Diplomarbeit

# Numerische Verfahren zur Berechnung von Lösungszweigen nichtlinearer Gleichungssysteme

Günther Eibl

eingereicht im August 1997
bei Univ.-Doz. Dr. Alexander Ostermann
Institut für Mathematik
Universität Innsbruck

# Inhaltsverzeichnis

# Einleitung

Das Ziel der vorliegenden Diplomarbeit sei an einem einfachen Beispiel erläutert. Man betrachte die Gleichung

$$F(x,\tau) \equiv x(x^2-\tau) = 0, \quad (x,\tau) \in \mathbb{R}^2$$

Gesucht ist eine Funktion $x(\tau)$ mit der Eigenschaft

$$F(x(\tau),\tau) = 0, \quad \forall \tau \in [-1,1] \tag{0.1}$$

$\tau$ wird Parameter genannt.
Wie man leicht sieht, besteht die Lösung aus der $x$-Achse und der Parabel $x = \pm\sqrt{\tau}$.

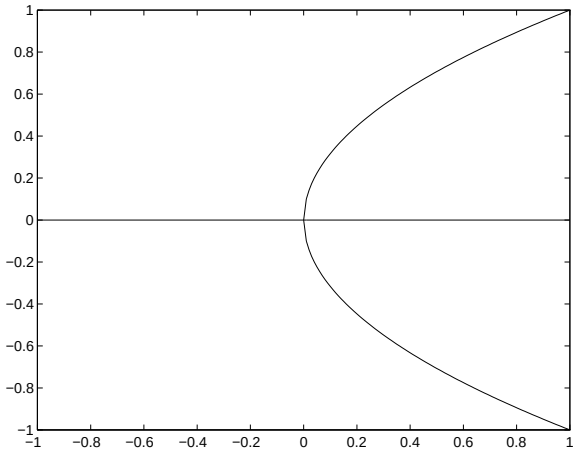


Abb.1 Lösungskurven von $x(x^2-\tau) = 0$ für $\tau \in [-1,1]$

Bei $(x,\tau) = (0,0)$ schneiden sich die Parabel und die $x$-Achse, solche Punkte werden Bifurkationspunkte genannt. Die Aufgabe besteht darin, alle Kurven $x(\tau)$ zu finden, die (0.1) erfüllen und für die $\tau$ in $[-1,1]$ liegt. Man muß daher allen Ästen, die bei $(0,0)$ abzweigen, entlanggehen.
In Kapitel 1 wird ein rein numerisches Verfahren vorgestellt, mit dessen Hilfe man obiges Problem lösen kann.
Kapitel 2 beschränkt sich auf polynomiale Gleichungssysteme und enthält ein überwiegend algebraisches Verfahren, das das Problem unter Verwendung von Gröbnerbasen

auf die Berechnung von Nullstellen von Polynomen zurückführt. Diese Nullstellen werden dann numerisch berechnet.
In Kapitel 3 werden die Programme vorgestellt und anhand von Testbeispielen verglichen.

# 1. Numerische Berechnung von Bifurkationsdiagrammen

Der Gegenstand dieser Diplomarbeit sind nichtlineare Gleichungssysteme der Form

$$F(x,\tau) = 0, \quad F : \mathbb{R}^n \times \mathbb{R} \longrightarrow \mathbb{R}^n.$$

Unter gewissen Voraussetzungen an die Jacobimatrix von $F$ besteht die Lösungsmenge dieses Gleichungssystem aus Kurven $x(\tau)$ mit $F(x(\tau),\tau) = 0$. Diese Kurven können sich in einzelnen Punkten schneiden, die Schnittpunkte heißen Bifurkationspunkte. Das Ziel ist es, die Lösungskurven und insbesondere die Bifurkationspunkte zu berechnen. In diesem Kapitel wird zuerst ein Verfahren vorgestellt, mit dem man Lösungskurven, die keine Bifurkationspunkte besitzen, berechnen kann. Dafür benötigt man die Tangenten in den berechneten Punkten. In Bifurkationspunkten schneiden sich zwei verschiedene Lösungskurven, und man kann daher im Bifurkationspunkt zwei verschiedene Tangenten, für jede Lösungskurve eine, finden. Um einer speziellen Lösungskurve zu folgen, berechnet man ihre Tangente im Bifurkationspunkt und wendet dann das erste Verfahren zu ihrer Berechnung an.

## 1.1. Fortsetzungsverfahren

**Definition**: Sei $D \subseteq \mathbb{R}^n$ offen und $F : D \times \mathbb{R} \longrightarrow \mathbb{R}^n$ gegeben. Ein Punkt $(x,\tau)$ heißt
*regulärer Punkt* $:\Longleftrightarrow \operatorname{Rg}(F'(x,\tau)) = n$
*Umkehrpunkt* $:\Longleftrightarrow$ $(x,\tau)$ ist regulär und $\operatorname{Rg}(F_x(x,\tau)) = n-1$
*Bifurkationspunkt* $:\Longleftrightarrow \operatorname{Rg}(F'(x,\tau)) = n-1$
Ein Weg $x : [a,b] \longrightarrow D : \tau \mapsto x(\tau)$ heißt *Homotopieweg von a nach b*
$:\Longleftrightarrow \quad F(x(\tau),\tau) = 0, \quad \forall \tau \in [a,b]$

**Bemerkung**: Unter einem Bifurkationspunkt stellt man sich einen Punkt vor, bei dem sich ein Homotopieweg in mehrere Äste aufspaltet. Nach dem Satz über implizite Funktionen kann das nicht bei $(x,\tau)$ mit $\operatorname{Rg}(F'(x,\tau)) = n$ der Fall sein. Außerdem soll $\operatorname{Rg}(F'(x,\tau))$ nicht kleiner als $n-1$ sein, was sogenannte „mehrfache Bifurkationspunkte“ ausschließen soll.

In diesem Kapitel wird ein Verfahren vorgestellt, das $x(\tau)$ berechnen kann, falls $x(\tau)$ nur aus regulären Punkten besteht.

### 1.1.1. Prädiktor-Korrektor-Verfahren

**Bemerkung**: Man kann sich darauf beschränken, $x(\tau)$ nur für $\tau \in [0,1]$ zu bestimmen. Mit Hilfe der linear-affinen Transformation

$$\varphi : \mathbb{R}^n \times [0,1] \longrightarrow \mathbb{R}^n \times [a,b] : (x,\tau) \mapsto (x, (1-\tau)a + \tau b)$$

erhält man aus $F : \mathbb{R}^n \times [a,b] \longrightarrow \mathbb{R}^n$ ein Gleichungssystem

$$\tilde{F} : \mathbb{R}^n \times [0,1] \longrightarrow \mathbb{R}^n : \tilde{F} := F \circ \varphi$$

Aus einem Homotopieweg $\tilde{x}(\tau)$ von $\tilde{F}$ erhält man $x(\tau)$ durch $(x(\tau),\tau) := \varphi(\tilde{x}(\tilde{\tau}),\tilde{\tau})$, denn es gilt:

$$0 = \tilde{F}(\tilde{x}(\tilde{\tau}),\tilde{\tau}) = (F \circ \varphi \circ \varphi^{-1})(x(\tau),\tau) = F(x(\tau),\tau).$$

Wenn man den Homotopieweg $x(\tau)$ berechnen will, ist es naheliegend, zuerst das Intervall $[0,1]$ zu diskretisieren. Man wählt also eine Unterteilung $0 = \tau_0 < \tau_1 < \ldots < \tau_N = 1$ und versucht dann, die Gleichungssysteme

$$F(x,\tau_i) = 0, \qquad i = 0, \ldots, N$$

mit dem Newtonverfahren zu lösen.

**Wahl des Startwertes:**

Dazu muß man dem Newtonverfahren einen Startwert für das erste Gleichungssystem $F(x,\tau_0) = 0$ übergeben. Startwerte für die restlichen Gleichungssysteme besorgt man sich dann von den zuvor ausgerechneten Lösungen. Dazu gibt es mehrere Möglichkeiten. Es seien $F(x,\tau_i) = 0,\ i = 0, \ldots, k$ schon gelöst d.h. $x_i$ mit $F(x_i,\tau_i) = 0$ berechnet worden. Bei der sogenannten klassischen Methode wählt man für den Startwert $\hat{x}_{k+1}$ von $F(x,\tau_{k+1}) = 0$ einfach $x_k$.

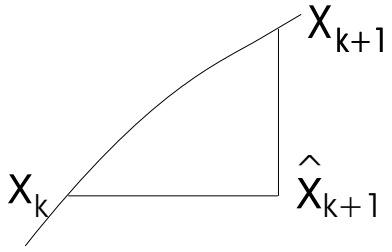


Abb.2: Klassische Methode

Falls die Lösungskurve $x(\tau)$ steil ist, kann $\hat{x}_{k+1}$ von $x_{k+1}$ weit weg sein, es wird also besser sein, $\hat{x}_{k+1}$ mit Hilfe der Tangente an $x_k$ zu bestimmen, weshalb man dieses Vorgehen auch als tangentiale Fortsetzungsmethode bezeichnet.

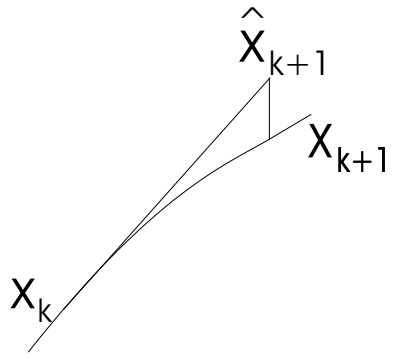


Abb.3: Tangentiale Fortsetzungsmethode

$$\hat{x}_{k+1} := x(\tau_k) + (\tau_{k+1} - \tau_k)x'(\tau_k).$$

$x'(\tau_k)$ wird wie folgt berechnet: Wenn man $F(x(\tau), \tau) = 0$ differenziert, ergibt sich

$$F_x(x(\tau), \tau)x'(\tau) + F_\tau(x(\tau), \tau) = 0,$$

also

$$x'(\tau_k) \approx -\left(F_x(x_k, \tau_k)\right)^{-1} F_\tau(x_k, \tau_k).$$

Die Matrix $F_x(x(\tau_k), \tau_k)$ wurde schon beim Newtonverfahren zur Berechnung von $x_k$ zerlegt, sodaß die Berechnung der Tangente nicht mehr sehr aufwendig ist.

Das Verfahren funktioniert nicht, wenn die Inverse von $F_x$ nicht existiert, da sie beim Newtonverfahren benötigt wird. Der Satz über implizite Funktionen besagt, daß man die Lösungskurve dort nicht mehr nach $\tau$ parametrisieren kann. Bei diesem Verfahren muß man also nicht nur Bifurkations-, sondern auch Umkehrpunkte ausschließen.

### 1.1.2. Pseudoarclength Continuation

Die Prädiktor-Korrektor-Verfahren scheitern bei Umkehrpunkten, obwohl dort der Rang der Jacobimatrix $F'(x, \tau)$ gleich $n$ ist. Der Satz über implizite Funktionen sagt aus, daß es dort eine lokal eindeutig bestimmte Lösungskurve $(x, \tau)(s)$ gibt. Man versucht daher, eine Kurve zu finden, die nicht nach $\tau$ parametrisiert ist, man gibt also die Bedeutung von $\tau$ als Parameter auf.

#### 1.1.2.1. Vorstellung des Verfahrens

**Bezeichnung**: Sei $D \subseteq \mathbb{R}^{n+1}$ offen, $F : D \longrightarrow \mathbb{R}^n : \ (x, \tau) \mapsto F(x, \tau)$ gegeben und $F(x, \tau) = 0$ das zu untersuchende Gleichungssystem.
Dann werden $u$ und $H$ wie folgt definiert:

$$u := (x, \tau)$$
$$H : D \longrightarrow \mathbb{R}^n : \quad u \mapsto F(x, \tau)$$

Ein Weg $u : [a, b] \longrightarrow D : s \mapsto u(s) = (x(s), \tau(s))$ heißt **Homotopieweg**
$:\Longleftrightarrow \quad H(u(s)) = 0, \quad \forall s \in [a, b]$

Das Gleichungssystem $H(u) = 0$ besteht aus $n+1$ Gleichungen in $n$ Unbekannten. Wenn man das Newtonverfahren anwenden will, muß man eine Gleichung dazunehmen, die einen neuen Punkt $u \in \{u(s);\ a \leq s \leq b\}$ auswählt. Diese Gleichung wird am besten durch eine Zeichnung veranschaulicht:

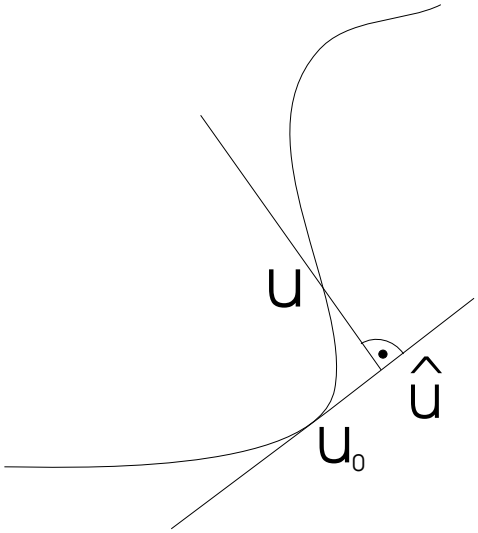


Abb.4: Hinzunahme einer Gleichung zur Berechnung von $u$

Es sei also ein Punkt $u_0$ auf $u(s)$ vorgegeben und $\Delta u_0$ die Tangente von $u(s)$ in $u_0$. Im Fall $n = 1$ soll der neue Punkt $u$ auf einer Normalen zur Tangente durch $u_0$ liegen. Als Startwert $\hat{u}$ für das Newtonverfahren wählt man einen Punkt auf der Tangente $\Delta u_0$, d.h.

$$\hat{u} := u_0 + \lambda \Delta u_0 \qquad \lambda \in \mathbb{R}$$

Die neue Gleichung lautet dann

$$\Delta u_0^T (u - \hat{u}) = 0.$$

Wie man aus der Zeichnung sehen kann, ist $u$ durch diese zusätzliche Bedingung im Fall $n = 1$ für festes $\lambda$ auch bei Umkehrpunkten eindeutig bestimmt. Daß dies auch für beliebiges $n$ gilt, zeigt folgender Satz:

**Satz 1.1:** Sei $u_0$ ein Punkt mit $H(u_0) = 0$ und $\mathrm{Rg}(H'(u_0)) = n$. Dann ist

$$\begin{pmatrix} H'(u_0) \\ \Delta u_0 \end{pmatrix}$$

invertierbar, d.h. das Newtonverfahren kann verwendet werden, um $u$ zu finden.

**Beweis**: Die Matrix $A := \begin{pmatrix} H'(u_0) \\ \Delta u_0 \end{pmatrix}$ ist eine $(n+1)\times(n+1)$-Matrix, also quadratisch.
Somit genügt es zu zeigen, daß $\mathrm{Ker}(A)$ nur aus dem Nullvektor besteht. Sei $w \in \mathrm{Ker}(A)$.
$w$ habe die Eigenschaft $Aw = 0$. Zeige $w = 0$.
Die ersten $n$ Zeilen von $Aw = 0$ lauten $H'(u_0)w = 0$, d.h. $w$ liegt in $\mathrm{Ker}(H'(u_0))$.
Man betrachtet also den Kern von $H'(u_0)$ genauer:
Sei $u(s)$ eine Kurve mit $H(u(s)) = 0$ und $u(s_0) = u_0$.
Durch Differenzieren von $H(u(s)) = 0$ im Punkt $s_0$ erhält man
$H'(u_0)\Delta u_0 = 0$ , d.h. $\Delta u_0$ liegt in $\mathrm{Ker}(H'(u_0))$.
Andererseits hat $\mathrm{Ker}(H'(u_0))$ die Dimension 1, denn
$\dim(\mathrm{Ker}(H'(u_0))) = n + 1 - \mathrm{Rg}(H'(u_0)) = 1$,
und hat daher folgende Form:

$$\mathrm{Ker}(H'(u_0)) = \mathbb{R}\Delta u_0$$

Insbesondere hat man eine Darstellung von $w$ gewonnen:

$$w = \alpha\Delta u_0, \quad \text{für } \alpha \in \mathbb{R}$$

Man setzt diesen Ausdruck in die letzte Gleichung von $Aw = 0$ ein und erhält
$0 = \Delta u_0 w = \alpha\|\Delta u_0\|^2$.
Wegen $\|u_0\| > 0$ gilt daher $\alpha = 0$
und somit $w = \alpha\Delta u_0 = 0$.

□

**Bemerkung**: Das Gleichungssystem $\begin{pmatrix} H(u) \\ \Delta u_0^T(u - u_0) \end{pmatrix} = 0$ hängt noch von $\lambda$, das in $\hat{u}$ enthalten ist, ab. Dieses $\lambda$ wird im Algorithmus vor jedem Schritt gewählt. Wenn $\lambda$ klein ist, ist der Startwert für das Newtonverfahren sehr gut, es wird also schnell konvergieren, dafür kommt man mit einem Schritt nicht sehr weit (siehe Abb.4). Man wird also versuchen, $\lambda$ möglichst groß zu wählen, unter der Bedingung, daß das Newtonverfahren noch konvergiert.

#### 1.1.2.2. Berechnung der Tangente $\Delta u_0$

Wie im Beweis gezeigt wurde, liegt $\Delta u_0$ im eindimensionalen Kern von $H'(u_0)$, $\Delta u_0$ ist daher nach einer Normierung eindeutig bestimmt.

$$\begin{aligned} H'(u_0)\Delta u_0 &= 0 \\ \Delta u_0^T \Delta u_0 &= 1 \end{aligned}$$

Die praktische Berechnung von $\Delta u_0$ wird mit der QR-Zerlegung von $H'(u_0)$ durchgeführt.

$$H'(u_0) = QRP$$

$Q$ ist eine orthogonale Matrix, $P$ eine Permutationsmatrix und

$$R = \begin{pmatrix} \alpha_1 & * & & & * \\ 0 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ 0 & & 0 & \alpha_n & * \end{pmatrix}$$

mit $|\alpha_1| \geq |\alpha_2| \geq \ldots \geq |\alpha_n|$. $|\alpha_n| > 0$, sonst wäre $\mathrm{Rg}(H'(u_0)) < n$.

