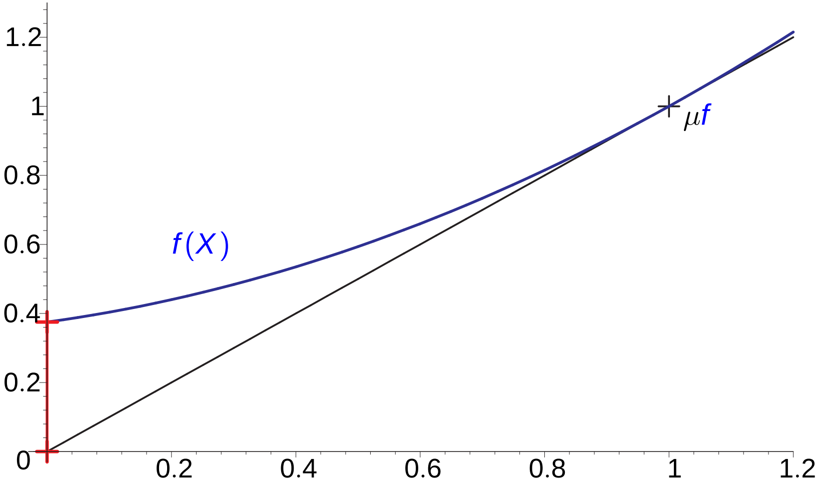
Kleene-Iteration (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



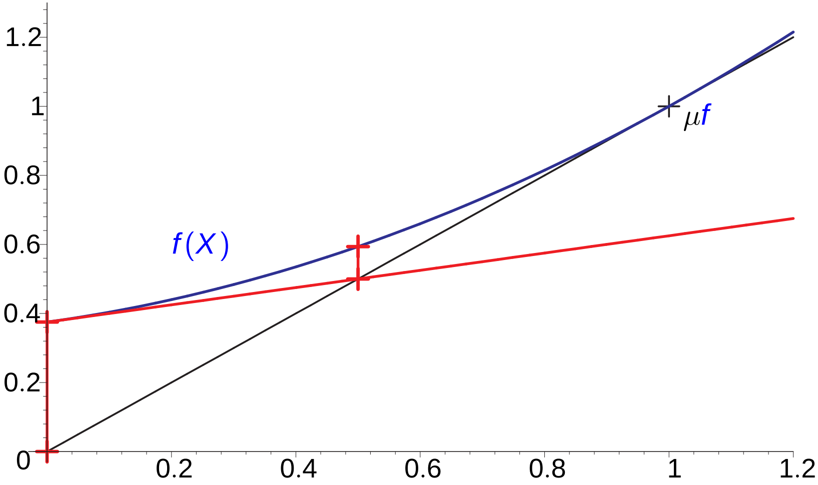
Newton-Verfahren (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



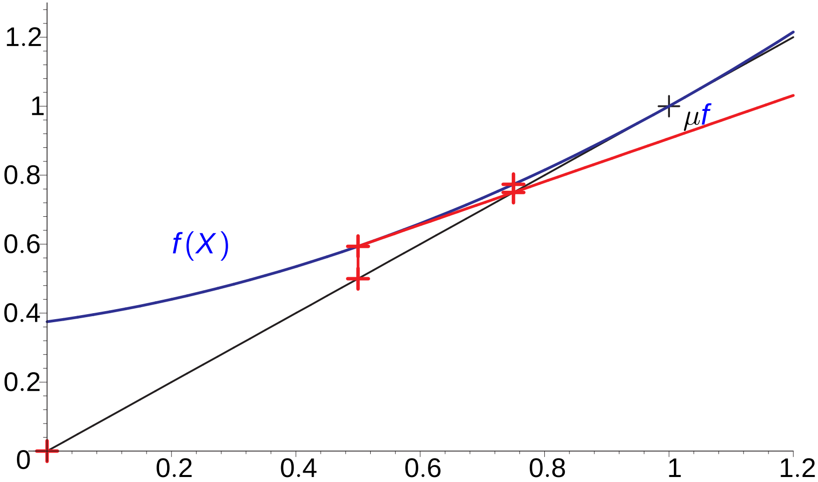
Newton-Verfahren (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



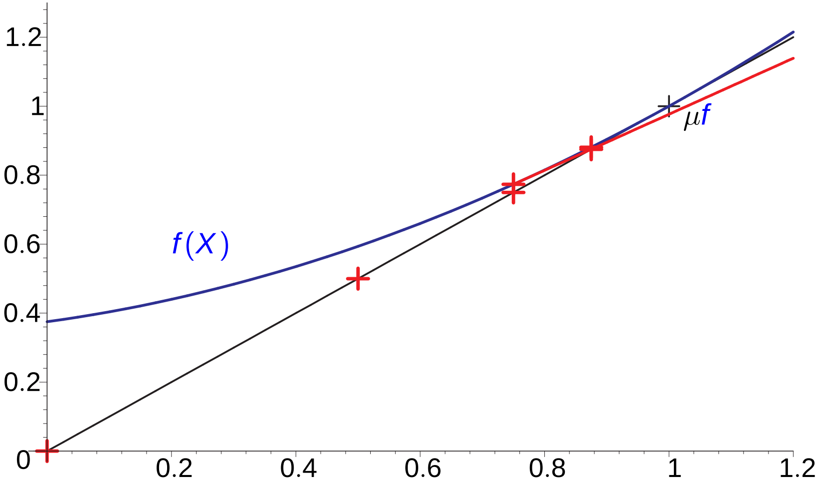
Newton-Verfahren (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



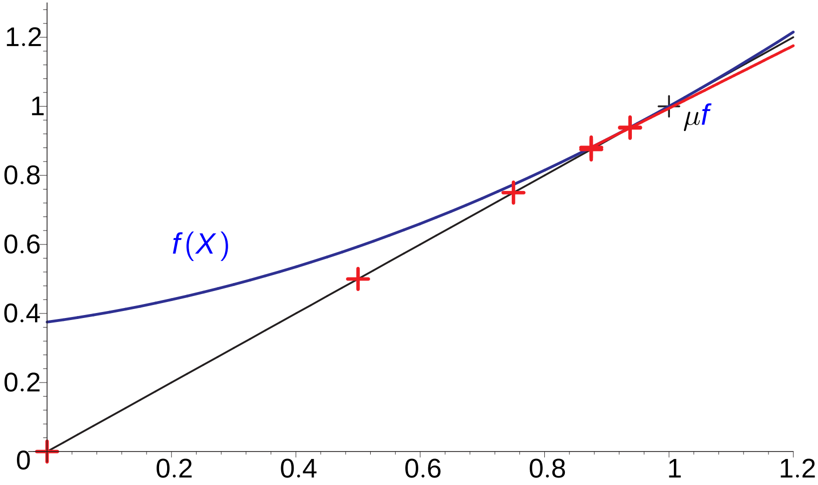
Newton-Verfahren (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