$H'(u_0)\Delta u_0 = QRP\Delta u_0 = 0$

$\iff R(P\Delta u_0) = 0$

$$\iff \begin{pmatrix} \alpha_1 & * & & & * \\ 0 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ 0 & & 0 & \alpha_n & * \end{pmatrix} (P\Delta u_0) = 0$$

$$\iff \begin{pmatrix} \alpha_1 & * & & * \\ 0 & \alpha_2 & \ddots & \\ & \ddots & \ddots & * \\ 0 & & 0 & \alpha_n \end{pmatrix} \begin{pmatrix} (P\Delta u_0)_1 \\ \vdots \\ (P\Delta u_0)_n \end{pmatrix} = -\left(R_{-,n+1}\right)(P\Delta u_0)_{n+1}.$$

Man kann also $(P\Delta u_0)_{n+1}$ frei wählen und die restlichen Komponenten von $P\Delta u_0$ aus obiger Gleichung berechnen. $\Delta u_0$ ist aus $P\Delta u_0$ leicht zu bestimmen durch:

$$\Delta u_0 = P^T (P\Delta u_0)$$

Schließlich muß man $\Delta u_0$ noch normieren und dabei berücksichtigen , daß $\Delta u_0$ mit der im vorhergehenden Schritt berechneten Tangente $\Delta u_{-1}$ einen spitzen Winkel einschließt, sonst könnte man wieder zurückgehen. Ein Winkel ist spitz, wenn sein Cosinus positiv ist, in diesem Fall also genau dann wenn $\Delta u_{-1}^T \Delta u_0 > 0$. Das ergibt:

$$\Delta u_0 = \begin{cases} +\frac{\Delta u_0}{\|\Delta u_0\|} & \Delta u_{-1}^T \Delta u_0 > 0 \\ -\frac{\Delta u_0}{\|\Delta u_0\|} & \Delta u_{-1}^T \Delta u_0 < 0\,. \end{cases}$$

#### 1.1.2.3. Algorithmus 1 (pseudoarclength continuation)

Es sei $u_0$ mit $H(u_0) = 0$ und $\lambda$ gegeben.

**Schritt 1:**
Berechne $\Delta u_0$ wie oben.

**Schritt 2:**
Setze $\hat{u} := u_0 + \lambda \Delta u_0$.
Wende das Newtonverfahren auf

$$\begin{pmatrix} H(u) \\ \Delta u_0^T (u - u_0) \end{pmatrix} = 0$$

solange an, bis entweder $\|H(u)\| <$TOL
oder $anz$ :=Anzahl der Iterationsschritte=6 ist.

**Schritt 3:**

Falls $anz \leq 2$: $\lambda := 2\lambda$
$u_0 := u$
Gehe zu Schritt 1.
Falls $anz$=6: $\lambda := \frac{\lambda}{2}$
Gehe zu Schritt 2.

TOL ist vom Benutzer einzugeben.
Falls nach 6 Iterationen abgebrochen wurde, war die Konvergenz zu schlecht. Man muß also die Konvergenz verbessern, indem man einen besseren Startwert wählt, d.h. man verkleinert $\lambda$ (siehe Abb.4).
Falls das Newtonverfahren schon nach 2 Iterationen konvergiert hat, kann man für den nächsten Schritt $\lambda$ vergrößern. Man will ja in möglichst großen Schritten der Kurve entlanggehen.

Das Verfahren läuft so lange, bis man nahe genug an $\tau = 1$ kommt. Um genau zu $\tau = 1$ zu kommen, kann man beim letzten Schritt die Bedingung $\Delta u_0^T (u - u_0) = 0$ durch die Bedingung $\tau = 1$ ersetzen.

## 1.2. Überwindung von Bifurkationspunkten

In diesem Kapitel wird ein Verfahren vorgestellt, mit dem man den einzelnen Ästen in einem Bifurkationspunkt entlanggehen kann. Dazu wird man zuerst den Bifurkations-

punkt berechnen, dann die Tangenten im Bifurkationspunkt bestimmen und sie dazu benutzen, den einzelnen Ästen mit dem soeben besprochenen Verfahren zu folgen.

**Bezeichnung**: Mit $u^*$ ist von jetzt an immer der zu untersuchende Bifurkationspunkt gemeint. In einem Bifurkationspunkt schneiden sich zwei Homotopiewege, $\Delta u$ und $\overline{\Delta u}$ bezeichnen die Tangenten dieser Wege in $u^*$.

### 1.2.1. Berechnung der Tangenten für den Fall n=1

Es wird jetzt gezeigt, wie man Tangenten bei $n = 1$ finden kann. Später wird der Fall, daß $n$ beliebig sein darf, auf diesen Fall zurückgeführt werden. Man nimmt vorläufig an, man hätte den Bifurkationspunkt $u^* = (x^*, \tau^*)$ schon berechnet. Wie das gemacht wird, wird ebenfalls später gezeigt werden.

$F$ sei 3-mal stetig differenzierbar. Man betrachtet die Taylorentwicklung von $F(x, \tau)$ um $(x^*, \tau^*)$:

$$\begin{aligned} F(x,\tau) &= F(x^*,\tau^*) + (F_x(x^*,\tau^*), F_\tau(x^*,\tau^*)) \begin{pmatrix} x-x^* \\ \tau-\tau^* \end{pmatrix} \\ &\quad + \frac{1}{2}(x-x^*, \tau-\tau^*) \begin{pmatrix} F_{xx} & F_{x\tau} \\ F_{x\tau} & F_{\tau\tau} \end{pmatrix}_* \begin{pmatrix} x-x^* \\ \tau-\tau^* \end{pmatrix} + O(\| \begin{pmatrix} x-x^* \\ \tau-\tau^* \end{pmatrix} \|^3), \end{aligned}$$

wobei $\begin{pmatrix} F_{xx} & F_{x\tau} \\ F_{x\tau} & F_{\tau\tau} \end{pmatrix}_* := \begin{pmatrix} F_{xx}(x^*,\tau^*) & F_{x\tau}(x^*,\tau^*) \\ F_{x\tau}(x^*,\tau^*) & F_{\tau\tau}(x^*,\tau^*) \end{pmatrix}$.

Man betrachtet nun einen Homotopieweg $(x(s), \tau(s))$ durch $u^* = (x^*, \tau^*)$. Wegen $\mathrm{Rg}(H'(u^*)) = \mathrm{Rg}(F_x(x^*,\tau^*), F_\tau(x^*,\tau^*)) = n - 1 = 0$ muß sowohl $F_x(x^*,\tau^*) = 0$ als auch $F_\tau(x^*,\tau^*) = 0$ gelten, da man sonst $\mathrm{Rg}(H'(u^*)) = 1$ hätte. Die Tangente $\begin{pmatrix} \Delta x \\ \Delta\tau \end{pmatrix}$ des Homotopieweges erfüllt daher die Gleichung

$$(\Delta x, \Delta\tau) \begin{pmatrix} F_{xx} & F_{x\tau} \\ F_{x\tau} & F_{\tau\tau} \end{pmatrix}_* \begin{pmatrix} \Delta x \\ \Delta\tau \end{pmatrix} = 0.$$

Man zerlegt $\begin{pmatrix} F_{xx} & F_{x\tau} \\ F_{x\tau} & F_{\tau\tau} \end{pmatrix}_*$ in eine orthogonale Matrix $W$ und eine Diagonalmatrix und erhält

$$(\Delta x, \Delta\tau)\, W^T \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} W \begin{pmatrix} \Delta x \\ \Delta\tau \end{pmatrix} = 0 \tag{1.1}$$

$\lambda_1$ und $\lambda_2$ sind die Eigenwerte von $\begin{pmatrix} F_{xx} & F_{x\tau} \\ F_{x\tau} & F_{\tau\tau} \end{pmatrix}_*$.

Falls sowohl $\lambda_1$ als auch $\lambda_2$ größer als 0 sind, ist $(x^*, \tau^*)$ ein lokales Minimum, falls beide kleiner als 0 sind ein lokales Maximum. Daher müssen $\lambda_1$ und $\lambda_2$ verschiedene Vorzeichen haben, d.h. $\lambda_1\lambda_2 < 0$. Man definiert nun $w := \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} := W \begin{pmatrix} \Delta x \\ \Delta\tau \end{pmatrix}$. Für $w$ gilt dann wegen (1.1)

$$w_1^2\lambda_1 + w_2^2\lambda_2 = 0 \tag{1.2}$$

Man bekommt linear unabhängige Lösungen $w$ und $\overline{w}$ von (1.2) durch

$$w := \begin{pmatrix} +\sqrt{|\lambda_2|} \\ +\sqrt{|\lambda_1|} \end{pmatrix}, \quad \overline{w} := \begin{pmatrix} +\sqrt{|\lambda_2|} \\ -\sqrt{|\lambda_1|} \end{pmatrix},$$

und linear unabhängige Lösungen $\begin{pmatrix} \Delta x \\ \Delta\tau \end{pmatrix}$ und $\begin{pmatrix} \overline{\Delta x} \\ \overline{\Delta\tau} \end{pmatrix}$ durch

$$\begin{pmatrix} \Delta x \\ \Delta\tau \end{pmatrix} = W^T w \quad \text{und} \begin{pmatrix} \overline{\Delta x} \\ \overline{\Delta\tau} \end{pmatrix} = W^T \overline{w}$$

**Bemerkung 1**: Es gibt also für den Fall $n = 1$ zwei linear unabhängige Tangenten. Man wird später sehen, daß dies auch für beliebiges $n$ der Fall sein wird.

**Bemerkung 2**: Um lokale Maxima und Minima auszuschließen, war es notwendig, zusätzlich zur Rangbedingung noch die Bedingung $\lambda_1\lambda_2 < 0$ zu fordern. Man definiert daher für den Fall $n = 1$:
Ein Punkt $u^* = (x^*, \tau^*)$ ist ein *strenger Bifurkationspunkt*
$:\Longleftrightarrow \mathrm{Rg}(H'(u^*)) = 0$, und für die Eigenwerte $\lambda_1$ und $\lambda_2$ der Hesseschen Matrix von $H$, ausgewertet in $u^*$, gilt: $\lambda_1\lambda_2 < 0$.
Diese Definition wird später auf beliebige $n$ erweitert werden.

## 1.2.2. Berechnung der Tangenten im allgemeinen Fall

### 1.2.2.1. Berechnung des Kerns von $H'(u_0)$

Man nimmt wiederum an, man hätte den Bifurkationspunkt $u^*$ schon berechnet. Sei $u(s)$ ein Homotopieweg, der bei $s = 0$ durch $u^*$ geht. Durch Differenzieren von $H(u(s)) = 0$ erhält man

$$H'(u(s))u'(s) = 0 \tag{1.3}$$

Da dies insbesondere für $s = 0$ gilt, liegt $u'(0) = \Delta u$ im Kern von $H'(u^*)$. Eine genaue Betrachtung des Kerns scheint daher lohnenswert zu sein. Eine Basis von $\mathrm{Ker}(H'(u^*))$ kann mit Hilfe der QR-Zerlegung von $H'(u^*)$ gefunden werden:

$$H'(u^*) = QRP \tag{1.4}$$

Da der Rang von $H'(u^*)$ nicht mehr $n$, sondern nur mehr $n-1$ ist, hat $R$ jetzt folgende Gestalt

$$R = \begin{pmatrix} \alpha_1 & * & & & & * \\ 0 & \alpha_2 & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \alpha_{n-1} & * & * \\ 0 & & & 0 & 0 & 0 \end{pmatrix} \tag{1.5}$$

Ein Vektor $v$ ist aus dem Kern von $H'(u^*)$
$\iff H'(u^*)v = 0$
$\iff QRPv = 0$
$\iff R(Pv) = 0$

$$\iff \begin{pmatrix} \alpha_1 & * & & & & * \\ 0 & \alpha_2 & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \alpha_{n-1} & * & * \\ 0 & & & 0 & 0 & 0 \end{pmatrix} (Pv) = 0$$

Man kann also die letzten beiden Komponenten von $Pv$ frei wählen. Die restlichen Komponenten sind dann durch

$$\begin{pmatrix} \alpha_1 & * & & * \\ 0 & \alpha_2 & \ddots & \\ & \ddots & \ddots & * \\ 0 & & 0 & \alpha_{n-1} \end{pmatrix} \begin{pmatrix} (Pv)_1 \\ \vdots \\ (Pv)_{n-1} \end{pmatrix} = -\left(R_{1:n-1,n}\right)(Pv)_n - \left(R_{1:n-1,n+1}\right)(Pv)_{n+1} \tag{1.6}$$

eindeutig bestimmt. Um eine Basis $v_1, v_2$ vom Kern zu berechnen, setzt man etwa

$$\begin{pmatrix} (Pv_1)_n \\ (Pv_1)_{n+1} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} (Pv_2)_n \\ (Pv_2)_{n+1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

und berechnet die ersten $n-1$ Komponenten von $Pv$ aus (1.6) und erhält $v$ aus $Pv$ durch $v = P^T(Pv)$. Man hat also eine Darstellung der Tangenten $\Delta u = y_1 v_1 + y_2 v_2$

gefunden. Insbesondere kann es wegen
$\dim(\mathrm{Ker}(H'(u^*))) = 2$ auch nur zwei linear unabhängige Tangenten in $u^*$ geben.

**Bezeichnung**: $V$ sei die $((n+1)\times 2)$-Matrix, die aus $v_1$ und $v_2$ besteht: $V := (v_1, v_2)$.

#### 1.2.2.2. Berechnung von z

Ein weiteres Hilfsmittel kann aus $H'(u^*)^T$ gewonnen werden. Wegen $\mathrm{Rg}(H'(u^*)) = n-1$ ist $H'(u^*)^T$ eine $((n+1)\times n)$-Matrix mit Rang $n-1$ und
$\dim(\mathrm{Ker}(H'(u^*)^T)) = n - \mathrm{Rg}(H'(u^*)^T) = n - (n-1) = 1$
Es gibt daher einen eindeutig bestimmten, normierten Vektor $z$ mit $H'(u^*)^T z = 0$. Durch Transponieren erhält man folgende wichtige Hilfsgleichung

$$z^T H'(u^*) = 0 \tag{1.7}$$

Man beachte, daß $z$ nicht wie $u$ aus $n+1$, sondern nur aus $n$ Komponenten besteht.

$z$ kann, ebenso wie $\Delta u_0$ im vorigen Kapitel, mit Hilfe der QR-Zerlegung ((1.4) und (1.5)) berechnet werden.
$0 = z^T H'(u^*) = z^T QRP$
$\Longleftrightarrow (z^T Q)R = 0$

$$\Longleftrightarrow (z^T Q) \begin{pmatrix} \alpha_1 & * & & & & * \\ 0 & \alpha_2 & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \alpha_{n-1} & * & * \\ 0 & & & 0 & 0 & 0 \end{pmatrix} = 0$$

Man kann leicht überprüfen, daß $z^T Q = (0, \ldots, 0, 1)$ die Gleichung erfüllt. Es gilt daher für $z$:

$$z = Q \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

Man braucht also nichts Neues berechnen, weil $z$ gerade die letzte Spalte von $Q$ ist. Insbesondere ist $z$ bereits normiert, da $Q$ orthonormal ist.

#### 1.2.2.3. Zurückführung auf den Fall n=1

Um die Tangentenberechnung auf den Fall $n = 1$ zurückzuführen, versucht man, $H$ so zu verändern, daß man eine Funktion $G : \mathbb{R}^2 \longrightarrow \mathbb{R}$ erhält, ohne Information über das Bifurkationsverhalten zu verlieren. Wie oben gezeigt, hat man eine Aufspaltung von $\mathbb{R}^n$

$$\mathbb{R}^n = \mathrm{Bi}(H'(u^*)) \oplus \mathbb{R}z.$$

Der Kern von $H'(u^*)$, in dem $\Delta u$ und $\overline{\Delta u}$ enthalten sind, wird durch $H$ auf $\mathbb{R}z$ abgebildet, man verliert daher durch Projektion auf diesen Unterraum keine Information über den Kern und daher auch keine über $\Delta u$ und $\overline{\Delta u}$. Man erhält also im ersten Schritt

$$\tilde{G} : \mathbb{R}^{n+1} :\longrightarrow \mathbb{R} \qquad u \mapsto z^T H(u)$$

Im zweiten Schritt schränkt man den Definitionsbereich so ein, daß man im Punkt $u^*$ nur mehr in Richtung des Kerns gehen kann. Das geschieht dadurch, daß man für $u$ nur mehr Werte der Form $u = u^* + Vy,\ y \in \mathbb{R}$ zuläßt. Somit ergibt sich folgende Funktion:

**Definition**: $G$: $\mathbb{R}^2 :\longrightarrow \mathbb{R} \quad y \mapsto \tilde{G}(u^* + Vy) = z^T H(u^* + Vy)$
Ein Bifurkationsunkt $u^*$ heißt *strenger Bifurkationspunkt* von $H$
$:\Longleftrightarrow u^*$ ist ein Bifurkationspunkt von $H$, 0 ist dann ein Bifurkationspunkt von $G$, und für die Eigenwerte $\lambda_1$ und $\lambda_2$ von $G''(0)$ gilt: $\lambda_1\lambda_2 < 0$.

**Bemerkung**: Man kann leicht nachprüfen, daß diese Definfition mit der vorhergehenden Definition für den Fall $n = 1$ verträglich ist.

Der nächste Satz zeigt, wie man aus $G$ die Tangenten $\Delta u$ und $\overline{\Delta u}$ berechnen kann:

**Satz 1.2:** $u^*$ sei ein strenger Bifurkationspunkt mit den linear unabhängigen Tangenten $\Delta u$ und $\overline{\Delta u}$.
Dann ist $y = 0$ ein strenger Bifurkationspunkt von $G(y) = 0$ mit den Tangenten $\Delta y$ und $\overline{\Delta y}$, für die $\Delta u = V\Delta y$ bzw. $\overline{\Delta u} = V\overline{\Delta y}$ gilt.

**Beweis**: Wegen $G(0) = z^T \underbrace{H(u^*)}_{=0} = 0$ ist $y = 0$ eine Lösung von $G(y) = 0$.
Außerdem gilt wegen (1.7) $G'(0) = z^T H'(u^* + Vy)|_{y=0} = z^T H'(u^*) = 0$,
womit bewiesen wäre, daß der Punkt $y = 0$ ein Bifurkationspunkt von $G$ ist. Die Forderung, daß $u^*$ ein strenger Bifurkationspunkt ist, bedeutet gerade, daß auch $y = 0$ ein strenger Bifurkationspunkt ist.
Es fehlt noch die Darstellung von $\Delta u$ und $\overline{\Delta u}$:
Zweimalige Differentiation von $G$ an der Stelle 0 liefert

$$G''(0)(\Delta y, \Delta y) = z^T H''(u^*)(V\Delta y, V\Delta y), \tag{1.8}$$

wobei $\left(z^T H''(u^*)\right)_{jk} := \sum_{i=1}^{n} z_i \frac{\partial^2 H_i}{\partial u_j \partial u_k}$.
Man betrachtet wiederum einen Homotopieweg $u(s)$ mit $u(0) = u^*$. Durch zweimalige Differentiation von $H(u(s)) = 0$ erhält man
$H''(u(s))(u'(s), u'(s)) + H'(u(s))u''(s) = 0$,
deshalb gilt

$$z^T H''(u^*)(\Delta u, \Delta u) = -\underbrace{z^T H'(u^*)}_{=0} u''(0) = 0. \tag{1.9}$$

Die Gleichung $G''(0)(\Delta y, \Delta y) = 0$ bestimmt $\Delta y$ und $\overline{\Delta y}$ bis auf einen skalaren Faktor (siehe Fall $n = 1$). Wegen (1.8) sind deshalb $V\Delta y$ und $V\overline{\Delta y}$ durch die Gleichung

$$z^T H''(u^*)(V\Delta y, V\Delta y) = 0 \tag{1.10}$$

und eine Normierung eindeutig gegeben. Andererseits erfüllen wegen (1.9) auch $\Delta u$ und $\overline{\Delta u}$ diese Gleichung. Wenn man auch $\Delta u$ und $\overline{\Delta u}$ normiert, müssen $V\Delta y$ und $\Delta u$ bzw. $V\overline{\Delta y}$ und $\overline{\Delta u}$ übereinstimmen.

□

### 1.2.3. Berechnung des Bifurkationspunktes

Wie schon vorher gezeigt wurde, erfüllt ein Bifurkationspunkt folgende Gleichungen:

$$\begin{aligned} H(u^*) &= 0 \\ z^T H'(u^*) &= 0 \\ z^T z &= 1 \end{aligned} \tag{1.11}$$

Im Gegensatz zum Kapitel über die Fortsetzungsverfahren fehlt jetzt nicht eine Gleichung, sondern eine Unbekannte, (1.11) besteht nämlich aus $2n+2$ Gleichungen für die $2n+1$ Unbekannten $u^*$ und $z$. In diesem Fall wird man deshalb statt einer Gleichung eine Unbekannte $\alpha$ zusätzlich verwenden.

**Bezeichnung**: Das Gleichungssystem

$$\begin{aligned} (z^T H'(u^*))^T &= 0 \\ H(u^*) + \alpha z &= 0 \\ \frac{1}{2}(z^T z - 1) &= 0 \end{aligned} \tag{1.12}$$

heißt *erweitertes System von Moore*.
Für $\alpha = 0$ ist das gerade das Gleichungssystem (1.11). Man wird also $\alpha = 0$ fordern müssen.

**Satz 1.3:** Sei $(u^*, z^*, \alpha^* = 0)$ eine Lösung des erweiterten Systems von Moore und $u^*$ ein strenger Bifurkationspunkt.
Dann ist die Jacobimatrix $J$ von (1.12), ausgewertet in $(u^*, z^*, \alpha^* = 0)$, invertierbar. Man kann daher das Newtonverfahren zur Berechnung von $(u^*, z^*, \alpha^*)$ verwenden.

**Beweis**: Zuerst wird $J$ berechnet:

$$\frac{\partial}{\partial u_k}\left(\left((z^T H'(u))^T\right)_j\right) = \frac{\partial}{\partial u_k}\left(\sum_{i=1}^{n} z_i H'_{ij}(u)\right) = \frac{\partial}{\partial u_k}\left(\sum_{i=1}^{n} z_i \frac{\partial H_i}{\partial u_j}(u)\right) = \sum_{i=1}^{n} z_i \frac{\partial^2 H_i}{\partial u_j \partial u_k}(u)$$

$\left(\sum_{i=1}^{n} z_i \frac{\partial^2 H_i}{\partial u_j \partial u_k}(u)\right)_{1\le j,k\le n+1} = z^T H''(u)$
$\frac{\partial}{\partial z_k}\left(\left((z^T H'(u))^T\right)_j\right) = \frac{\partial}{\partial z_k}\left(\sum_{i=1}^{n} z_i H'_{ij}\right) = \sum_{i=1}^{n} \delta_{ik} H'_{ij} = H'_{kj}$
$\frac{\partial}{\partial \alpha}\left(\left((z^T H'(u))^T\right)_j\right) = 0$
Die ersten $n+1$ Zeilen von $J$ lauten daher $\begin{pmatrix} z^T H''(u) & H'(u)^T & 0 \end{pmatrix}$.
Die restlichen Zeilen kann man direkt hinschreiben und erhält

$$J = \begin{pmatrix} z^T H''(u) & H'(u)^T & 0 \\ H'(u) & \alpha I_n & z \\ 0 & z^T & 0 \end{pmatrix}$$

$J_* := J(u^*, z^*, \alpha^*)$ ist eine $(2n+2) \times (2n+2)$ -Matrix, es genügt deshalb, zu zeigen, daß der Kern von $J_*$ nur aus dem Nullvektor besteht.
Sei $\begin{pmatrix} \Delta v \\ \Delta z \\ \Delta \alpha \end{pmatrix}$ ein Vektor mit

$$J_* \begin{pmatrix} \Delta v \\ \Delta z \\ \Delta \alpha \end{pmatrix} = 0 \tag{1.13}$$

Zeige: $\Delta v = 0$, $\Delta z = 0$, $\Delta \alpha = 0$.
Wenn man die Gleichungen $n+2$ bis $2n+1$ von (1.13) von links mit $(z^*)^T$ multipliziert, erhält man wegen $\alpha^* = 0$:

$$\underbrace{(z^*)^T H'(u^*)}_{=0} \Delta v + \underbrace{(z^*)^T z^*}_{=1} \Delta \alpha = \Delta \alpha = 0$$

Setzt man $\Delta v = V \Delta y$ in die ersten $n+1$ Gleichungen von (1.13) ein und multipliziert von links mit $(Vy)^T$, y beliebig, ergibt sich

$$\underbrace{(Vy)^T (z^*)^T H''(u^*) V \Delta y}_{=(z^*)^T H''(u^*)(V\Delta y, Vy)} + y^T \underbrace{V^T H'(u^*)^T}_{=0} \Delta z = 0$$

Der übriggebliebene Term ist gerade $G''(0)(\Delta y, y)$, man hat also die Gleichung

$$y^T G''(0)\Delta y = 0$$

Da $y$ beliebig ist, muß die lineare Abbildung, die $y$ auf $y^T G''(0)\Delta y$ abbildet, die Nullabbildung und daher $G''(0)\Delta y = 0$ sein. Laut Voraussetzung sind aber die Eigenwerte $\lambda_1$ und $\lambda_2$ von $G''(0)$ von 0 verschieden (sonst wäre $\lambda_1\lambda_2$ nicht kleiner als 0), deshalb muß $\Delta y = 0$ sein.

$$\Longrightarrow \Delta v = V\Delta y = 0$$

Wenn man unter Verwendung von $\Delta v = 0$ die ersten $n+1$ Gleichungen anschreibt, erhält man $H'(u^*)^T\Delta z = 0$, d.h. $\Delta z$ ist im Kern von $H'(u^*)^T$. Deshalb gibt es ein $\beta$ mit $\Delta z = \beta z^*$. Diese Darstellung wird jetzt in der letzten Gleichung von (1.13) verwendet, und man bekommt wegen $\|z^*\| = 1$

$$(z^*)^T \Delta z = \beta\|z^*\|^2 = 0 \quad \Longrightarrow \beta = 0 \quad \Longrightarrow \Delta z = \beta z^* = 0$$

□

### 1.2.4. Algorithmus 2 (Überwindung von Bifurkationspunkten)

$H$ sei 3-mal stetig differenzierbar, $u_0$ mit $H(u_0) = 0$ und $\lambda$ gegeben.

**Schritt 1:**
Stelle den Rang von $H'(u_0)$ mit Hilfe der QR-Zerlegung fest.

**Schritt 2:**

Falls $\mathrm{Rg}(H'(u_0)) = n$:
Verwende Algorithmus 1, um $u$ zu berechnen.

Falls $\mathrm{Rg}(H'(u_0)) = n-1$:
Berechne $u^*$ durch Anwendung des Newtonverfahrens
auf das erweiterte System von Moore.
Finde die Tangenten $\Delta u$ und $\overline{\Delta u}$.
Solange $\mathrm{Rg}(H'(u_0)) = n-1$:
Verwende Algorithmus 1, um $u$ zu berechnen,
setze jedoch im ersten Schritt des Algorithmus $\Delta u_0 := \Delta u$ bzw. $\overline{\Delta u}$.

**Schritt 3:**
Setze $u_0 := u$.
Falls $\tau = 1$ erreicht wurde, beende das Verfahren.
Sonst gehe zurück zu Schritt 1.

In der Praxis wird der Rang von $H'(u)$ numerisch bestimmt (siehe (A.2)), es kann daher vorkommen, daß der Rang von $H'(u)$ nicht nur im Bifurkationspunkt, sondern auch in einer kleinen Umgebung $n - 1$ ist. Die Tangente im Bifurkationspunkt wird dann als Näherung für die Tangenten in dieser Umgebung genommen.

# 2. Berechnung von Bifurkationsdiagrammen mit Gröbnerbasen

In diesem Kapitel werden so wie im vorigen Kapitel Gleichungssysteme

$$H(u) = 0, \quad H : \mathbb{R}^{n+1} \longrightarrow \mathbb{R}^n$$

untersucht, jedoch mit der Einschränkung, daß die einzelnen Komponenten $H_i$ von $H$ Polynome in $n+1$ Variablen sein sollen. Diese Einschränkung erlaubt es, Gröbnerbasen zur Berechnung von Homotopiewegen einzusetzen. Im ersten Teil dieses Kapitels werden Gröbnerbasen vorgestellt, und im zweiten Teil wird gezeigt, wie man mit Hilfe einer Gröbnerbasis Systeme von Polynomgleichungen lösen kann.

**Bezeichnungen**:
1.) Im folgenden sei $\mathbb{K}$ ein Körper und $\mathbb{K}[y] := \mathbb{K}[y_1, \ldots, y_m]$ der Polynomring in den Variablen $y_1, \ldots, y_m$ mit Koeffizienten in $\mathbb{K}$.
In den meisten Fällen wird $\mathbb{K}$ der Körper der reellen Zahlen und $m = n+1$ sein.
2.) Ein *Ideal* $I = {}_{\mathbb{K}[y]}\langle f_1, \ldots, f_k\rangle \subseteq \mathbb{K}[y]$ ist die Menge $\{a_1 f_1 + \ldots + a_k f_k;\ a_i \in \mathbb{K}[y]\}$ aller polynomialen Kombinationen der Erzeugenden $f_1, \ldots, f_k$.
$\{f_1, \ldots, f_k\}$ heißt *Erzeugendensystem des Ideals I.*
Statt $I$ ist ein Ideal in $\mathbb{K}[y]$ schreibt man auch $I \trianglelefteq \mathbb{K}[y]$.
3.) $F$ sei eine Teilmenge von $\mathbb{K}[y]$.
$N_{\mathbb{K}^m}(F)$ und $N_{\mathbb{K}^m}(f)$ sind folgendermaßen definiert:

$$N_{\mathbb{K}^m}(f) := \{u \in \mathbb{K}^m \mid f(u) = 0\} \text{ und } N_{\mathbb{K}^m}(F) := \bigcap_{f \in F} N_{\mathbb{K}^m}(f)$$

$N_{\mathbb{K}^m}(F)$ heißt *Nullstellenmenge von F.*
Ist $I = {}_{\mathbb{K}[y]}\langle F\rangle$, so ist $N_{\mathbb{K}^m}(I) = N_{\mathbb{K}^m}(F)$.
Der Einfachheit halber wird statt $N_{\mathbb{K}^m}(F)$ oft nur $N(F)$ geschrieben.
4.) Sei $i \in \mathbb{N}^m$. Dann wird $y^i$ durch $y^i := y_1^{i_1} \cdot \ldots \cdot y_m^{i_m}$ definiert.
$P := \{y^i \mid\ i \in \mathbb{N}^m\} \subseteq \mathbb{K}[y]$ sei die Menge der Potenzprodukte in $\mathbb{K}[y]$.

# 2.1. Gröbnerbasen

## 2.1.1. Termordnungen

**Definition**: Eine Relation $H \subseteq P \times P$ auf $P$ heißt *Ordnung* auf $P$ (und man schreibt meist $p \leq q$ für $(p,q) \in H$ und $p < q$ für $p \leq q$ und $p \neq q$), wenn folgende Bedingungen erfüllt sind:
1.) $p \leq p$ für alle $p \in P$.
2.) Ist $p \leq q$ und $q \leq p$, so folgt $p = q$.
3.) Ist $p \leq q$ und $q \leq r$, so folgt $p \leq r$.
Gilt zusätzlich
4.) Für je zwei $p, q \in P$ ist stets $p \leq q$ oder $q \leq p$,
so spricht man von einer *totalen Ordnung*.
Eine totale Ordnung $<$ auf $P$ heißt *Termordnung*
$:\Longleftrightarrow \ \forall p \in P \backslash \{1\} : \ 1 < p \ \text{ und } \ \forall p,q,r \in P : \ p < q \implies rp < rq.$

**Definition und Beispiel**: Die *Lexikographische Ordnung (mit $y_1 > \ldots > y_m$)* $<_{lex}$ ist definiert durch
$y^i <_{lex} y^j :\Longleftrightarrow \ \exists\, k \leq m-1$ so, daß $i_1 = j_1, \ldots, i_k = j_k$ und $i_{k+1} < j_{k+1}$,
und ist eine Termordnung.

**Bemerkung**: Im Programm ist $m = n + 1$, und es werden sowohl die lexikographische Ordnung mit $y_1 > \ldots > y_m$ als auch die lexikographische Ordnung mit $y_1 > \ldots > y_{m-2} > y_m > y_{m-1}$ verwendet.

**Definitionen**: Sei $0 \neq f = \sum_{p \in P} c_p(f) p \in \mathbb{K}[y]$. Dann heißt:

$$\begin{aligned} \operatorname{supp}(f) &:= \{p \in P \mid c_p(f) \neq 0\} \ \textit{Träger von } f, \\ \operatorname{lt}(f) &:= \max_{<} \operatorname{supp}(f) \ \textit{Leitterm von } f \textit{ bezüglich } <, \\ \operatorname{lk}(f) &:= c_{\operatorname{lt}(f)} \ \textit{Leitkoeffizient von } f, \\ \operatorname{lt}(F) &:= \{\operatorname{lt}(f) \mid f \in F, \ f \neq 0\}. \end{aligned}$$

**Bemerkung**: Der Leitterm von f hängt von der gewählten Termordnung ab.

## 2.1.2. Division mit Rest

**Satz 2.1:** (Division mit Rest):
Sei $g \in \mathbb{K}[y], F \subseteq \mathbb{K}[y]$ und $<$ eine Termordnung.

Dann gibt es eine Familie $(h_f)_{f\in F}$ in $\mathbb{K}[y]$ so, daß folgende Bedingungen erfüllt sind:

1.) Es gilt entweder $g = \sum_{f\in F} h_f f$

oder $g \neq \sum_{f\in F} h_f f$, wobei $\mathrm{lt}(g - \sum_{f\in F} h_f f) \notin {}_{\mathbb{K}[y]}\langle \mathrm{lt}(F)\rangle$

2.) $\max_{f\in F} \mathrm{lt}(h_f f) = \mathrm{lt}(g)$ oder $h_f = 0$.

Die Polynome $h_f,\ f \in F$ können folgendermaßen berechnet werden:
Setze zuerst $h_f := 0$.
Solange $g \neq 0$ und $\mathrm{lt}(g) \in {}_{\mathbb{K}[y]}\langle \mathrm{lt}(F)\rangle$, d.h. $\exists f \in F, \exists p \in P$ mit $\mathrm{lt}(g) = p \cdot \mathrm{lt}(f)$:
Ersetze $g$ durch $g - \mathrm{lk}(g)\mathrm{lk}(f)^{-1} pf$ und $h_f$ durch $h_f + \mathrm{lk}(g)\mathrm{lk}(f)^{-1} p$.

**Beweis**: siehe [3] oder [5].

**Definition**: Sei $(h_f)_{f\in F}$ eine Familie in $\mathbb{K}[y]$ wie im vorigen Satz.
Dann heißt $r(g, F) := g - \sum_{f\in F} h_f f$ ein *Rest von g nach Division durch F.*

**Beispiel**: $g := y_1^3 + y_1^2 y_2 + 1,\ f_1 := y_1^2,\ f_2 := y_1 y_2 + 1,\ F := \{f_1, f_2\}$.
$<$ sei die lexikographische Ordnung mit $y_1 > y_2$, daher ist $\langle \mathrm{lt}(F)\rangle = \langle y_1^2, y_1 y_2\rangle$.

$$\begin{array}{ll} g - y_1 f_1 = y_1^2 y_2 + 1, & h_{f_1} := y_1 \\ (g - y_1 f_1) - y_2 f_1 = 1 \notin \langle \mathrm{lt}(F)\rangle, & h_{f_1} := h_{f_1} + y_2 = y_1 + y_2 \end{array}$$

Man erhält: $h_{f_1} = y_1 + y_2,\ h_{f_2} = 0,\ r(g, F) = 1$.
Die Division mit Rest ist nicht eindeutig bestimmt, man kann $g$ auch folgendermaßen durch F dividieren:

$$\begin{array}{ll} g - y_1 f_1 = y_1^2 y_2 + 1, & h_{f_1} := y_1 \\ (g - y_1 f_1) - y_1 f_2 = 1 - y_1,\ \mathrm{lt}(1 - y_1) = y_1 \notin \langle \mathrm{lt}(F)\rangle, & h_{f_2} := y_1 \end{array}$$

Man bekommt nun: $h_{f_1} = y_1,\ h_{f_2} = y_1,\ r(g, F) = -y_1 + 1$.
Es ist also auch der Rest der Division nicht eindeutig bestimmt.