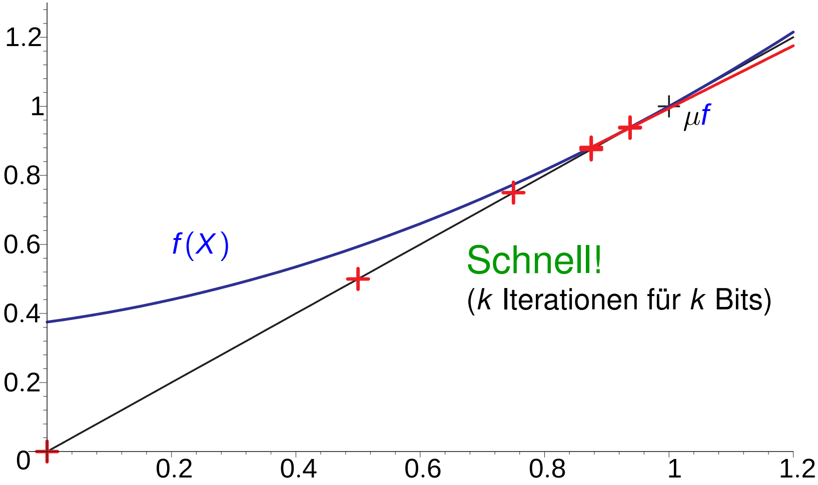
Newton-Verfahren (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



Newton-Verfahren (1-dimensional)

Betrachte $f(X) = 3/8 \cdot X \cdot X + 1/4 \cdot X + 3/8$



Schnell!

(k Iterationen für k Bits)

Das Newton-Verfahren konvergiert linear.

Satz

Sei $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$ stark zusammenhängend.
Dann gibt es eine "Schwelle" $t_{\boldsymbol{f}} \in \mathbb{N}$,
sodass der $(t_{\boldsymbol{f}} + k)$-te Newton-Iterant
mindestens k gültige Bits von $\boldsymbol{\mu f}$ aufweist.

Das Newton-Verfahren konvergiert linear.

Satz

Sei $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$ stark zusammenhängend.
Dann gibt es eine "Schwelle" $t_{\boldsymbol{f}} \in \mathbb{N}$,
sodass der $(t_{\boldsymbol{f}} + k)$-te Newton-Iterant
mindestens k gültige Bits von $\boldsymbol{\mu f}$ aufweist.

Insbesondere gilt das für
$t_{\boldsymbol{f}} = \lceil 4mn + 3n \max\{0, -\log \mu_{min}\} \rceil$ *und für*
$t_{\boldsymbol{f}} = 4mn\,2^n$, *wobei*

- *n die Zahl der Gleichungen (= Zahl der Variablen),*
- *m die Größe des Systems (Koeffizienten binär),*
- *μ_{min} die kleinste Komponente von $\boldsymbol{\mu f}$ ist.*

Wenn $\boldsymbol{f}(\mathbf{0}) > \mathbf{0}$ gilt, noch besser: $t_{\boldsymbol{f}} = 7mn$.
Nicht stark zusammenhängend: auch lineare Konvergenz.

Anwendungsbeispiel für den Satz

Betrachte das Beispiel vom Anfang:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4YX + 0.6 \\ 0.3XY + 0.4ZY + 0.3 \\ 0.3XZ + 0.7 \end{pmatrix}$$

14 Newton-Schritte (z.B. mit Maple) ergeben:

$$\nu^{(14)} = \begin{pmatrix} 0.98283\cdots \\ 0.97380\cdots \\ 0.99269\cdots \end{pmatrix}$$

Ist die Lösung $\boldsymbol{\mu f} = (1, 1, 1)$?

Anwendungsbeispiel für den Satz

Betrachte das Beispiel vom Anfang:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.4YX + 0.6 \\ 0.3XY + 0.4ZY + 0.3 \\ 0.3XZ + 0.7 \end{pmatrix}$$

14 Newton-Schritte (z.B. mit Maple) ergeben:

$$\nu^{(14)} = \begin{pmatrix} 0.98283\cdots \\ 0.97380\cdots \\ 0.99269\cdots \end{pmatrix}$$

Ist die Lösung $\boldsymbol{\mu f} = (1, 1, 1)$? Nein! Der Satz impliziert:
Der Fehler nach 14 Iterationen ist höchstens 0.004 (8 Bits).
Daher:

$$\boldsymbol{\mu f} \leq \nu^{(14)} + \begin{pmatrix} 0.004 \\ 0.004 \\ 0.004 \end{pmatrix} \leq \begin{pmatrix} 0.987 \\ 0.978 \\ 0.997 \end{pmatrix} < \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Minimum und Maximum zulassen

Wir lassen in positiven polynomiellen Gleichungssystemen

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$$

jetzt auch Minimum und Maximum zu.

Beispiel

$$\begin{aligned} X &= \max\{1.2X + 0.3Y^2, \quad 0.9Y\} \\ Y &= \min\{0.7X^2 + 0.2, \quad 0.4X\} \end{aligned}$$

Damit besteht jede Komponente von $\boldsymbol{f}(\boldsymbol{x})$ aus:

- Variablen
- positiven reellen Zahlen
- und den Operatoren $+, \cdot, \max, \min$

Wie zuvor: Wir möchten $\mu\boldsymbol{f}$ approximieren.

Beispiel: Grippe

Ein Patient hat Grippe. Die Ärztin hat zwei Möglichkeiten:

- Verschreibe ihm nichts.
- Behandle ihn mit Muniflu.