## 2.1.3. Gröbnerbasen und ihre Eigenschaften

**Definitionen**:
Eine endliche Teilmenge $F$ eines Ideals $I \trianglelefteq \mathbb{K}[y]$ heißt *Gröbnerbasis von I bezüglich* $<$
$:\Longleftrightarrow {}_{\mathbb{K}[y]}\langle \mathrm{lt}(F)\rangle = {}_{\mathbb{K}[y]}\langle \mathrm{lt}(I)\rangle$.
Eine endliche Teilmenge $F$ von $\mathbb{K}[y]$ heißt *Gröbnerbasis bezüglich* $<$
$:\Longleftrightarrow$ F ist eine Gröbnerbasis von ${}_{\mathbb{K}[y]}\langle F\rangle$.

**Bemerkungen**:
1.) Sei $F'$ endlich, $I \supseteq F' \supseteq F$ und $F$ eine Gröbnerbasis von $I$.
Dann ist auch $F'$ eine Gröbnerbasis von $I$.
2.)Es läßt sich zeigen, daß jedes Ideal ausgenommen das Nullideal ${}_{\mathbb{K}[y]}\langle 0\rangle$ (lt(0) ist ja nicht definiert) eine Gröbnerbasis besitzt (siehe etwa [3] oder [5]).

**Satz 2.2:** $F$ sei eine endliche Teilmenge eines Ideals $I$ in $\mathbb{K}[y]$ und $0 \notin F$.
Dann sind folgende Aussagen zueinander äquivalent:
1.) $F$ ist eine Gröbnerbasis von $I$.
2.) Für alle $g \in I$ sind alle Reste von $g$ nach Division durch $F$ gleich 0.
3.) Für alle $g \in I$ ist ein Rest von $g$ nach Division durch $F$ gleich 0.

Beweis: siehe [3] oder [5].

**Folgerungen**:
1.) Man kann mit Hilfe einer Gröbnerbasis $F$ von $I$ entscheiden, ob ein Polynom $g$ in $I$ liegt: $g \in I \iff$ ein Rest von $g$ nach Division durch $F$ ist 0.
2.) Ist $F$ eine Gröbnerbasis von $I$, dann ist F auch ein Erzeugendensystem von $I$ (Aussage 3 des Satzes).

### 2.1.4. Berechnung von Gröbnerbasen

**Definitionen**: Für $i, j \in \mathbb{N}^m$ sei $\mathrm{kgV}(y^i, y^j) := y_1^{\max(i_1,j_1)} \cdot \ldots \cdot y_m^{\max(i_m,j_m)}$.
Sei $<$ eine Termordnung.
Für $f, g \in K[y]\backslash\{0\}$ sei $S(f,g) := \mathrm{lk}(g) \cdot p \cdot f - \mathrm{lk}(f) \cdot q \cdot g$, wobei $p, q \in P$ so zu wählen sind, daß $p \cdot \mathrm{lt}(f) = q \cdot \mathrm{lt}(g) = \mathrm{kgV}(\mathrm{lt}(f), \mathrm{lt}(g))$ erfüllt ist.
$S(f,g)$ heißt *S-Polynom von $f$ und $g$ bezüglich $<$*.

Der folgende Satz zeigt, wie man überprüfen kann, ob eine gegebene Menge eine Gröbnerbasis ist.

**Satz 2.3:** (Test auf Gröbnerbasis):
$F$ sei eine endliche Teilmenge von $\mathbb{K}[y]\backslash\{0\}$.
Dann sind folgende Aussagen äquivalent:
1.) $F$ ist eine Gröbnerbasis.
2.) Für alle $f, g \in F$ ist ein Rest von $S(f,g)$ nach Division durch $F$ gleich 0.
Man kann also in endlich vielen Schritten überprüfen, ob $F$ eine Gröbnerbasis ist.

Beweis: siehe [3] oder [5].

**Bemerkung**: In den Satz geht wesentlich ein, daß es wegen Satz 2.2 ausreicht, einen beliebigen Rest von $S(f,g)$ nach Division durch $F$ zu berechnen.

Dieser Satz ist der Schlüssel zur Berechnung einer Gröbnerbasis von ${}_{\mathbb{K}[y]}\langle F\rangle$. Falls $F$ nicht schon eine Gröbnerbasis ist, gibt es S-Polynome von Elementen von $F$, deren Reste nach Division durch $F$ nicht gleich 0 sind. Man ergänzt $F$ durch diese Reste zu einer Menge $F'$ und hofft nun, daß man eine Gröbnerbasis erhält. Falls nicht, führt man denselben Schritt mit $F'$ durch. Dieses Vorgehen wird im folgenden Algorithmus genauer formuliert:

**Satz 2.4:** (Buchberger-Algorithmus zur Berechnung von Gröbnerbasen):
Der folgende Algorithmus berechnet eine Gröbnerbasis von ${}_{\mathbb{K}[y]}\langle F\rangle$:
$F_0 := F$
Sei $F_i,\ i \geq 0$ schon berechnet.
$F_{i+1} := F_i \ \cup \ \{r(S(f,g), F_i);\ f,g \in F_i \backslash \{0\}\}$
Wenn $F_{i+1} \subseteq F_i \ \cup \ \{0\}$, dann ist $F_i$ eine Gröbnerbasis von ${}_{\mathbb{K}[y]}\langle F\rangle$.

Beweis: siehe [3] oder [5].

## 2.2. Systeme von Polynomgleichungen

Man kann mit Hilfe von Gröbnerbasen ein System von Polynomgleichungen lösen, falls es endlich viele Lösungen besitzt. Im ersten Kapitel wird daher gezeigt, wie man die Anzahl der Lösungen mit Gröbnerbasen abschätzen kann. Im zweiten Kapitel wird ein Algorithmus zur Berechnung dieser Lösungen vorgestellt, und im dritten Kapitel wird ein Fortsetzungsverfahren, das auf diesem Algorithmus basiert, entwickelt.

### 2.2.1. Wann gibt es (endlich viele) Lösungen

**Definition**: Ein Körper $\mathbb{K}$ heißt *algebraisch abgeschlossen*
$:\Longleftrightarrow$ jedes nichtkonstante Polynom einer Variablen mit Koeffizienten in $\mathbb{K}$ hat eine Nullstelle in $\mathbb{K}$. In unserem Fall wird das immer der Körper der komplexen Zahlen, der ja der algebraische Abschluß des Körpers der reellen Zahlen ist, sein.

Der folgende Satz zeigt, wann ein System von Polynomgleichungen Lösungen besitzt:

**Satz 2.5:** (Hilbertscher Nullstellensatz, 1.Teil):
Sei $\mathbb{K}$ algebraisch abgeschlossen und $F \subseteq \mathbb{K}[y]$.
Dann gilt: $N_{\mathbb{K}^m}(F) = \varnothing \iff {}_{\mathbb{K}[y]}\langle F\rangle = \mathbb{K}[y] \iff 1 \in {}_{\mathbb{K}[y]}\langle F\rangle$

Beweis: siehe [3], [5] oder [6].

**Folgerung**: Sei $\mathbb{K}$ algebraisch abgeschlossen, $F \subseteq \mathbb{K}[y]$ endlich und G eine Gröbnerbasis von ${}_{\mathbb{K}[y]}\langle F\rangle$ (bezüglich einer beliebigen Termordnung). Dann kann an G abgelesen werden, ob es eine Lösung gibt:
$N_{\mathbb{K}^m}(F) = \varnothing \iff G \cap (\mathbb{K}\backslash\{0\}) \neq \varnothing$, das heißt, es gibt eine Lösung genau dann, wenn $G$ keine Konstanten außer der Null enthält.

Nun noch ein Satz zur Abschätzung der Anzahl der Lösungen:

**Satz 2.6:** (Abschätzung der Anzahl der Lösungen):
Sei $\mathbb{K}$ algebraisch abgeschlossen, $I$ ein Ideal in $\mathbb{K}[y]$ und $G$ eine Gröbnerbasis von $I$.
Dann ist $|N(I)| \leq |P\backslash \mathrm{lt}(I)|$.
Insbesondere gilt: $\forall y_i\ \exists e_i \in \mathbb{N}$ so, daß $y_i^{e_i} = \mathrm{lt}(g_i),\ g_i \in G$ und $|N(I)| \leq \prod_{i=1}^{m} e_i$.

Beweis: siehe [5].

## 2.2.2. Berechnung von Lösungen

In diesem Kapitel wird ein Algorithmus zur Berechnung von Lösungen eines Systems von Polynomgleichungen vorgestellt. Dabei versucht man, diese Berechnung auf die Berechnung der Nullstellen von Polynomen in einer Variable zurückzuführen.

**Definitionen**: $M \neq \{0\}$ sei eine Menge von Polynomen in einer Variable $x$.
Dann heißt $ggt(M) := \max\{h \in \mathbb{K}[x] \mid h \text{ teilt alle Polynome in M}\}$ größter gemeinsamer Teiler von $M$. Bemerkung: In $\mathbb{K}[x]$ gibt es nur eine Termordnung.
$I_k := I \cap \mathbb{K}[y_k, \ldots, y_m]$ heißt *k-tes Eliminationsideal von I.*
Sei $b = (b_{k+1}, \ldots, b_m)$. Dann wird $I_k(b) \in \mathbb{K}[y_k]$ durch $I_k(b) := \{f(y_k, b) \mid f \in I_k\}$ definiert.

Der größte gemeinsame Teiler ist bei der Berechnung der Nullstellenmenge von Polynomen in einer Variable von Interesse, es gilt nämlich: Sei $F$ ein Erzeugendensystem eines Ideals $I \trianglelefteq \mathbb{K}[x]$. Dann ist

$$N_{\mathbb{K}}(I) = N_{\mathbb{K}}(ggt(I)) = N_{\mathbb{K}}(ggt(F)) \text{ und} \tag{2.1}$$

$$N_{\mathbb{K}}(I) \text{ ist endlich } \iff I \neq \{0\} \tag{2.2}$$

Die Idee zur Reduktion auf die Berechnung von Nullstellen von Polynomen in einer Variable kann man am besten anhand eines Beispiels erläutern:

**Beispiel**:
$f_1 := y_1 + 3$
$f_2 := y_2^2 - 2y_2y_3$
$f_3 := y_2^3 - 4y_2^2y_3 + 4y_2y_3^2 + 3y_3^4 - 9y_3^2 - 12$
$f_4 := y_3^4 - 3y_3^2 - 4$
$F := \{f_1, f_2, f_3, f_4\}$
$I := {}_{\mathbb{R}[y_1,y_2,y_3]}\langle F\rangle \trianglelefteq \mathbb{R}[y]$.
Man kann mit Satz 2.3 nachprüfen, daß $F$ eine Gröbnerbasis bezüglich der lexikographischen Ordnung mit $y_1 > y_2 > y_3$ ist.
Man sucht nun alle Nullstellen $b = (b_1, b_2, b_3)$ von $f_1,\ f_2,\ f_3$ und $f_4$.
$b_3$ muß eine Nullstelle aller Polynome in $I$, die nur von $y_3$ abhängen, d.h. aller Polynome in $I \cap \mathbb{K}[y_3] = I_3$, sein. Es fällt auf, daß $F$ ein Polynom enthält, das nur von $y_3$ abhängt, nämlich $f_4$. Es gilt sogar $I_3 = \langle f_4\rangle$ (siehe Satz 2.7), $b_3$ muß daher eine Nullstelle von $f_4$ sein, d.h.

$$b_3 \in \{\pm 2, \pm i\}$$

$b_2$ wiederum muß eine Nullstelle aller Polynome in $I_2 = \mathbb{K}[y_2, y_3]$ sein, in die für $y_3$ alle möglichen $b_3$-Werte eingesetzt wurden, d.h. $b_2 \in N_{\mathbb{K}}(I_2(b_3))$. Die Polynome $f_2$ und $f_3$ hängen nur von $y_2$ und $y_3$ ab, wegen Satz 2.7 gilt $I_2 = \langle f_2, f_3, f_4\rangle$. Es folgt daher, daß $b_2$ eine Nullstelle von $f_2(y_2, b_3) = y_2^2 - 2y_2b_3$ und $f_3(y_2, b_3) = y_2^3 - 4y_2^2b_3 + 4y_2b_3^2 + 3b_3^4 - 9b_3^2 - 12$ sein muß. Man muß nun alle 4 möglichen $b_3$-Werte untersuchen.
$b_3 = +2$:
Für $b_3 = +2$ erhält man die beiden Polynome $f_2(y_2, +2) = y_2^2 - 4y_2 = y(y-4)$ und $f_3(y_2, +2) = y_2^3 - 8y_2^2 + 16y_2 = y(y-4)^2$. $b_2$ muß daher für $b_3 = +2$ Nullstelle von $ggt(\{y_2^3 - 8y_2^2 + 16y_2,\ y_2^2 - 4y_2)\} = y_2(y_2 - 4)$, also entweder 0 oder 4 sein.
$b_3 = -2$:
Für $b_3 = -2$ erhält man die beiden Polynome $f_2(y_2, -2) = y_2^2 + 4y_2 = y(y+4)$ und $f_3(y_2, -2) = y_2^3 + 8y_2^2 + 16y_2 = y(y+4)^2$. $b_2$ muß daher für $b_3 = -2$ Nullstelle von

$ggt(\{y_2^3 + 8y_2^2 + 16y_2,\ y_2^2 + 4y_2)\} = y_2(y_2 + 4)$, also entweder 0 oder $-4$ sein.
$b_3 = +i$:
Für $b_3 = +i$ erhält man die beiden Polynome $f_2(y_2, +i) = y_2^3 - 2iy_2 = y(y - 2i)$ und $f_3(y_2, +i) = y_2^3 - 4iy_2^2 - 4y_2 = y(y - 2i)^2$. $b_2$ muß daher für $b_3 = +i$ Nullstelle von $ggt(\{y_2^3 - 4iy_2^2 - 4y_2,\ y_2^2 - 2iy_2)\} = y_2(y_2 - 2i)$, also entweder 0 oder $+2i$ sein.
$b_3 = -i$:
Für $b_3 = -i$ erhält man die beiden Polynome $f_2(y_2, -i) = y_2^3 + 2iy_2 = y(y + 2i)$ und $f_3(y_2, -i) = y_2^3 + 4iy_2^2 - 4y_2 = y(y + 2i)^2$. $b_2$ muß daher für $b_3 = -i$ Nullstelle von $ggt(\{y_2^3 + 4iy_2^2 - 4y_2,\ y_2^2 + 2iy_2)\} = y_2(y_2 + 2i)$, also entweder 0 oder $-2i$ sein.
Man erhält daher im zweiten Schritt

$$(b_2, b_3) \in \{(0, 2),\ (4, 2),\ (0, -2),\ (-4, -2),\ (0, i),\ (2i, i),\ (0, -i),\ (-2i, -i)\}.$$

Schließlich erhält man $I_1(b_2, b_3) = \langle y_1 + 3\rangle$ und deshalb $b_1$ als eine Nullstelle des Polynoms $y_1 + 3$, also $b_1 = -3$.
Es gilt also:

$$\begin{aligned} N_{\mathbb{C}^3}(F) &= \{(-3, 0, 2),\ (-3, 4, 2),\ (-3, 0, -2),\ (-3, -4, -2),\\ &\quad (-3, 0, i),\ (-3, 2i, i),\ (-3, 0, -i),\ (-3, -2i, -i)\}\\ \text{und } N_{\mathbb{R}^3}(F) &= \{(-3, 0, 2),\ (-3, 4, 2),\ (-3, 0, -2),\ (-3, -4, -2)\}. \end{aligned}$$

Man beachte, daß man auch komplexe Nullstellen erhält. Die komplexen Zahlen treten bei der Berechnung der Nullstellen der Polynome in einer Variablen auf. Da aber in unserem Fall nur die reellen Nullstellen von Interesse sind, wird man die komplexen Nullstellen verwerfen und deshalb in jedem Schritt nur Polynome in einer Variablen mit reellen Koeffizienten erhalten.

Man muß noch zeigen, daß das Verfahren nicht nur bei diesem einfachen Beispiel zum Ziel führt. Einerseits braucht man so wie im Beispiel ein „günstiges“ Erzeugendensystem von $I$, andererseits muß in jedem Schritt $N(I_k(b))$ endlich sein, damit der Algorithmus nur endlich viele Schritte besitzt. Zur ersten Bedingung nun folgender Satz:

**Satz 2.7:** $I$ sei ein Ideal in $\mathbb{K}[y]$, $G$ eine Gröbnerbasis von $I$ bezüglich der lexikographischen Ordnung mit $y_1 > \ldots > y_m$.
Dann gilt für $1 \le k \le m$ :
Wenn $I_k = I \cap \mathbb{K}[y_k, \ldots, y_m] \neq \{0\}$ ist, dann ist $G \cap \mathbb{K}[y_k, \ldots, y_m]$ eine Gröbnerbasis und insbesondere ein Erzeugendensystem von $I_k$.

Beweis: siehe [3] oder [5].

Zur zweiten Bedingung:
Wenn $|N(I)|$ endlich ist, gibt es nach Satz 2.6 für jedes $y_k$ ein $g \in G$ und ein $e_k \in \mathbb{N}$ mit $y_k^{e_k} = \mathrm{lt}(g)$. Man kann dieses $g$ also folgendermaßen schreiben:

$$g = cy_k^{e_k} + h_{k-1}y_k^{e_{k-1}} + \ldots + h_0, \qquad c \in \mathbb{K}\backslash\{0\},\ h_i \in \mathbb{K}[y_{k+1}, \ldots, y_m]$$

Für ein beliebiges $b \in \mathbb{K}^{m-k}$ ist dann der Leitterm von $g(y_k, b) \in \mathbb{K}[y_k]$ gerade $y_k^{e_k}$, deshalb kann $g(y_k, b)$ nicht das Nullpolynom sein. Daher ist $I_k(b) \neq \{0\}$, und wegen (2.2) ist $N_{\mathbb{K}}(I_k)$ endlich.

### 2.2.3. Algorithmus zur Lösung von Polynomgleichungssystemen

Nun formulieren wir dieses Verfahren als Algorithmus:

**Satz 2.8: Algorithmus 3** (Algorithmus zur Berechnung von Lösungen):
Es sei $I \trianglelefteq \mathbb{K}[y]$ ein Ideal von $\mathbb{K}[y]$ so, daß $|P\backslash\mathrm{lt}(I)|$ endlich ist. Dann können die Nullstellen in $\mathbb{K}^m$ mit folgendem Algorithmus berechnet werden:

**Schritt 1:**
Berechne eine Gröbnerbasis $G$ von $I$
bezüglich der lexikographischen Ordnung mit $y_1 > \ldots > y_m$
Berechne $N_m := N_{\mathbb{K}}(ggt(G \cap \mathbb{K}[y_m]))$.

**Schritt2**:
Falls $N_m = \varnothing$ : $N(I) := \varnothing$. Ende des Algorithmus.
Falls $m = 1$ : $N(I) := N_m$. Ende des Algorithmus.
Falls $N_m \neq \varnothing$ und $m \neq 1$ : setze $j := m$ und gehe zu Schritt 3.

**Schritt 3:**
Sei $N_j \subseteq \mathbb{K}^{m-j+1}$ schon berechnet.
Berechne $G_{j-1} := G \cap \mathbb{K}[y_{j-1}, \ldots, y_m]$.
Für $b = (b_j, \ldots, b_m) \in N_j$ definiere $G_{j-1}(b) := \{f(y_{j-1}, b) \mid f \in G_j\}$.
Berechne $N_{j-1}(b) := N_{\mathbb{K}}(ggt(G_{j-1}(b)))$
Setze $N_{j-1} := \{(b_{j-1}, b) \mid b \in N_j,\ b_{j-1} \in N_{j-1}(b)\}$

**Schritt 4**:
Falls $N_{j-1} = \varnothing$ : $N(I) := \varnothing$. Ende des Algorithmus.
Falls $j - 1 = 1$ : $N(I) := N_1$. Ende des Algorithmus
Falls $N_{j-1} \neq \varnothing$ und $j - 1 \neq 1$: setze $j := j - 1$ und gehe zu Schritt 3.

Beweis: siehe [5].

**Bemerkung**: In der Praxis berechnet man die reellen Nullstellen der größten gemeinsamen Teiler numerisch, etwa mit Sturmschen Ketten. Durch die numerische Berechnung werden die komplexen Nullstellen automatisch ausgesondert.

## 2.3. Entwicklung eines Fortsetzungsverfahrens

### 2.3.1. Idee des Fortsetzungsverfahrens

Im vorigen Kapitel wurde behandelt, wie man die Lösungen eines Systems von Polynomgleichungen $F$, das nur endlich viele Lösungen besitzt, berechnen kann. Wenn man einer Lösungskurve $u(s),\ u \in \mathbb{K}^{n+1}$ eines Gleichungssystems $H(u) = 0$ folgen will, versucht man daher, durch Festsetzen einer Variable ein System mit endlich vielen Lösungen zu bekommen.

Die einfachste Möglichkeit besteht darin, eine Variable $y_i$ als Parameter $\tau$ festzusetzen und eine Unterteilung $\tau_0 < \tau_1 < \ldots < \tau_N,\ N \in \mathbb{N}$ zu wählen. Dann setzt man in das Gleichungssystem $H(y) = 0$ für $y_i = \tau_j,\ j \in \{1, \ldots N\}$ ein, was einem Schnitt der Kurve mit der Hyperebene $y_i = \tau_j$ entspricht. Falls nicht ein Stück der Lösungskurve in dieser Hyperebene liegt, besitzt das Gleichungssystem $F_{i,j}(u) = 0$, wobei

$$F_{i,j} := \{y_i - \tau_j,\ H_1(y), \ldots, H_n(y)\}$$

nur endlich viele Lösungen, weshalb mit dem Algorithmus des vorigen Kapitels die Nullstellenmenge von $F_{i,j}$ berechnet werden kann. Man versucht also, die Gleichungssysteme

$$F_{i,j}(u) = 0, \qquad j = 0, \ldots, N$$

zu lösen. Da im Algorithmus Nullstellen von Polynomen in einer Variable numerisch berechnet werden, schätzt man, ähnlich wie bei den Prädiktor-Korrektor-Verfahren aus Kapitel 1.1.1, den Wert des nächsten Punktes $u_j$ mit Hilfe der zwei vorher berechneten Punkte $u_{j-1}$ und $u_{j-2}$. Der folgende Schätzwert $\hat{u}$ ist der Schnittpunkt der Geraden durch $u_{j-1}$ und $u_{j-2}$ und der Hyperebene $y_i = \tau_j$:

$$\hat{u} := u_{j-1} + \frac{\tau_j - \tau_{j-1}}{(u_{j-1})_i - (u_{j-2})_i}(u_{j-1} - u_{j-2})$$

Man sucht dann in einer Umgebung

$$V_{i,r}(\hat{u}) := \{u \in \mathbb{R}^m \mid u_i = \tau_j,\ \|u - \hat{u}\| \leq r\}$$

des geschätzten Punktes $\hat{u}$ nach $u_j$. (siehe Abb.5).

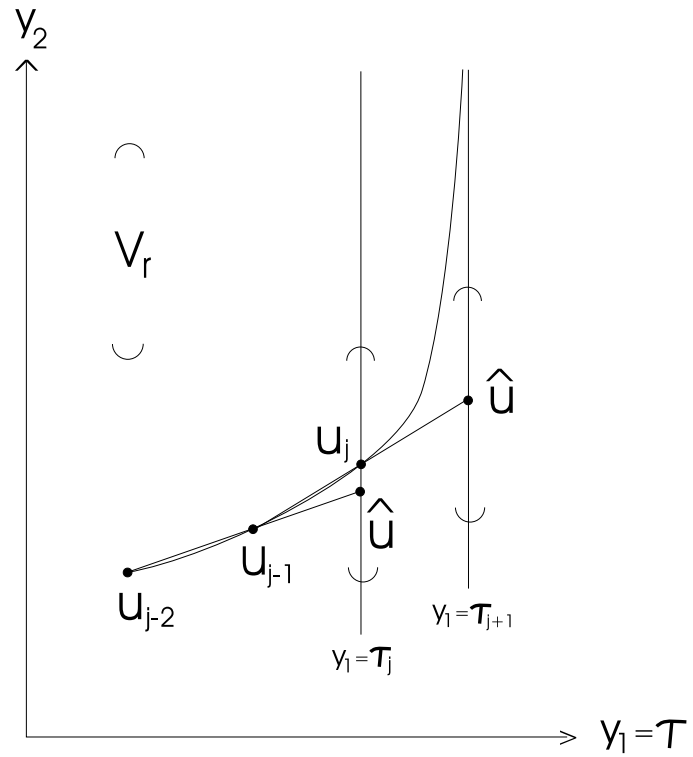


Abb.5 Verfahren ohne Schrittweitensteuerung und ohne Parameterwechsel

### 2.3.2. Schrittweitensteuerung und Parameterwechsel

In diesem Teil werden Probleme des Verfahrens aufgezeigt. Um diese zu beheben, werden eine Schrittweitensteuerung und ein Parameterwechsel verwendet. Im folgenden bezeichne $\tau$ jeweils den aktuellen Parameter.

Wie man an Abbildung 5 sehen kann, scheitert man mit einer vorgegebenen Unterteilung $\tau_0 < \tau_1 < \ldots < \tau_N,\ N \in \mathbb{N}$ und einem konstant großen Bereich $V_{i,r}$, in dem nach einer Lösung gesucht wird, schon in zwei Dimensionen bei steilen Anstiegen der Lösungskurve. Bei einer Verkleinerung der Schrittweite könnte man aber wieder eine Lösung finden. In einem verbesserten Verfahren wird daher die Schrittweite so lange verkleinert, bis eine Lösung gefunden wird (siehe Abb.6).
Die Verkleinerung der Schrittweite bewirkt allerdings eine größere Rechenzeit, da man mit einem Schritt nicht mehr so weit kommt. Ein größeres Problem ist jedoch, daß man an Umkehrpunkten nicht vorbeikommt. In der Nähe von Umkehrpunkten erhält man im Suchbereich $V_{i,r}$ mehrere Lösungen. Wenn man nur eine Lösung bekommen will, muß man die Schrittweite immer wieder verkleinern (siehe Abb.7).
Man könnte versuchen, zu einer Lösung zu springen, die vom vorher berechneten Punkt möglichst weit weg entfernt ist, und hoffen, damit um die Kurve zu kommen. Es gibt

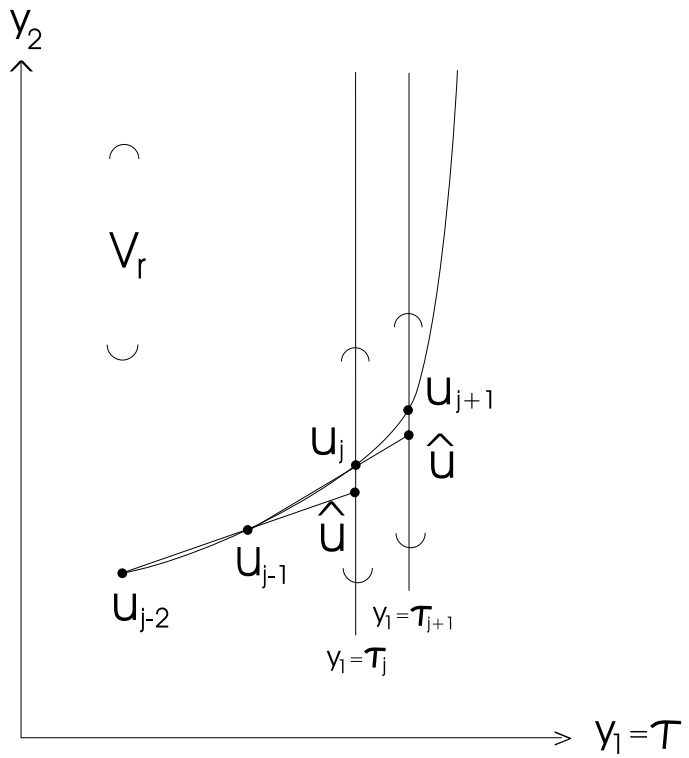


Abb.6 Verkleinerung der Schrittweite bei großen Anstiegen

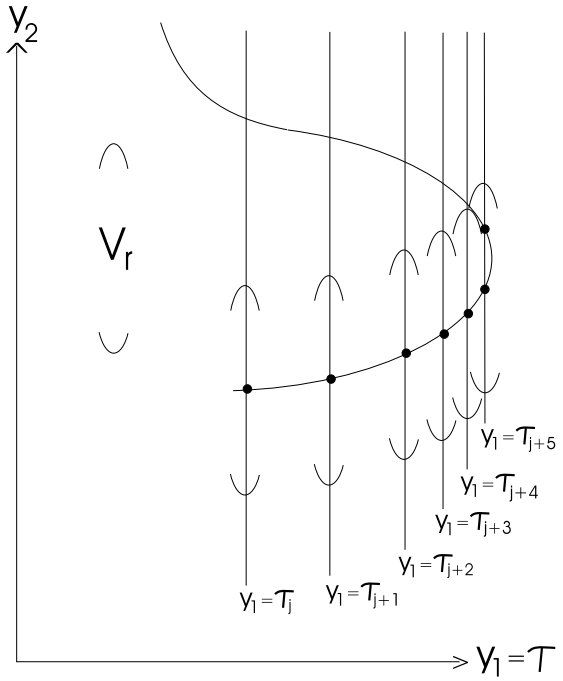


Abb.7 Kurve mit einem Umkehrpunkt ohne Parameterwechsel

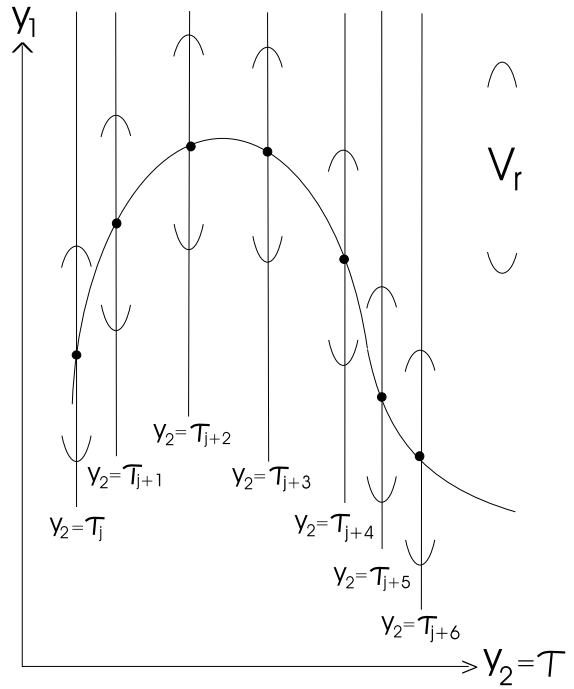


Abb.8 Kurve mit einem Umkehrpunkt nach Parameterwechsel

jedoch eine einfachere und sicherere Methode. Die Prädiktor-Korrektor-Verfahren aus Kapitel 1 scheiterten ebenfalls bei Umkehrpunkten. Die Lösung bestand darin, die Bedeutung von $y_i$ als Parameter aufzugeben. Bei einer weiteren Verbesserung des Verfahrens kann man eine andere Variable $y_k \neq y_i$ als Parameter auszeichnen und erhält in unserem Beispiel die um die Mediane gespiegelte Kurve, die keinen Umkehrpunkt mehr besitzt (siehe Abb.8).
Die Schrittweitensteuerung und der Parameterwechsel wird nun mit Hilfe eines groben Schemas für ein Fortsetzungsverfahren beschrieben. Dabei zeichnet $k$ den aktuellen Parameter aus. Diejenige Koordinate $y_k$, deren Wert sich am stärksten ändert, wird der neue Parameter. Im Fall $m = 2$ bedeutet das, daß der Parameter gewechselt wird, wenn der Anstieg der Lösungskurve größer als 1 wird.

**Algorithmus 4** (Schrittweitensteuerung und Parameterwechsel):
Es sei $r$ festgesetzt, $u_0$, $u_1$ und $kold$ vorgegeben, $j := 2$.

**Schritt 1:**

$$\text{Finde } knew \in \{1, \dots, m\} \text{ mit } \left|\frac{(u_{j-1}-u_{j-2})_{knew}}{(u_{j-1}-u_{j-2})_l}\right| \geq 1 \quad \forall l \in \{1, \dots, m\}.$$

$$\text{Falls } knew \neq kold: \ \Delta\tau := (u_{j-1})_{knew} - (u_{j-2})_{knew}.$$

**Schritt 2:**

Definiere $\tau_j := (u_{j-1})_{knew} + \Delta\tau$

$F_{knew,j} := \{y_{knew} - \tau_j,\ H_1(y), \ldots, H_n(y)\}$ und

$\hat{u} := u_{j-1} + \frac{\Delta\tau}{(u_{j-1})_{knew} - (u_{j-2})_{knew}}(u_{j-1} - u_{j-2})$.

Berechne $u_j$ mit Algorithmus 3 aus folgenden Bedingungen:

$$F_{knew,j}(u_j) = 0$$

$$u_j \in V_{knew,r}(\hat{u}) = \{u \in \mathbb{R}^m \mid u_{knew} = \tau_j,\ \|u - \hat{u}\| \leq r\}$$

$$kold := knew.$$

Abbruchbedingung: wenn beispielsweise $(u_j)_1 > c$ für $c$ konstant.

**Schritt 3:**

Sei $anz$ die Anzahl der in Schritt 2 berechneten Punkte $u_j$

Falls $anz = 0$: $\Delta\tau := \frac{\Delta\tau}{2}$, gehe zu Schritt 2.

Falls $anz = 1$: $j := j + 1$, $\Delta\tau := 2\Delta\tau$, gehe zu Schritt 1.

**Bemerkungen:**
1.) Im Fall $anz \geq 2$ wird der Rang von $H'$ betrachtet, um herauszufinden, ob man auf einen Bifurkationspunkt gestoßen ist (siehe Kapitel 3.1.2).
2.) Bei diesem Algorithmus wird in Schritt 2 beim Lösen des Gleichungssystems mit Algorithmus 3 immer von neuem eine Gröbnerbasis berechnet, was zu einer langen Rechenzeit führt. Besser ist es, die benötigten Gröbnerbasen (für jeden Parameter eine) getrennt zu berechnen und am Anfang des Programmes zu laden.
3.) Die Schrittweitensteuerung kann noch leicht verbessert werden, indem man im Fall $anz = 1$ die neue Schrittweite von der Qualität der Vorhersage für den neuen Punkt $u_j$ abhängig macht: Wenn $u_j$ „nicht weit“ vom Schätzwert $\hat{u}$ entfernt ist, vergrößert man die Schrittweite $\Delta\tau$. Wenn $u_j$ im äußeren Bereich des Suchbereichs $V_{knew,r}(\hat{u})$ ist, wird $\Delta\tau$ verkleinert. Sonst wird $\Delta\tau$ gleich gelassen.

# 3. Implementierung und Anwendung der Verfahren

## 3.1. Implementierung

In diesem Kapitel wird kurz beschrieben, wie die beiden Verfahren programmiert wurden. Für eine genauere Untersuchung wurde eine Diskette beigelegt, auf der beide Programme gespeichert sind.

### 3.1.1. Implementierung des numerischen Verfahrens

Das numerische Verfahren wurde in matlab programmiert. Das hatte den Vorteil, daß das Programm relativ einfach zu schreiben war, insbesondere deswegen, weil einige Werkzeuge, wie zum Beispiel die QR-Zerlegung oder Grafikbefehle, schon bereitstehen.

Das Programm ist in viele Unterprogramme aufgeteilt. Die wichtigsten sind hauptpro.m, fort.m und hauptsch.m.
Mit dem Aufruf von hauptpro.m wird das Programm gestartet. Es enthält die Eingabe von Startwerten, den Aufruf der Hauptschleife hauptsch.m und den Grafikteil.
Fort.m ist im wesentlichen das Fortsetzungsverfahren. Es steuert die Schrittweite und ruft das Newtonverfahren newt.m auf.

Das Programm hauptsch.m legt den Programmablauf fest. Zuerst wird das Fortsetzungsverfahren angewendet. Wenn der Rang von $H'(u)$ kleiner oder gleich $n-2$ wird, wird der betreffende Ast nicht mehr weiter berechnet.
Wenn der Rang $n-1$ ist, werden der Bifurkationspunkt und 2 linear unabhängige Tangenten $Du1$ und $Du2$ im Bifurkationspunkt berechnet. Dann sondert es aus den möglichen 4 Richtungen $\pm Du1$ und $\pm Du2$ die Richtung, aus der man gekommen ist, aus und geht den 3 neuen Ästen getrennt entlang. Solange der Rang gleich $n-1$ bleibt, können die Tangenten nicht wie in Kapitel 1.1.2.2 berechnet werden. Deshalb werden die Tangenten im Bifurkationspunkt solange im Fortsetzungsverfahren verwendet, bis

der Rang wieder gleich $n$ wird. Dann wird die Hauptschleife aufgerufen, die wieder mit dem Fortsetzungsverfahren beginnt.