N = W'keit, den (ursprünglich *N*icht behandelten) Patienten zu kurieren, sowie alle, die er evtl. ansteckt

B = W'keit, einen *B*ehandelten Patienten zu kurieren, sowie alle, die er evtl. ansteckt

$$
\begin{aligned}
N &= \max\{\overbrace{0.3 + 0.7NN}^{\text{nicht behandeln}},\ \overbrace{0.9B + 0.1N}^{\text{behandeln}}\} \\
B &= \underbrace{0.35 + 0.65BN}_{}
\end{aligned}
$$

Beispiel: Grippe

Ein Patient hat Grippe. Die Ärztin hat zwei Möglichkeiten:

- Verschreibe ihm nichts.
- Behandle ihn mit Muniflu.

$N =$ W'keit, den (ursprünglich *N*icht behandelten) Patienten
zu kurieren, sowie alle, die er evtl. ansteckt

$B =$ W'keit, einen *B*ehandelten Patienten
zu kurieren, sowie alle, die er evtl. ansteckt

$$
\begin{aligned}
N &= \max\{\overbrace{0.3 + 0.7NN}^{\text{nicht behandeln}},\ \overbrace{0.9B + 0.1N}^{\text{behandeln}}\} \\
B &= \underbrace{0.35 + 0.65BN}
\end{aligned}
$$

Die kleinste Lösung ergibt die gesuchten W'keiten.
(Nicht unbedingt $N = B = 1$!)

Beispiel: Grippe

Ein Patient hat Grippe. Die Ärztin hat zwei Möglichkeiten:

- Verschreibe ihm nichts.
- Behandle ihn mit Muniflu.

$N =$ W'keit, den (ursprünglich *N*icht behandelten) Patienten zu kurieren, sowie alle, die er evtl. ansteckt

$B =$ W'keit, einen *B*ehandelten Patienten zu kurieren, sowie alle, die er evtl. ansteckt

$$\begin{aligned} N &= \max\{\overbrace{0.3 + 0.7NN}^{\text{nicht behandeln}},\ \overbrace{0.9B + 0.1N}^{\text{behandeln}}\} \\ B &= \qquad \underbrace{0.35 + 0.65BN}_{\text{normale Grippe}},\ \underbrace{0.5 + 0.2BN + 0.3BNN}_{\text{Schweinegrippe}} \end{aligned}$$

Die kleinste Lösung ergibt die gesuchten W'keiten.

(Nicht unbedingt $N = B = 1$!)

Beispiel: Grippe

Ein Patient hat Grippe. Die Ärztin hat zwei Möglichkeiten:

- Verschreibe ihm nichts.
- Behandle ihn mit Muniflu.

N = W'keit, den (ursprünglich *N*icht behandelten) Patienten zu kurieren, sowie alle, die er evtl. ansteckt

B = W'keit, einen *B*ehandelten Patienten zu kurieren, sowie alle, die er evtl. ansteckt

$$N = \max\{\overbrace{0.3 + 0.7NN}^{\text{nicht behandeln}},\ \overbrace{0.9B + 0.1N}^{\text{behandeln}}\}$$

$$B = \min\{\underbrace{0.35 + 0.65BN}_{\text{normale Grippe}},\ \underbrace{0.5 + 0.2BN + 0.3BNN}_{\text{Schweinegrippe}}\}$$

Die kleinste Lösung ergibt die gesuchten W'keiten.

(Nicht unbedingt $N = B = 1$!)

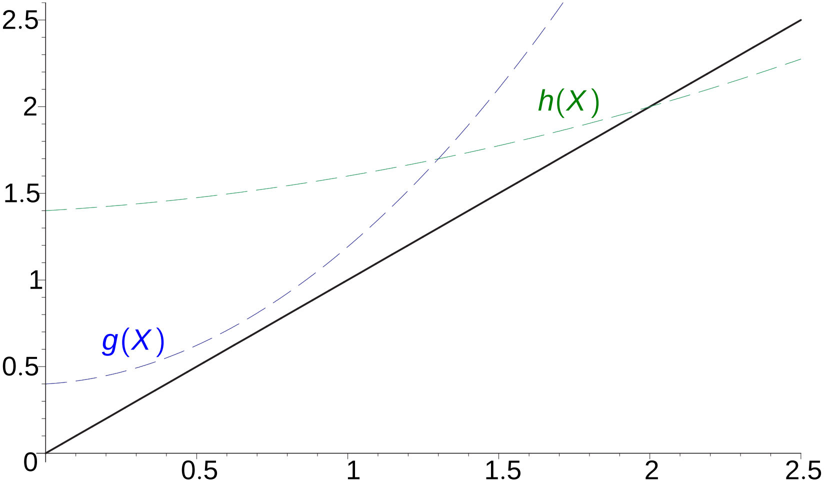
Kleene-Iteration $\mathbf{0}, \boldsymbol{f}(\mathbf{0}), \boldsymbol{f}(\boldsymbol{f}(\mathbf{0})), \ldots$

- funktioniert immer noch
- ist immer noch so langsam.

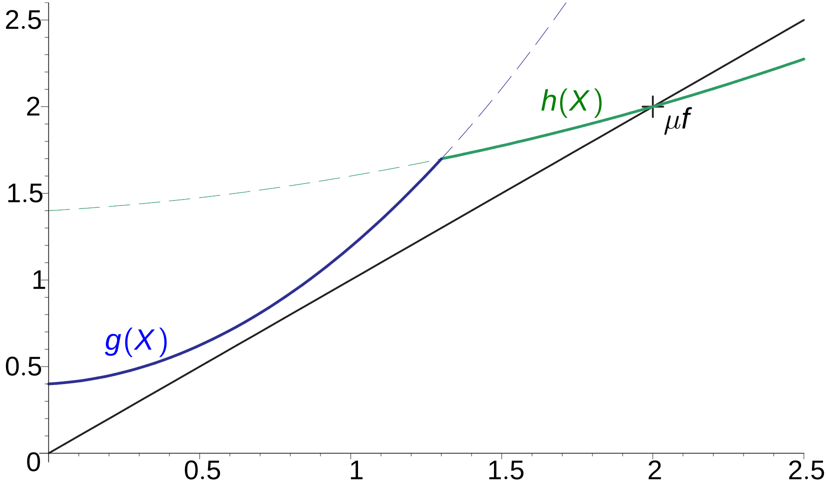
Was ist mit dem Newton-Verfahren?

Newton-Verfahren (1-dimensional)

Betrachte Newton-Verfahren mit $f(X) = \min\{g(X), h(X)\}$.