Um nicht versehentlich mehrmals denselben Weg zu berechnen oder gar im Kreis zu gehen, werden schon behandelte Bifurkationspunkte gespeichert. Bei einem neuerlichen Auftreten von Bifurkationspunkten werden deren Äste nicht mehr berechnet.

Die Probleme des Verfahrens bestehen zum einen darin, daß der Startpunkt, den der Benutzer einzugeben hat, nahe genug an der Kurve liegen muß, damit das Newtonverfahren konvergiert.
Das Hauptproblem ist aber, daß Bifurkationspunkte übersehen werden können. Dabei spielen vor allem die maximal zugelassene Schrittweite $max\lambda$ (siehe 1.1.2.1) und die Konstante *tolrang* (siehe A.2) zur Beurteilung des Ranges der Jacobimatrix $H'(u)$ eine Rolle.
Wenn man wegen einer zu großen Schrittweite nicht nahe genug an den Bifurkationspunkt herankommt, tritt kein Rangverlust auf, und deshalb wird der Bifurkationspunkt nicht berechnet. Man kann aber die Schrittweite nicht zu sehr beschränken, da das Programm sonst zu lange dauert.
Die Konstante *tolrang* darf auch nicht zu groß gewählt werden, da man sonst den Bifurkationspunkt zu früh, d.h. nicht im Konvergenzbereich des Newtonverfahrens zur Berechnung des Bifurkationspunktes, bemerkt. Wenn *tolrang* zu klein gewählt wird, können wiederum Bifurkationspunkte übersehen werden.

### 3.1.2. Implementierung des algebraischen Verfahrens

Das algebraische Verfahren wurde in maple programmiert, weil maple in der Bibliothek grobner Befehle zur Berechnung von Gröbnerbasen enthält. Allerdings benötigte es aufgrund der Vielfalt an Datentypen und der komplizierteren Syntax eine längere Einarbeitungszeit als matlab.

Da das Verfahren auf denselben Ideen wie das numerische Verfahren beruht, ist der Aufbau des algebraischen Programmes im wesentlichen gleich wie der Aufbau des numerischen Programmes. Die Unterprogramme sind in 3 Bibliotheken eingebettet.

Die Bibliothek diplpro enthält die Unterprogramme hauptpro und hauptsch, die ihren numerischen Gegenstücken sehr ähneln. In der Bibliothek hilfspro stehen die restlichen Unterprogramme, insbesondere die Unterprogramme fort und newt. Die Gleichungssysteme mit voreingestellten Startwerten sind in der Bibliothek glsys.

Der größte Unterschied zum numerischen Programm liegt in der Steuerung. Um sich die Berechnung der Jacobimatrix $H'(u)$ in jedem Punkt zu ersparen, wird das algebraische Programm über die Anzahl der im Suchbereich $V_{i,r}$ (siehe Kapitel 2.3.1) gefundenen Nullstellen gesteuert.
Wenn genau eine Nullstelle gefunden wurde, wird die berechnete Nullstelle der nächste Punkt und der Parameter und die Schrittweite $\Delta\tau$ neu festgelegt (siehe 2.3.2).
Wenn keine Nullstelle gefunden wurde, wird die Schrittweite $\Delta\tau$ verkleinert und noch einmal nach einer Lösung gesucht.
Wenn mehrere Nullstellen gefunden wurden, wird der Rang der Jacobimatrix $H'(u)$ untersucht:
Bei $\mathrm{Rg}(H'(u)) = n$ wird die Nullstelle, die dem vorher berechneten Punkt am nächsten ist, der nächste Punkt. Dann wird der Parameter neu festgesetzt. Diese Situation kann bei Umkehrpunkten auftreten.
Bei $\mathrm{Rg}(H'(u)) \leq n-2$ wird dem betreffenden Ast nicht mehr weiter entlanggerechnet.
Bei $\mathrm{Rg}(H'(u)) \leq n-1$ wird der Bifurkationspunkt so wie im ersten Programm mit Hilfe des erweiterten Systems von Moore berechnet.

Die Probleme des Verfahrens liegen hauptsächlich in der beschränkten Einsetzbarkeit. Aufgrund der Verwendung von Gröbnerbasen können nur polynomiale Gleichungssysteme behandelt werden. Da die Zeit zur Berechnung von Gröbnerbasen mit der Anzahl der Gleichungen und der Anzahl der Unbekannten sehr stark zunimmt, muß man sich außerdem auf „kleine“ Gleichungssysteme beschränken.
Aufgrund desselben Verfahrens zur Berechnung können Bifurkationspunkte auch bei diesem Programm übersehen werden. Die Rolle der maximalen Schrittweite $max\lambda$ übernehmen jetzt die Konstante $r$, die die Größe des Suchbereiches $V_{i,r}$ festlegt und die maximal zugelassene Schrittweite des aktuellen Parameters $Dparammax$. Um überhaupt zur Betrachtung des Ranges von $H'(u)$ zu kommen, müssen zuerst mehrere Nullstellen gefunden werden, was wiederum vom $Dparammax$ und $max\lambda$ abhängt.

## 3.2. Anwendung der Verfahren

In diesem Kapitel werden einige Beispiele mit den beiden Verfahren berechnet.
Die Lösungskurven der zwei Beispiele zum Homotopieverfahren enthalten keine Bifurkationspunkte. Die Gleichungen sind keine Polynomgleichungen und können daher nur mit dem numerischen Verfahren behandelt werden.
Der Brusselator ist ein Differentialgleichungssystem. Wenn man sich für die konstanten Lösungen des Brusselators interessiert, erhält man ein Gleichungssystem. Dieses kann

in ein System von Polynomgleichungen umgeschrieben werden, auf das beide Verfahren angewendet werden können. Beim Brusselator mit 4 Boxen erhält man 8 Gleichungen in 9 Variablen. Dieses Gleichungssystem ist für die Gröbnerbasenberechnung mit maple zu groß. Die Gröbnerbasen für den Brusselator mit 2 Boxen konnten hingegen berechnet werden.

### 3.2.1. Homotopieverfahren

Parameterabhängige Systeme können dazu genutzt werden, ein Verfahren zur Lösung nichtlinearer Gleichungssysteme zu bekommen. Die Idee dabei ist, sich von einem bereits gelösten Problem

$$G(x) = 0$$

Zug um Zug zur Lösung des eigentlichen Problems

$$F(x) = 0$$

durchzuarbeiten. Dazu konstruiert man sich ein parameterabhängiges Problem

$$H(x, \tau) = 0, \quad \tau \in [0, 1],$$

welches die beiden Probleme miteinander verbindet, d.h.

$$H(x, 0) = G(x) \text{ und } H(x, 1) = F(x)$$

für alle $x$. Eine solche Abbildung heißt Einbettung des Problems $F(x) = 0$ oder auch Homotopie. Das einfachste Beispiel ist die sogenannte lineare Einbettung

$$H(x, \tau) := \tau F(x) + (1 - \tau)G(x),$$

der allerdings problemangepaßte Einbettungen vorzuziehen sind. Wenden wir eine Fortsetzungsmethode auf dieses parameterabhängige Problem $H(x, \tau) = 0$ an, wobei wir mit einer uns bekannten Lösung $x_0$ von $G(x) = 0$ beginnen, so spricht man auch von einer Homotopiemethode zur Lösung von $F(x) = 0$.

In [8] ist das folgende Problem gestellt:

$$F(x) := x - \Phi(x) = 0, \text{ wobei} \tag{3.1}$$

$$\Phi_i(x) := e^{\cos(i \cdot \sum_{j=1}^{10} x_j)}, \quad i = 1, \ldots, 10.$$

#### 3.2.1.1. Die problemangepaßte Einbettung

Die problemangepaßte Einbettung von (3.1) lautet:

$$H_i(x,\tau) = x_i - e^{\tau\cdot\cos(i\cdot\sum_{j=1}^{10} x_j)}, \quad i = 1,\ldots,10.$$

Die Fortsetzung von $\tau = 0$ mit Startwert $x_i^0 = 1$ für $i = 1,\ldots,10$ führt bei $\tau = 1$ zur Lösung $x = (0,52 \quad 0,50 \quad 0,38 \quad 0,89 \quad 2,43 \quad 2,31 \quad 0,81 \quad 0,37 \quad 0,54 \quad 1,66)$.

Das Verfahren konnte der Lösungskurve noch mit $max\lambda = 1.2$ folgen und benötigte 16 Sekunden und 24 Bildpunkte bis zu $\tau = 1$. Eine maximale Schrittweite von 0.3 führte zu 46 Sekunden und 74 Bildpunkten (siehe Abb.9).

Es sei angemerkt, daß in diesem Beispiel keine Bifurkationspunkte auftreten, die Überschneidungen der Lösungskurven ergeben sich nur in der Projektion auf die jeweilige Koordinatenebene.

#### 3.2.1.2. Die triviale Einbettung

Die triviale Einbettung von (3.1) lautet:

$$H(x,\tau) = \tau F(x) + (1-\tau)x = x - \tau\Phi(x).$$

Die Fortsetzung von $\tau = 0$ mit Startwert $x_i^0 = 0$ für $i = 1,\ldots,10$ führt bei $\tau = 1$ zur Lösung.

Das Verfahren konnte der Lösungskurve noch mit $max\lambda = 0.3$ folgen und benötigte 338 Sekunden und 396 Bildpunkte bis zum selben Endpunkt bei $\tau = 1$, also bei gleicher maximaler Schrittweite etwa 7-mal so lange wie für die problemangepaßte Einbettung (siehe Abbildungen 10 bis 19).

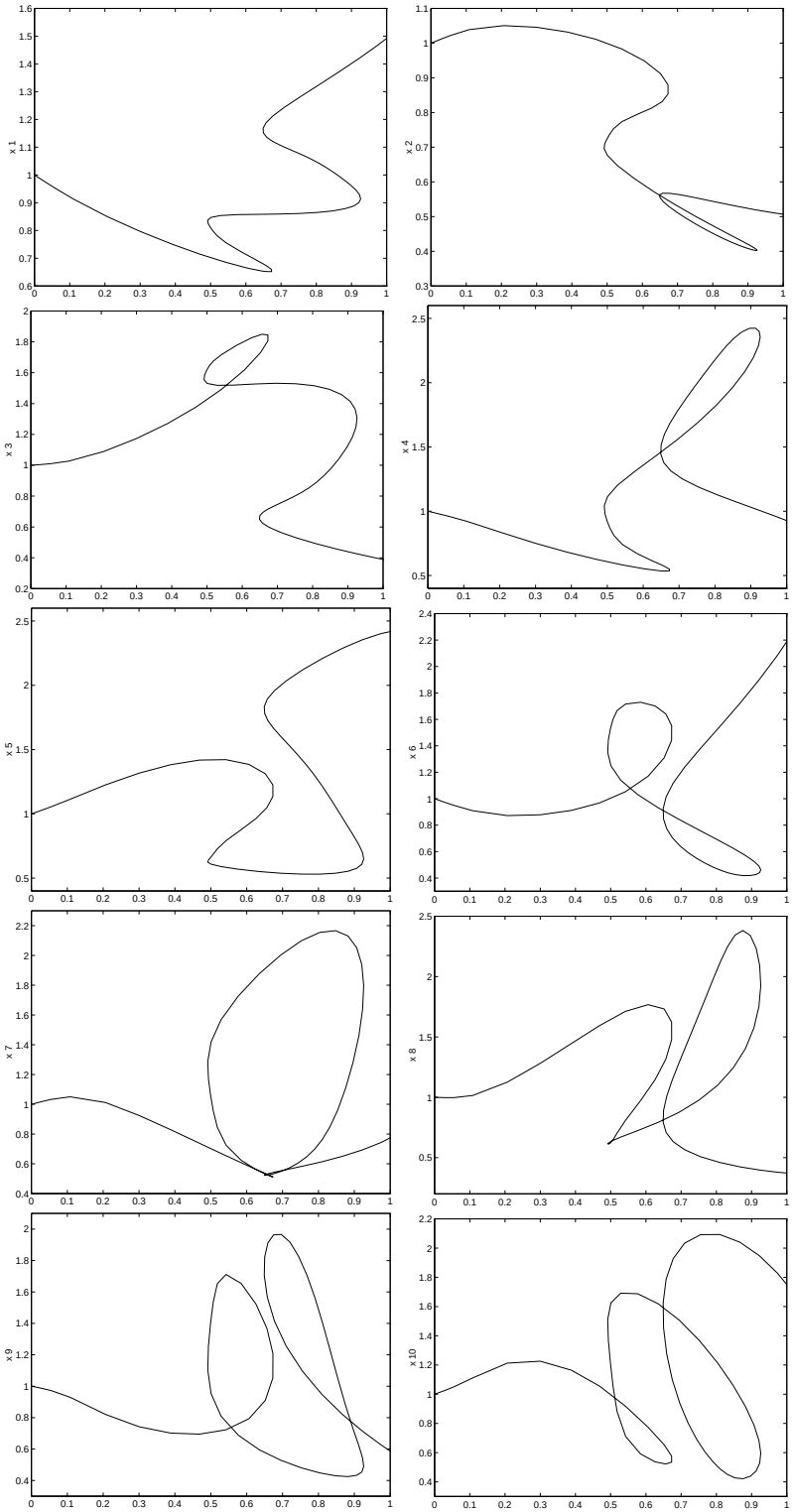


Abb.9 Die Koordinatenfunktionen $x_1(\tau), \ldots, x_{10}(\tau)$ der Lösungkurven der problemangepaßten Einbettung.

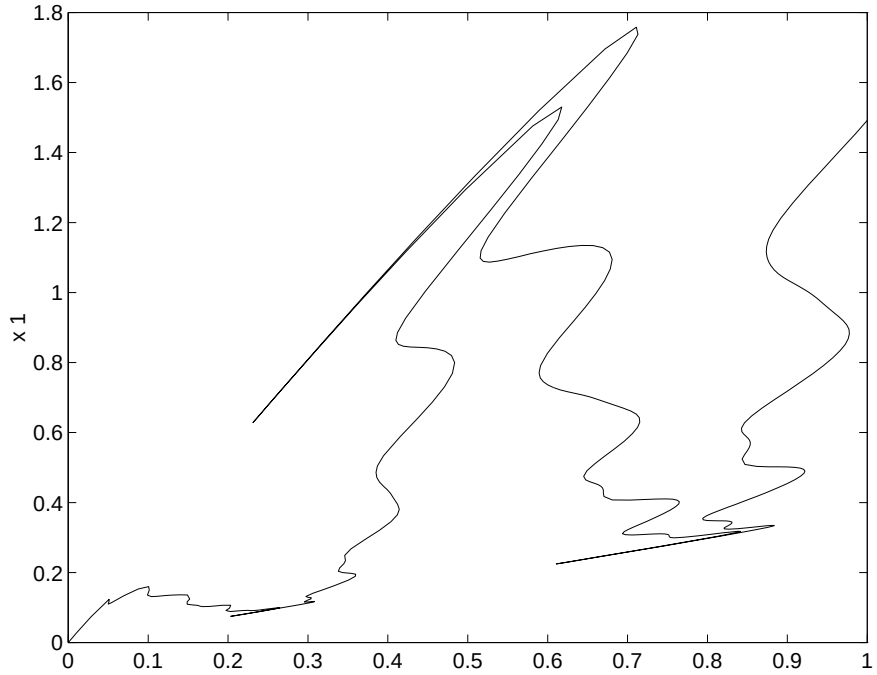


Abb.10 Die Koordinatenfunktion $x_1(\tau)$ der Lösungkurve der trivialen Einbettung.

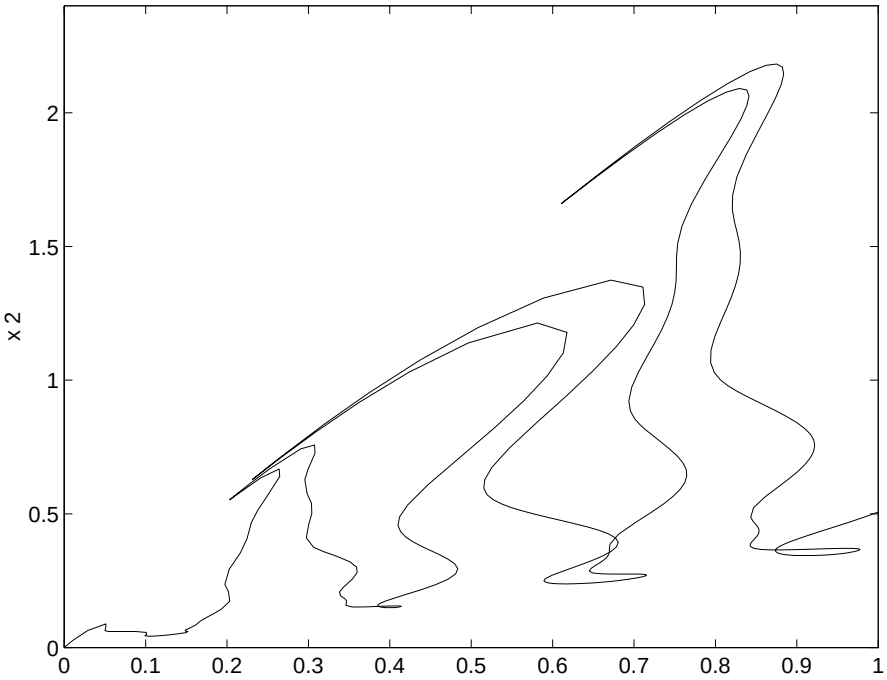


Abb.11 Die Koordinatenfunktion $x_2(\tau)$ der Lösungkurve der trivialen Einbettung.

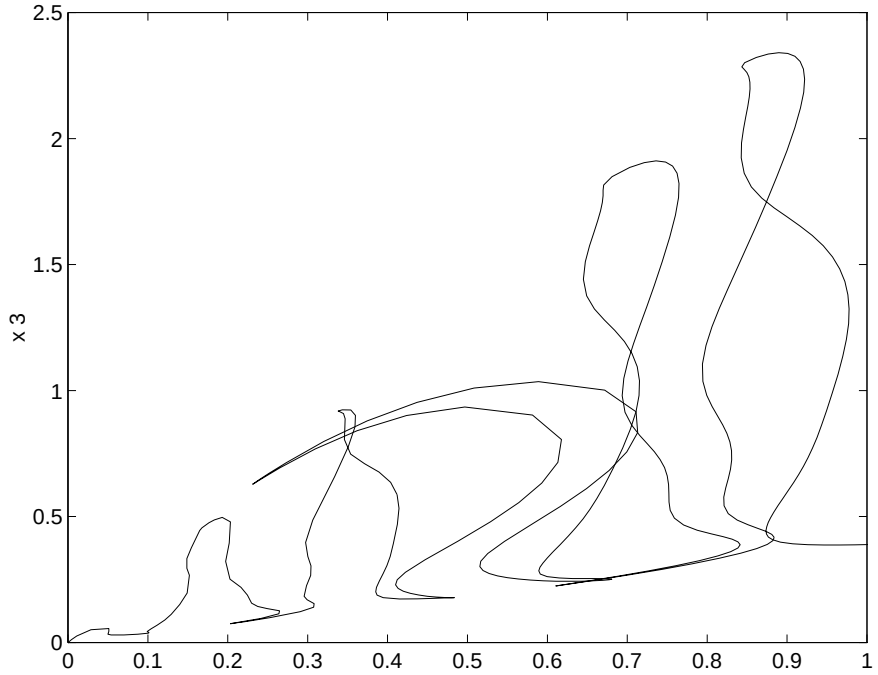


Abb.12 Die Koordinatenfunktion $x_3(\tau)$ der Lösungkurve der trivialen Einbettung.

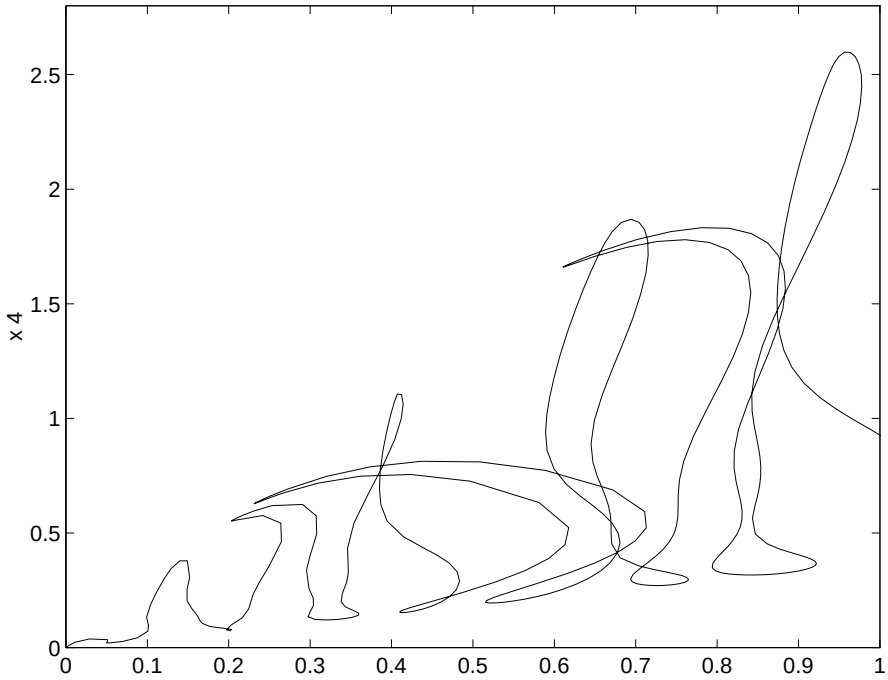


Abb.13 Die Koordinatenfunktion $x_4(\tau)$ der Lösungkurve der trivialen Einbettung.

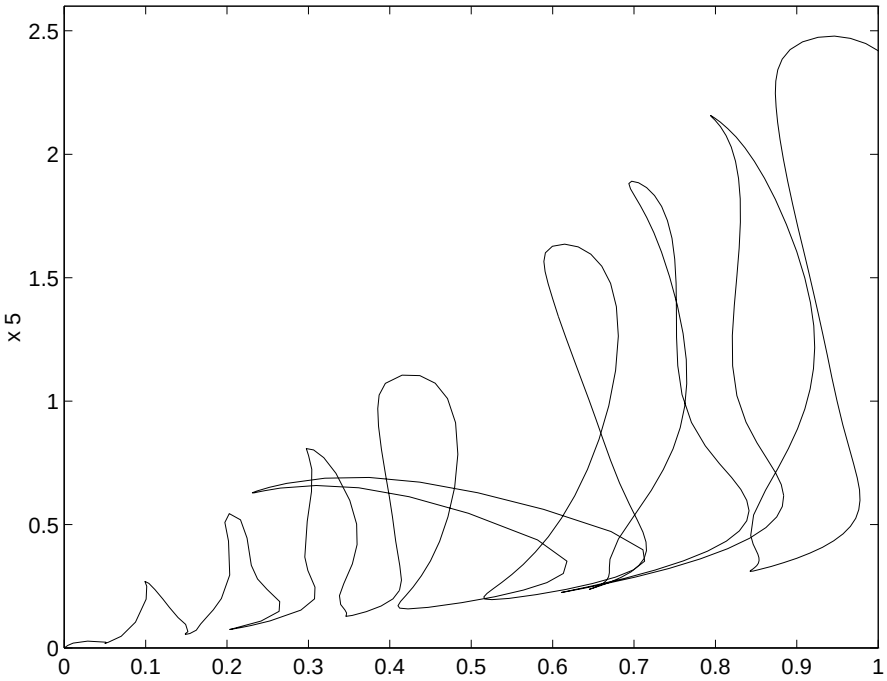


Abb.14 Die Koordinatenfunktion $x_5(\tau)$ der Lösungkurve der trivialen Einbettung.

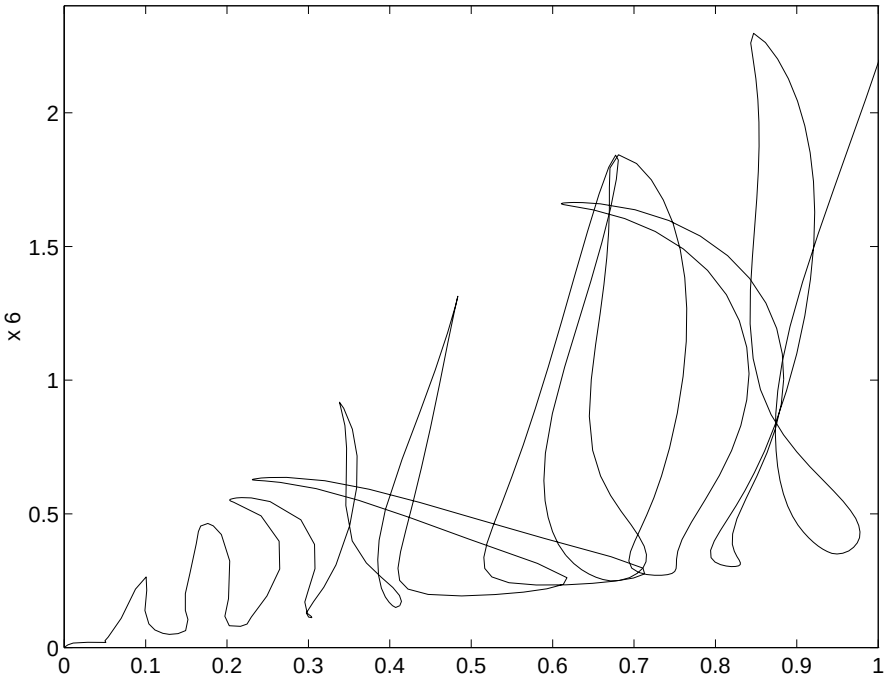


Abb.15 Die Koordinatenfunktion $x_6(\tau)$ der Lösungkurve der trivialen Einbettung.

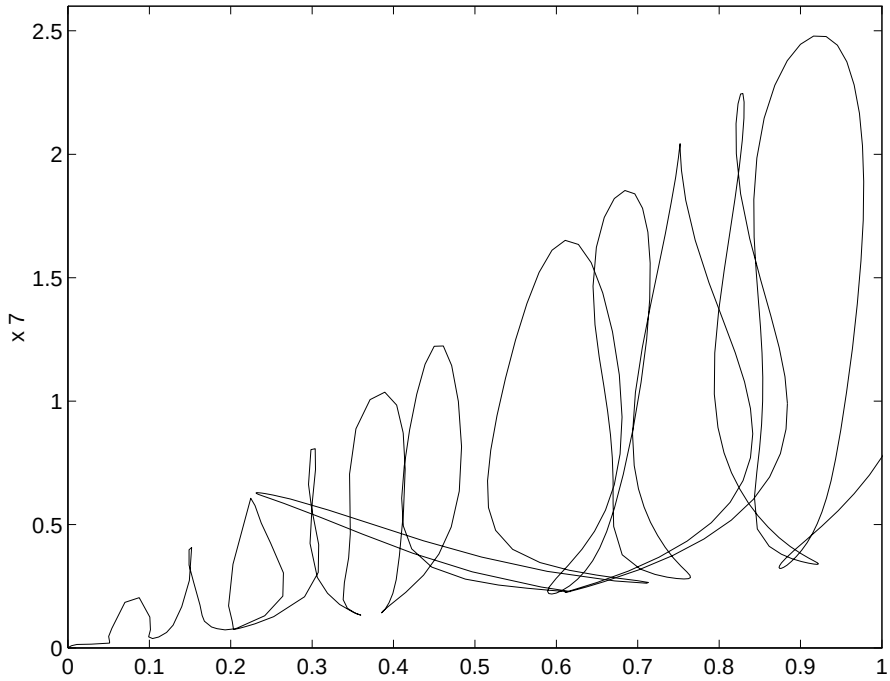


Abb.16 Die Koordinatenfunktion $x_7(\tau)$ der Lösungkurve der trivialen Einbettung.

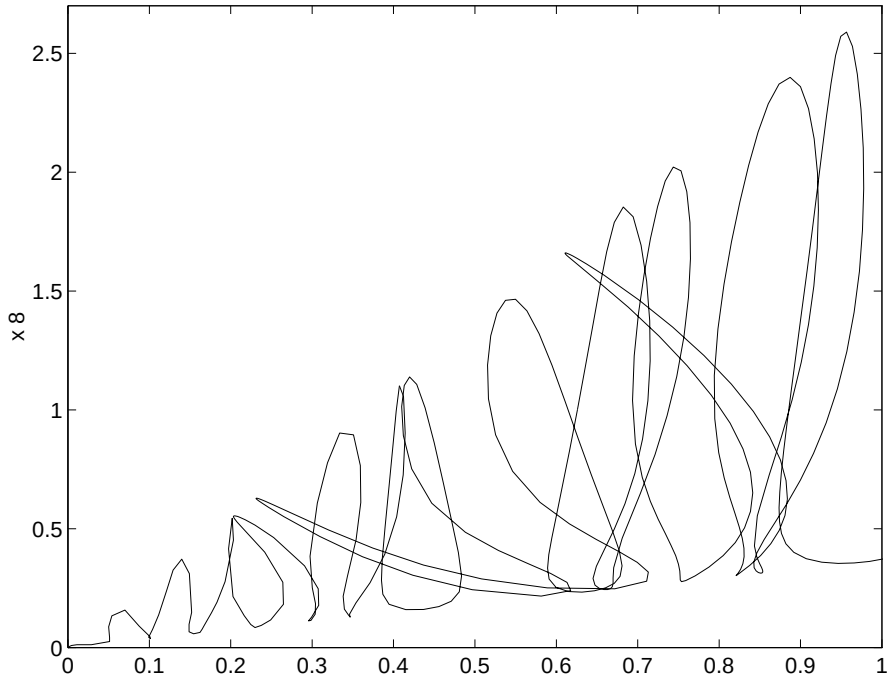


Abb.17 Die Koordinatenfunktion $x_8(\tau)$ der Lösungkurve der trivialen Einbettung.

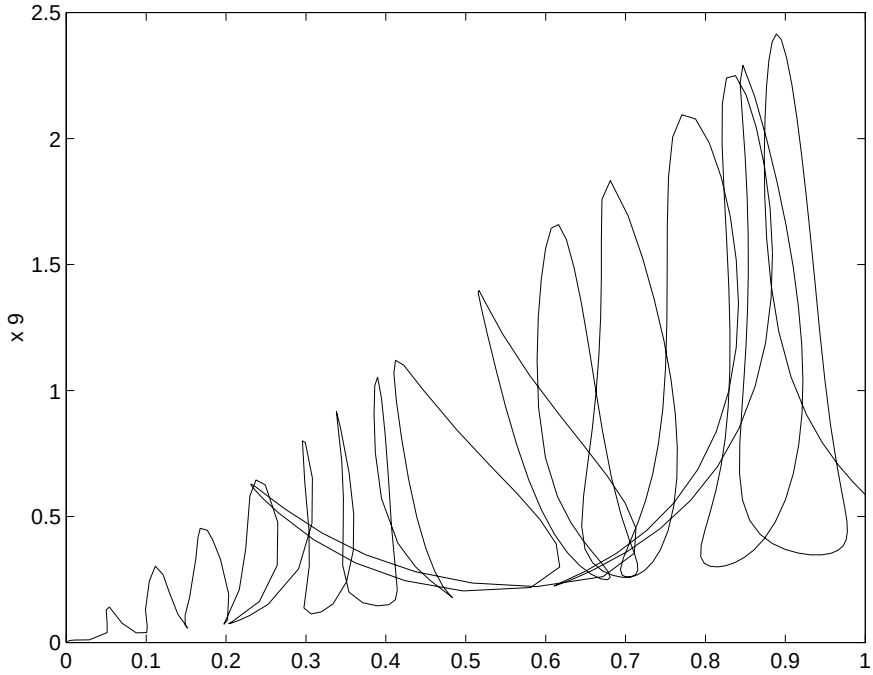


Abb.18 Die Koordinatenfunktion $x_9(\tau)$ der Lösungkurve der trivialen Einbettung.

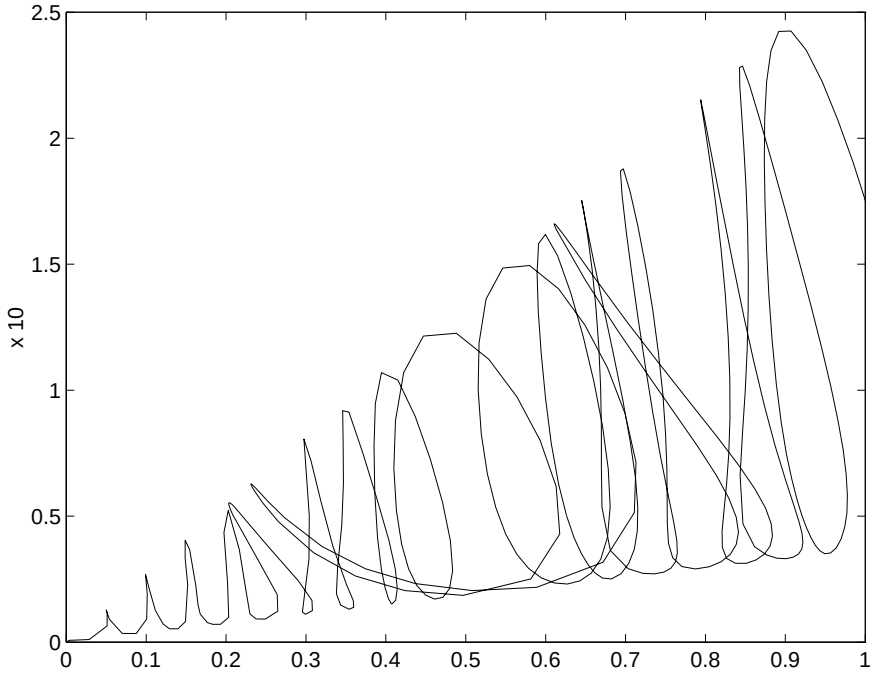


Abb.19 Die Koordinatenfunktion $x_{10}(\tau)$ der Lösungkurve der trivialen Einbettung.

### 3.2.2. Der Brusselator

Ein Beispiel aus der Chemie ist der Brusselator (siehe [7]). Es handelt sich dabei um ein System von mehreren Boxen, in denen 4 chemische Substanzen $A, B, W, Y$ in verschiedenen Konzentrationen enthalten sind. Wir betrachten den Fall, in dem die Konzentrationen der Stoffe $W$ und $Y$ in den einzelnen Boxen verschieden sein können. Die Substanzen reagieren nach folgendem Schema:

$$\begin{aligned} A &\longrightarrow W \\ 2W + Y &\longrightarrow 3W \\ B + W &\longrightarrow Y + D \\ W &\longrightarrow E \end{aligned}$$

D und E sind Zerfallsstoffe, die im weiteren Verlauf keine Rolle mehr spielen.

Zusätzlich zu den chemischen Reaktionen diffundieren die Stoffe zwischen den einzelnen Boxen. Es ergeben sich für den Brusselator mit 2 Boxen die folgenden kinetischen Gleichungen:

$$\begin{aligned} \frac{dw_1}{dt} &= a + w_1^2 y_1 - b w_1 - w_1 + \frac{1}{\tau^2}(w_2 - w_1) \\ \frac{dy_1}{dt} &= b w_1 - w_1^2 y_1 + \frac{1}{\tau^2}(y_2 - y_1) \\ \frac{dw_2}{dt} &= a + w_2^2 y_2 - b w_2 - w_2 + \frac{1}{\tau^2}(w_1 - w_2) \\ \frac{dy_2}{dt} &= b w_2 - w_2^2 y_2 + \frac{1}{\tau^2}(y_1 - y_2) \end{aligned}$$

Dabei sind $a$ und $b$ die Konzentrationen der Stoffe $A$ und $B$, $w_1$ und $y_1$ bzw. $w_2$ und $y_2$ die Konzentrationen der Substanzen $W$ und $Y$ in der ersten bzw. zweiten Box. Der Diffusionsparameter $\frac{1}{\tau^2}$ ist für die Diffusion der Substanzen $W$ und $Y$ gleich groß.

Interessiert man sich für die stationären Lösungen, wird die rechte Seite gleich Null, und man erhält ein Gleichungssystem. In unserem Beispiel wurde $a = 6$ und $b = 2$ gewählt.

**Bemerkungen**: 1.) Zur leichteren Bezeichnung führen wir folgende Bezeichnung ein: $(x_1, \ldots, x_{2N}) := (w_1, y_1, \ldots, w_N, y_N)$, $N \ldots$ Anzahl der Boxen des Brusselators.
2.) Es ist leicht nachzurechnen, daß die Gerade
$L := \{(x_1, \ldots, x_{2N}, \tau) \mid x_1 = x_3 = \ldots = x_{2N-1} = 2, \ , x_2 = x_4 = \ldots = x_{2N} = 3, \ \tau \in \mathbb{R}\}$
eine Lösungskurve des Brusselators ist.