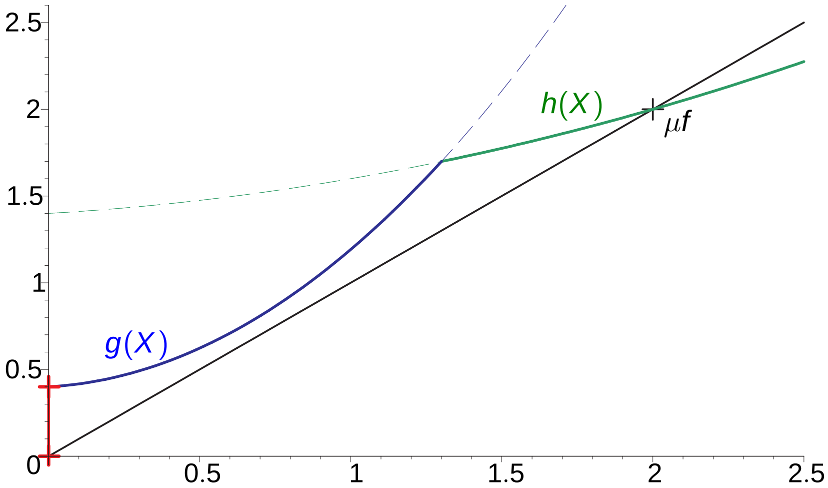


Betrachte Newton-Verfahren mit $f(X) = \min\{g(X), h(X)\}$.



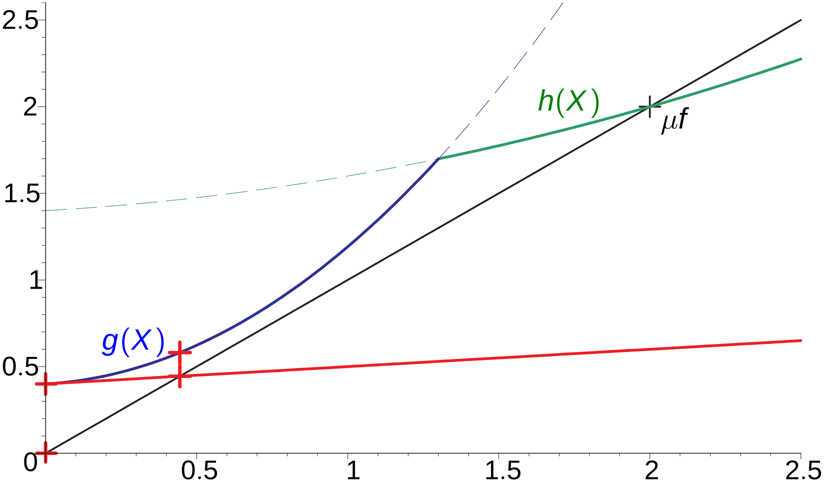
Newton-Verfahren (1-dimensional)

Betrachte Newton-Verfahren mit $f(X) = \min\{g(X), h(X)\}$.

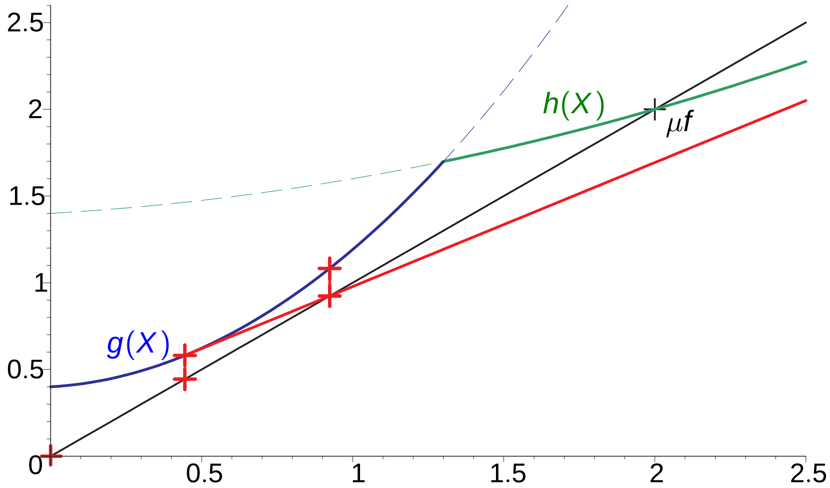


Newton-Verfahren (1-dimensional)

Betrachte Newton-Verfahren mit $f(X) = \min\{g(X), h(X)\}$.

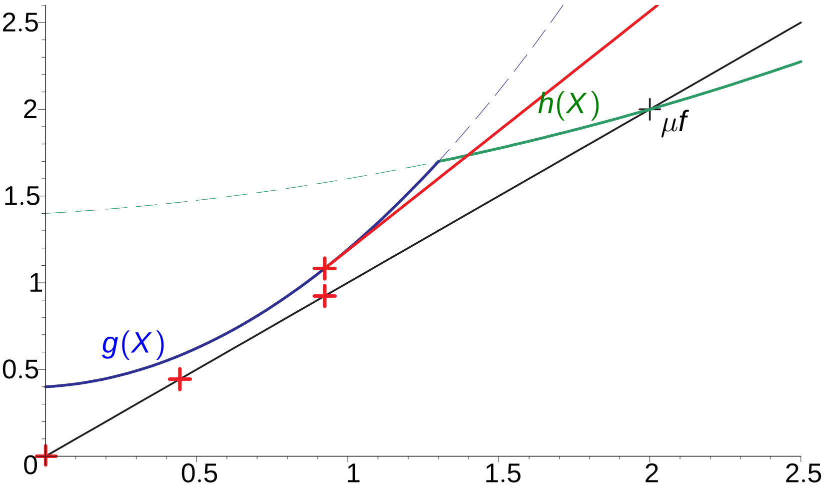


Betrachte Newton-Verfahren mit $f(X) = \min\{g(X), h(X)\}$.



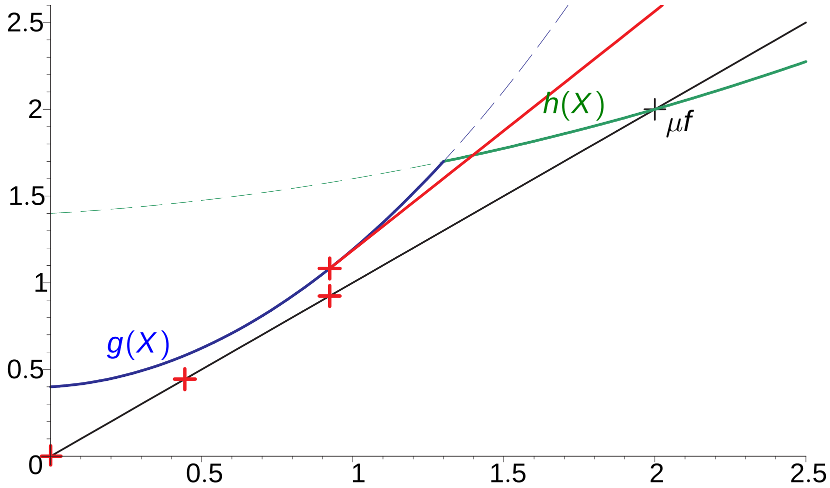
Newton-Verfahren (1-dimensional)

Betrachte Newton-Verfahren mit $f(X) = \min\{g(X), h(X)\}$.



Newton-Verfahren (1-dimensional)

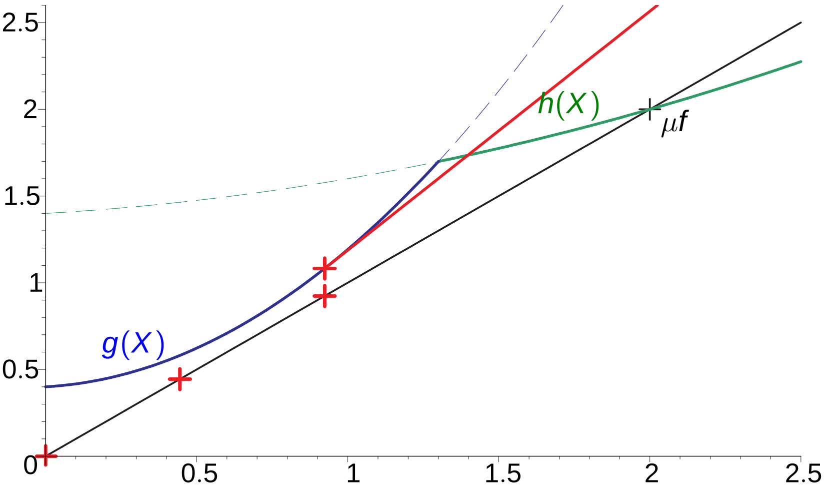
Betrachte Newton-Verfahren mit $f(X) = \min\{g(X), h(X)\}$.



Das Newton-Verfahren funktioniert nicht!

Newton-Verfahren (1-dimensional)

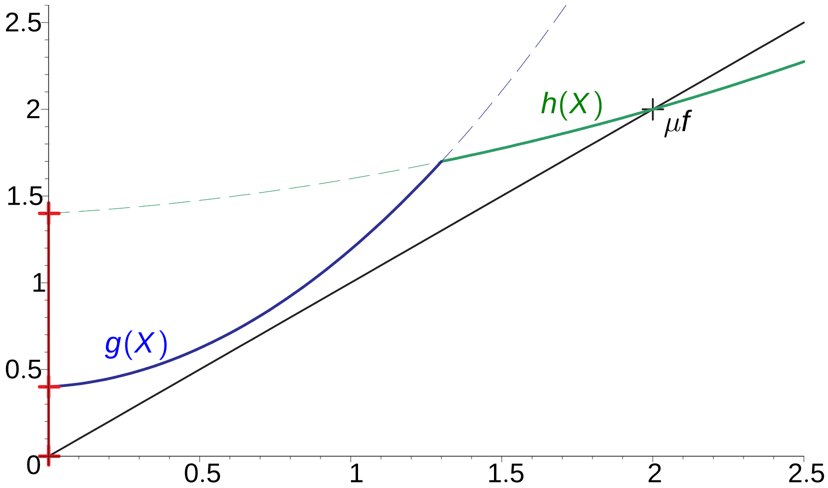
Betrachte Newton-Verfahren mit $f(X) = \min\{g(X), h(X)\}$.



Dann verändern wir es!

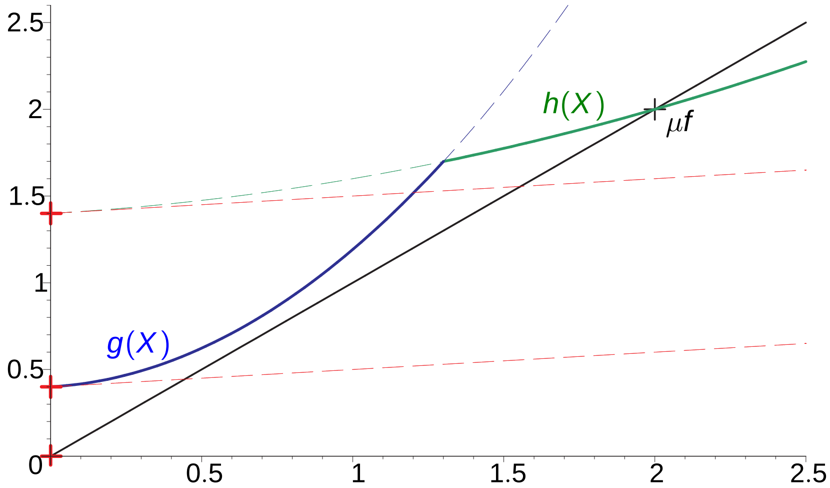
Verändertes Newton-Verfahren

Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



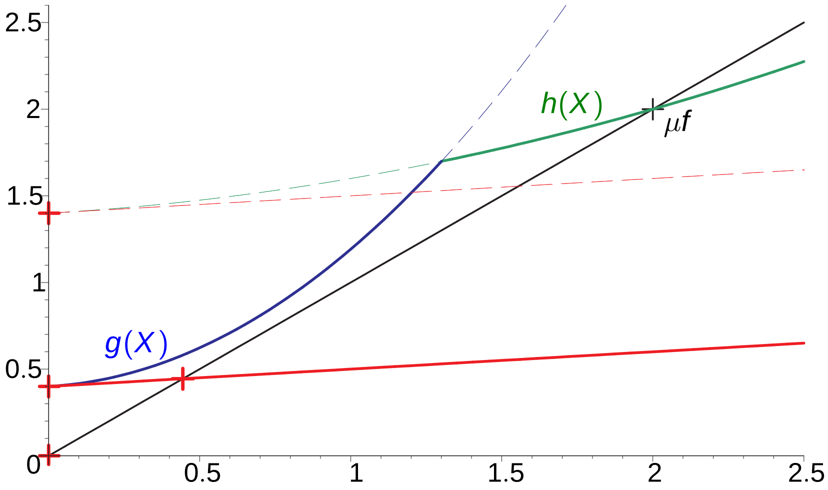
Verändertes Newton-Verfahren

Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



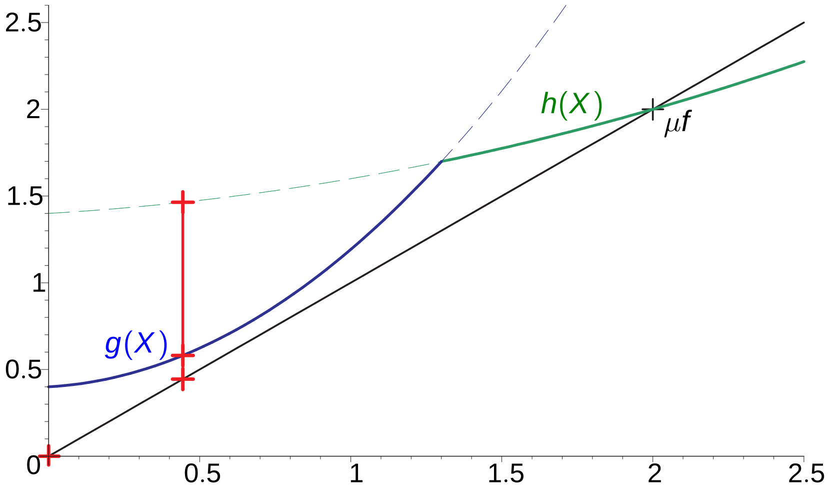
Verändertes Newton-Verfahren

Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



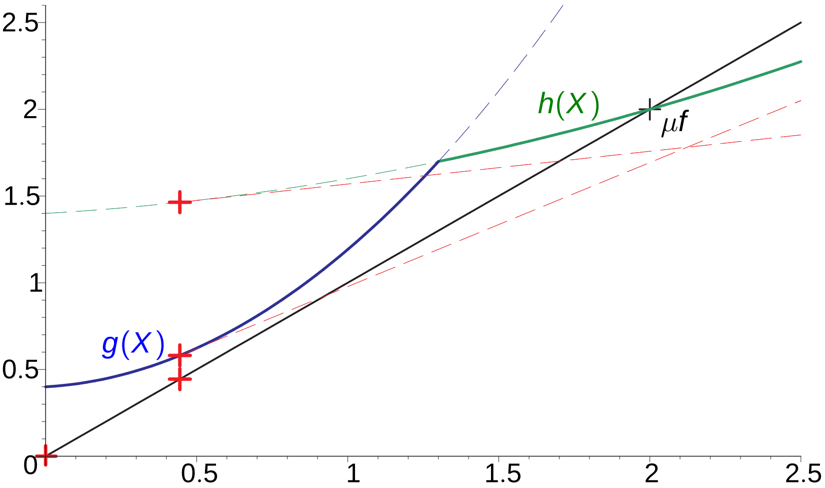
Verändertes Newton-Verfahren

Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



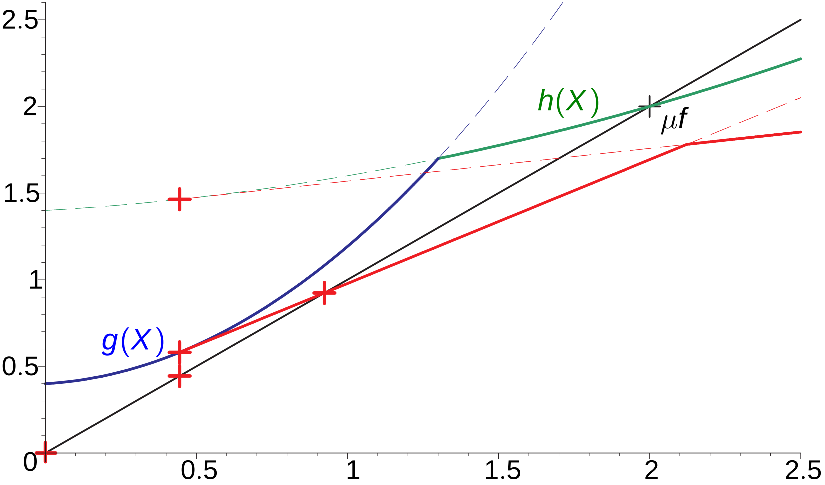
Verändertes Newton-Verfahren

Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



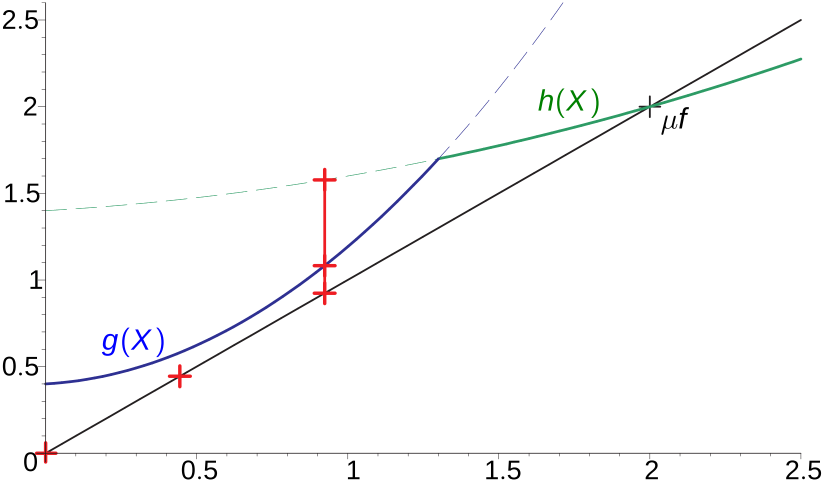
Verändertes Newton-Verfahren

Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



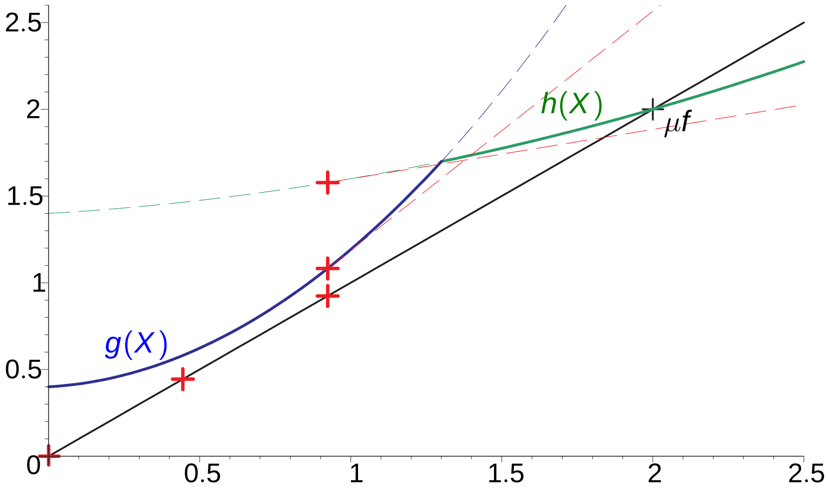
Verändertes Newton-Verfahren

Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



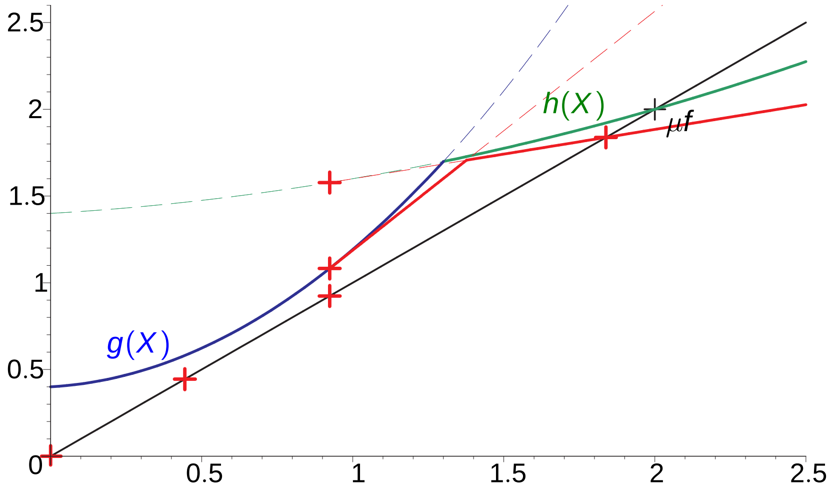
Verändertes Newton-Verfahren

Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



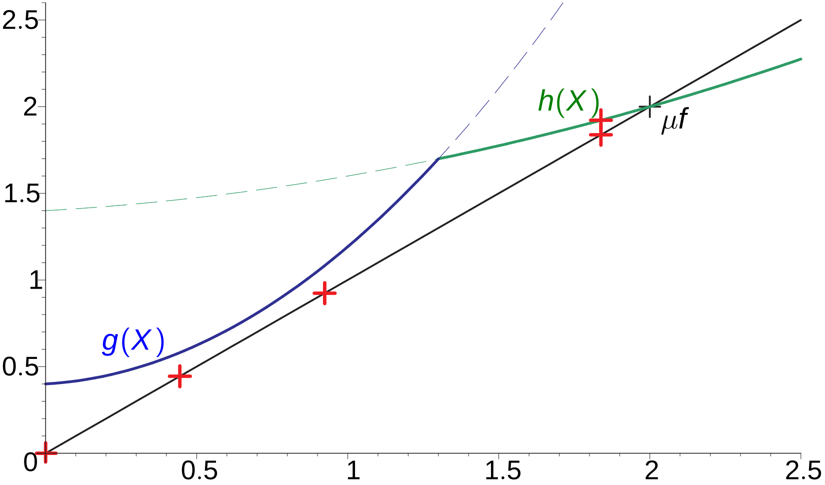
Verändertes Newton-Verfahren

Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



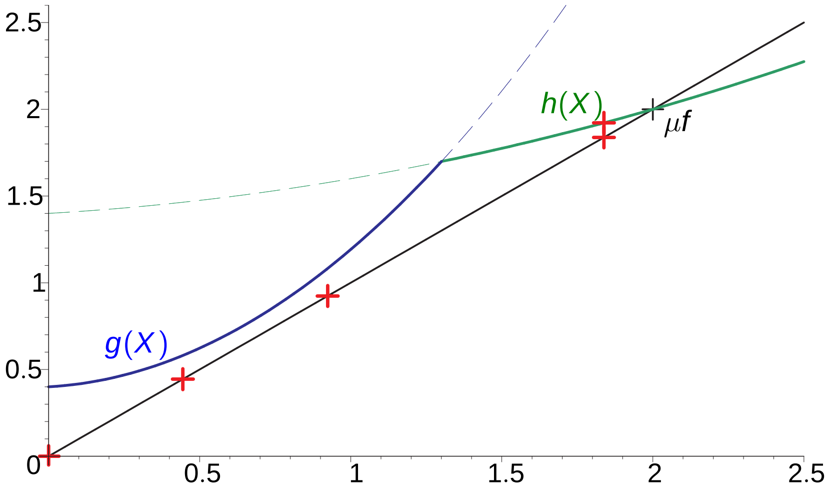
Verändertes Newton-Verfahren

Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



Verändertes Newton-Verfahren

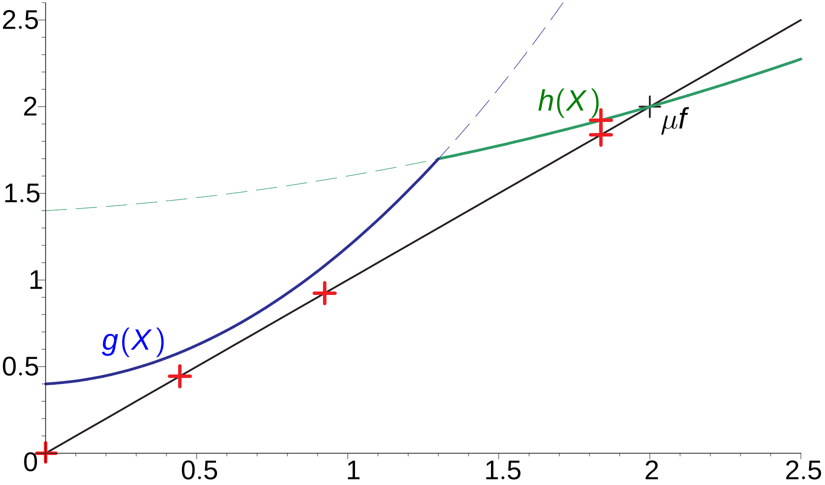
Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



Dieses Verfahren funktioniert!

Verändertes Newton-Verfahren

Wieder $f(X) = \min\{g(X), h(X)\}$. Aber beide Teile linearisieren!



Asymptotisch konvergiert es so schnell zu $\boldsymbol{\mu f}$
wie das Newton-Verfahren ohne min/max.

Verallgemeinerung auf Maximum

Das beschriebene Verfahren lässt sich auf Systeme mit Minimum und Maximum verallgemeinern.

Zwei Varianten:

- In jedem Schritt min-max-lineares System lösen
- In jedem Schritt min-lineares System lösen.

Verallgemeinerung auf Maximum

Das beschriebene Verfahren lässt sich auf Systeme mit Minimum und Maximum verallgemeinern.

Zwei Varianten:

- In jedem Schritt min-max-lineares System lösen
  $\Longrightarrow$ lineares Programmieren und Strategie-Iteration
  $\Longrightarrow$ große, aber teure Schritte
- In jedem Schritt min-lineares System lösen.
  $\Longrightarrow$ lineares Programmieren
  $\Longrightarrow$ kleine, aber billige Schritte

Verallgemeinerung auf Maximum

Das beschriebene Verfahren lässt sich auf Systeme mit Minimum und Maximum verallgemeinern.

Zwei Varianten:

- In jedem Schritt min-max-lineares System lösen
  $\Longrightarrow$ lineares Programmieren und Strategie-Iteration
  $\Longrightarrow$ große, aber teure Schritte
- In jedem Schritt min-lineares System lösen.
  $\Longrightarrow$ lineares Programmieren
  $\Longrightarrow$ kleine, aber billige Schritte

Beide Varianten konvergieren mindestens linear zu $\mu\boldsymbol{f}$.
Die erste berechnet ε-optimale Strategien.

Zurück zum Grippe-Beispiel

$N =$ W'keit, den (ursprünglich *N*icht behandelten) Patienten zu kurieren, sowie alle, die er evtl. ansteckt

$B =$ W'keit, einen *B*ehandelten Patienten zu kurieren, sowie alle, die er evtl. ansteckt

$$N = \max\{\overbrace{0.3 + 0.7NN}^{\text{nicht behandeln}},\ \overbrace{0.9B + 0.1N}^{\text{behandeln}}\}$$
$$B = \min\{\underbrace{0.35 + 0.65BN}_{\text{normale Grippe}},\ \underbrace{0.5 + 0.2BN + 0.3BNN}_{\text{Schweinegrippe}}\}$$

Mit der Variante mit kleinen, aber billigen Schritten:

$$\begin{pmatrix} N \\ B \end{pmatrix} \;\Big|\; \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0.300 \\ 0.350 \end{pmatrix} \begin{pmatrix} 0.409 \\ 0.465 \end{pmatrix} \begin{pmatrix} 0.524 \\ 0.524 \end{pmatrix} \begin{pmatrix} 0.538 \\ 0.538 \end{pmatrix} \begin{pmatrix} 0.538 \\ 0.538 \end{pmatrix} \dots$$

Zurück zum Grippe-Beispiel

N = W'keit, den (ursprünglich *N*icht behandelten) Patienten zu kurieren, sowie alle, die er evtl. ansteckt

B = W'keit, einen *B*ehandelten Patienten zu kurieren, sowie alle, die er evtl. ansteckt

$$N = \max\{\overbrace{0.3 + 0.7NN}^{\text{nicht behandeln}},\ \overbrace{0.9B + 0.1N}^{\text{behandeln}}\}$$
$$B = \min\{\underbrace{0.35 + 0.65BN}_{\text{normale Grippe}},\ \underbrace{0.5 + 0.2BN + 0.3BNN}_{\text{Schweinegrippe}}\}$$

Mit der Variante mit kleinen, aber billigen Schritten:

$$\begin{pmatrix} N \\ B \end{pmatrix} \ \Bigg|\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0.300 \\ 0.350 \end{pmatrix} \begin{pmatrix} 0.409 \\ 0.465 \end{pmatrix} \begin{pmatrix} 0.524 \\ 0.524 \end{pmatrix} \begin{pmatrix} 0.538 \\ 0.538 \end{pmatrix} \begin{pmatrix} 0.538 \\ 0.538 \end{pmatrix} \dots$$

Schnelle Konvergenz!

Zurück zum Grippe-Beispiel

$N =$ W'keit, den (ursprünglich *N*icht behandelten) Patienten
zu kurieren, sowie alle, die er evtl. ansteckt

$B =$ W'keit, einen *B*ehandelten Patienten
zu kurieren, sowie alle, die er evtl. ansteckt

$$N = \max\{\overbrace{0.3 + 0.7NN}^{\text{nicht behandeln}},\ \overbrace{0.9B + 0.1N}^{\text{behandeln}}\}$$
$$B = \min\{\underbrace{0.35 + 0.65BN}_{\text{normale Grippe}},\ \underbrace{0.5 + 0.2BN + 0.3BNN}_{\text{Schweinegrippe}}\}$$

Mit der Variante mit kleinen, aber billigen Schritten:

$$\begin{array}{c|cccccccc} \begin{pmatrix} N \\ B \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0.300 \\ 0.350 \end{pmatrix} & \begin{pmatrix} 0.409 \\ 0.465 \end{pmatrix} & \begin{pmatrix} 0.524 \\ 0.524 \end{pmatrix} & \begin{pmatrix} 0.538 \\ 0.538 \end{pmatrix} & \begin{pmatrix} 0.538 \\ 0.538 \end{pmatrix} & \cdots \\ \text{Aktion} & \text{nicht} & \text{nicht} & \text{beh.} & \text{beh.} & \text{beh.} & \text{beh.} & \cdots \\ \text{d. Ärztin} & \text{beh.} & \text{beh.} & & & & & \end{array}$$

Schnelle Konvergenz!

Zusammenfassung

- Die Lösung positiver polynomieller Gleichungssysteme
  - ist ein natürliches Problem
  - hat vielerlei Informatik-Anwendungen
- Die entwickelten Lösungsverfahren basieren auf
  - dem Newton-Verfahren
  - linearem Programmieren und/oder Strategie-Iteration
- Es wird lineare Konvergenz gezeigt
  - mit Schranken für die Konvergenzrate
  - und teilweise Schwellen

Ende des Vortrags

Danke!