#### 3.2.2.1. Der Brusselator mit 2 Boxen

Das untersuchte Gleichungssystem des Brusselators mit 2 Boxen hatte folgendes Aussehen:

$$\begin{aligned} 2 + x_1^2 x_2 - 7x_1 + \frac{1}{\tau^2}(x_3 - x_1) &= 0 \\ 6x_1 - x_1^2 x_2 + \frac{10}{\tau^2}(x_4 - x_2) &= 0 \\ 2 + x_3^2 x_4 - 7x_3 + \frac{1}{\tau^2}(x_1 - x_3) &= 0 \\ 6x_3 - x_3^2 x_4 + \frac{10}{\tau^2}(x_2 - x_4) &= 0 \end{aligned}$$

Das Bifurkationsdiagramm konnte vollständig berechnet werden (siehe Abb.20).
Es gibt 2 Bifurkationspunkte:

$$(x^1, \tau^1) = (2 \quad 3 \quad 2 \quad 3 \quad 0,6658) \quad \text{und} \quad (x^2, \tau^2) = (2 \quad 3 \quad 2 \quad 3 \quad 4{,}794)$$

Die Bifurkationspunkte sind am leichtesten von der Gerade $L$ aus zu berechnen:
$(x^1, \tau^1)$ etwa mit den Einstellungen $(x^0, \tau^0)$ =(2 3 2 3 0,6668), $tolrang = 0.03$, einer Anfangsschrittweite $\lambda^0 = 10^{-6}$ und $max\lambda = 0,01$.
$(x^2, \tau^2)$ mit den Einstellungen $(x^0, \tau^0) = (2 \quad 3 \quad 2 \quad 3 \quad 5)$, $tolrang = 0,03$, $\lambda^0 = 0,01$ und $max\lambda = 0,01$.
Die Berechnung der Tangenten und das Fortsetzungsverfahren für die einzelnen Äste bereitete dann keine Probleme, allerdings braucht man mit $max\lambda = 0,01$ zur Berechnung der einzelnen Äste viel länger als nötig.
Die Berechnung der geschlossenen Kurve gelingt mit einem passenden Startpunkt (z.B. $(x^0, \tau^0)$ =(2,91 2,18 1,09 4,69 5,05) bei $max\lambda = 0,5$ in 37 Sekunden. Man beachte, daß $max\lambda$ um einiges größer als bei der Berechnung der Bifurkationspunkte ist. Die beiden Bifurkationspunkte werden bei dieser Einstellung (mit $tolrang = 0.03$) übersehen.

**Bemerkung**: Der Graph der Koordinatenfunktionen $x_1(\tau)$ und $x_3(\tau)$ bzw. $x_2(\tau)$ und $x_4(\tau)$ ist zwar gleich, aber es gilt: $x_1(\tau) \neq x_3(\tau)$ und $x_2(\tau) \neq x_4(\tau)$
Genauer: $x_1(\tau) \leq 0 \iff x_3(\tau) \geq 0$ bzw. $x_2(\tau) \leq 0 \iff x_4(\tau) \geq 0$.

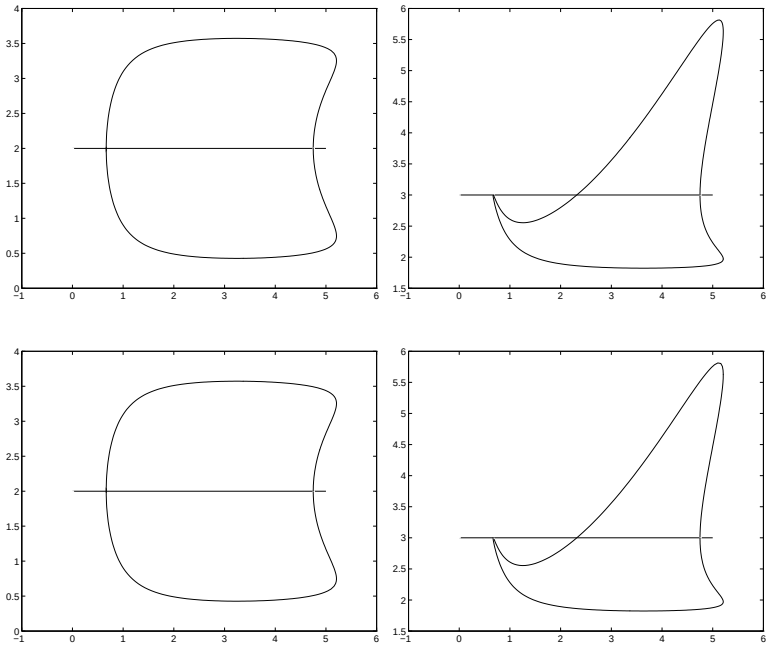

Abb.20 Die Koordinatenfunktionen $x_1(\tau), \ldots, x_4(\tau)$ der Lösungkurven des Brusselators mit 2 Boxen.

Wenn man im Gleichungssystem des Brusselators den Diffusionsparameter $\frac{1}{\tau^2}$ durch $\tau^2$ ersetzt, erhält man folgendes polynomiale Gleichungssystem:

$$\begin{aligned} 2 + x_1^2 x_2 - 7x_1 + \tau^2(x_3 - x_1) &= 0 \\ 6x_1 - x_1^2 x_2 + 10\tau^2(x_4 - x_2) &= 0 \\ 2 + x_3^2 x_4 - 7x_3 + \tau^2(x_1 - x_3) &= 0 \\ 6x_3 - x_3^2 x_4 + 10\tau^2(x_2 - x_4) &= 0 \end{aligned}$$

Bei diesem Gleichungssystem kann man nun das numerische mit dem algebraischen Verfahren vergleichen:

Zur Beurteilung der Geschwindigkeit wurde untersucht, wie lange die zwei Programme für die Berechnung der geschlossenen Kurve (siehe Abb.21) brauchen:
Bei $max\lambda = 0.17$ benötigte das numerische Programm 227 Punkte und 73 Sekunden. Das algebraische Programm benötigte (bei $r = 0.01$ und $Dparammax = 0.2$) 211 Bildpunkte und 80 Sekunden, wobei allerdings noch die Zeit zur Berechnung der Gröbnerbasen dazuzurechnen ist. Beide Programme brauchen also zur Berechnung eines Bildpunktes ungefähr gleich lange.
Das numerische Programm konnte allerdings $max\lambda = 0.5$ die Kurve mit 84 Punkten in 30 Sekunden berechnen, während für das algebraische Programm keine Einstellung zu finden war, die die Kurve schneller berechnen konnte.

Wie schon in Kapitel 3.1.2 angedeutet, kam es beim algebraischen Verfahren bei der Erkennung des Bifurkationspunktes $(2 \quad 3 \quad 2 \quad 3 \quad \frac{1}{4{,}794})$ zu Problemen. Damit das Programm den Rang von $H'(u)$ berechnet, müssen zuerst mehrere Nullstellen gefunden werden. Dieser Fall trat jedoch nicht auf, sodaß dieser Bifurkationspunkt unerkannt blieb.

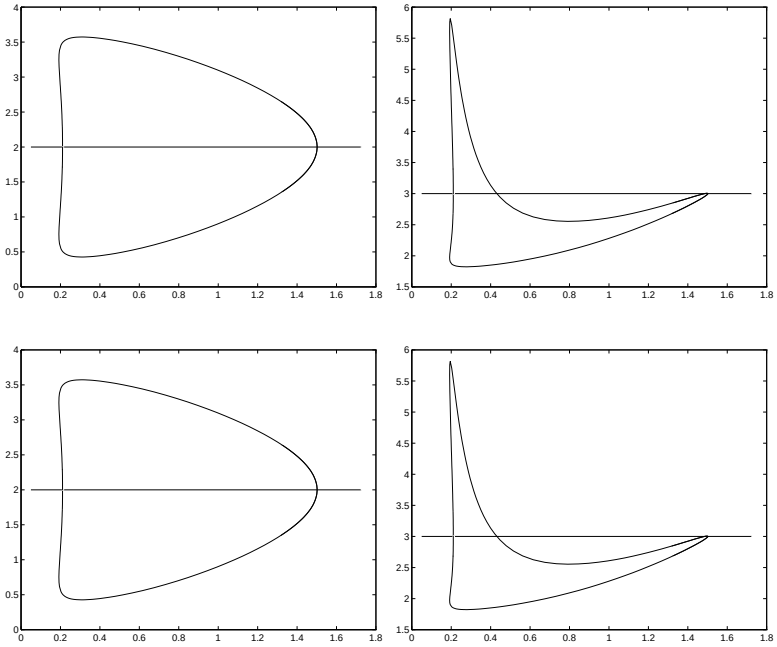

Abb.21 Die Koordinatenfunktionen $x_1(\tau), \ldots, x_4(\tau)$ der Lösungkurven des polynomialen Brusselators mit 2 Boxen.

#### 3.2.2.2. Der Brusselator mit 4 Boxen

Die 4 Boxen sind in einer Kette angeordnet, deshalb können die Substanzen nur zwischen benachbarten Boxen diffundieren. Deshalb unterscheidet sich der Diffusionsterm der beiden inneren Boxen Nummer 2 und 3, die ja zwei benachbarte Boxen besitzen,

von dem der beiden äußeren, die nur an je eine Box angrenzen.

$$\begin{aligned}
2 + x_1^2 x_2 - 7x_1 + \frac{1}{\tau^2}(x_3 - x_1) &= 0 \\
6x_1 - x_1^2 x_2 + \frac{10}{\tau^2}(x_4 - x_2) &= 0 \\
2 + x_3^2 x_4 - 7x_3 + \frac{1}{\tau^2}(x_1 + x_5 - 2x_3) &= 0 \\
6x_3 - x_3^2 x_4 + \frac{10}{\tau^2}(x_2 + x_6 - 2x_4) &= 0 \\
2 + x_5^2 x_6 - 7x_5 + \frac{1}{\tau^2}(x_7 + x_3 - 2x_5) &= 0 \\
6x_5 - x_5^2 x_6 + \frac{10}{\tau^2}(x_8 + x_4 - 2x_6) &= 0 \\
2 + x_7^2 x_8 - 7x_7 + \frac{1}{\tau^2}(x_5 - x_7) &= 0 \\
6x_7 - x_7^2 x_8 + \frac{10}{\tau^2}(x_6 - x_8) &= 0
\end{aligned}$$

Auch hier ist es möglich, durch Ersetzen von $\frac{1}{\tau^2}$ durch $\tau^2$ ein System von Polynomgleichungen zu erhalten. Die im algebraischen Verfahren benötigten Gröbnerbasen konnten mit maple jedoch nicht in einer annehmbaren Zeit berechnet werden. Somit wurde dieses Beispiel nur in obiger Gestalt mit dem numerischen Verfahren behandelt.

Das Bifurkationsdiagramm konnte vollständig berechnet werden (siehe Abbildungen 22 bis 25). Es gibt 8 Bifurkationspunkte: $(x, \tau) =$

(2 3 2 3 2 3 2 3 0,3603), (2 3 2 3 2 3 2 3 0,6658),
(2 3 2 3 2 3 2 3 0,8699), (2 3 2 3 2 3 2 3 2,5703),
(2 3 2 3 2 3 2 3 4,7494), (2 3 2 3 2 3 2 3 6,2054),
(3,45 1,95 0,55 2,63 3,45 1,95 0,55 2,63 1,65),
(3,35 1,92 0,65 5,80 0,65 5,80 3,35 1,92 5,17).

Um möglichst keine Bifurkationspunkte zu übersehen, wurde das Programm bei $(x^0, \lambda^0)$ =(2 3 2 3 2 3 2 3 0.1) mit $tolrang = 0.03$ und einer maximalen Schrittweite von nur $max\lambda = 0,01$ gestartet. Das Programm rechnete zwar einige Stunden, aber dafür wurden alle Bifurkationspunkte erkannt und das ganze Bifurkationsdiagramm berechnet. Allerdings wurden bei dieser Einstellung die zwei Bifurkationspunkte, die nicht auf $L$ liegen, zu früh erkannt, sodaß das Newtonverfahren zur ihrer Berechnung nicht konvergierte.

**Bemerkung:** Es sind nur die Graphen der Koordinatenfunktionen $x_1, x_2, x_3$ und $x_4$ abgebildet, weil sie aufgrund der Symmetrie der Angabe mit den Graphen der Koordinatenfunktionen $x_7, x_8, x_5$ und $x_6$ identisch sind.

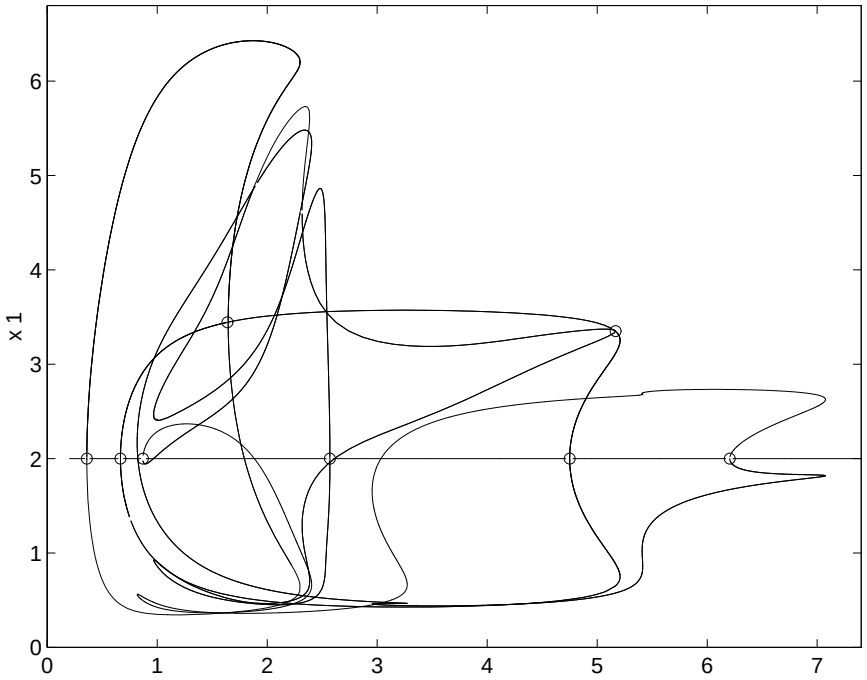


Abb.22 Der Graph der Koordinatenfunktion $x_1(\tau)$ der Lösungkurve des Brusselators mit 4 Boxen.

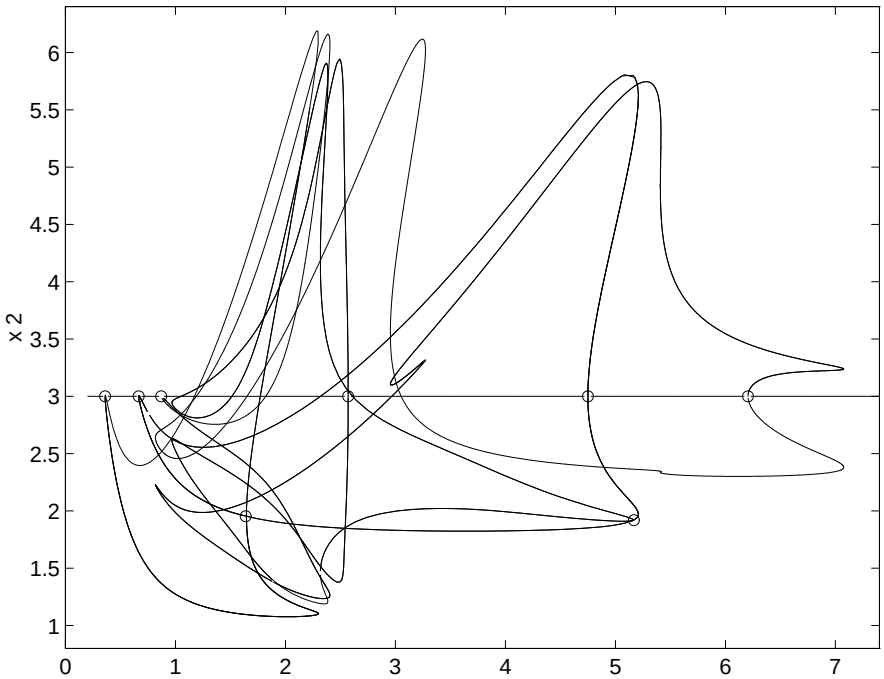


Abb.23 Der Graph der Koordinatenfunktion $x_2(\tau)$ der Lösungkurve des Brusselators mit 4 Boxen.

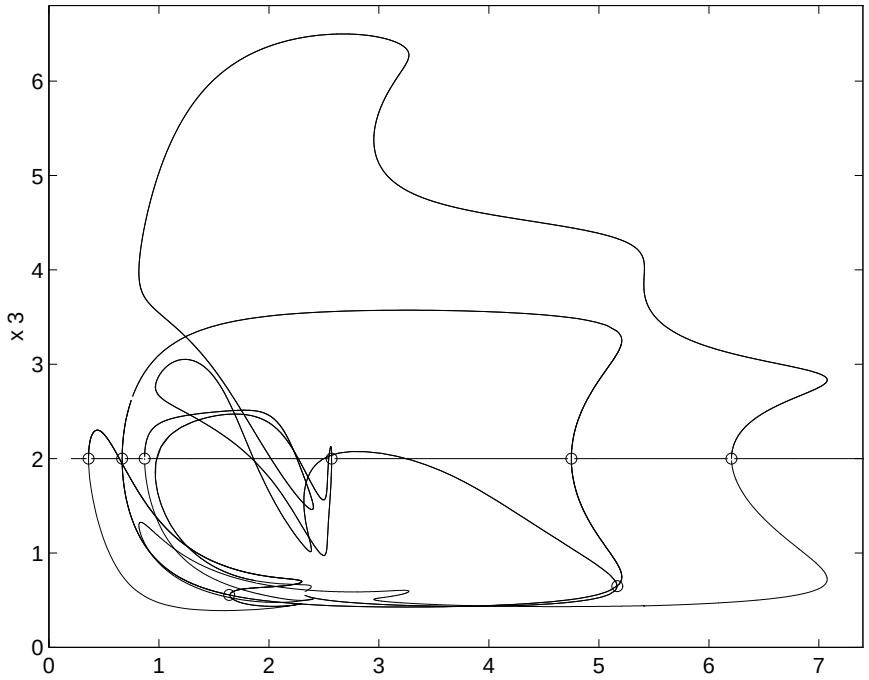


Abb.24 Der Graph der Koordinatenfunktion $x_3(\tau)$ der Lösungkurve des Brusselators mit 4 Boxen.

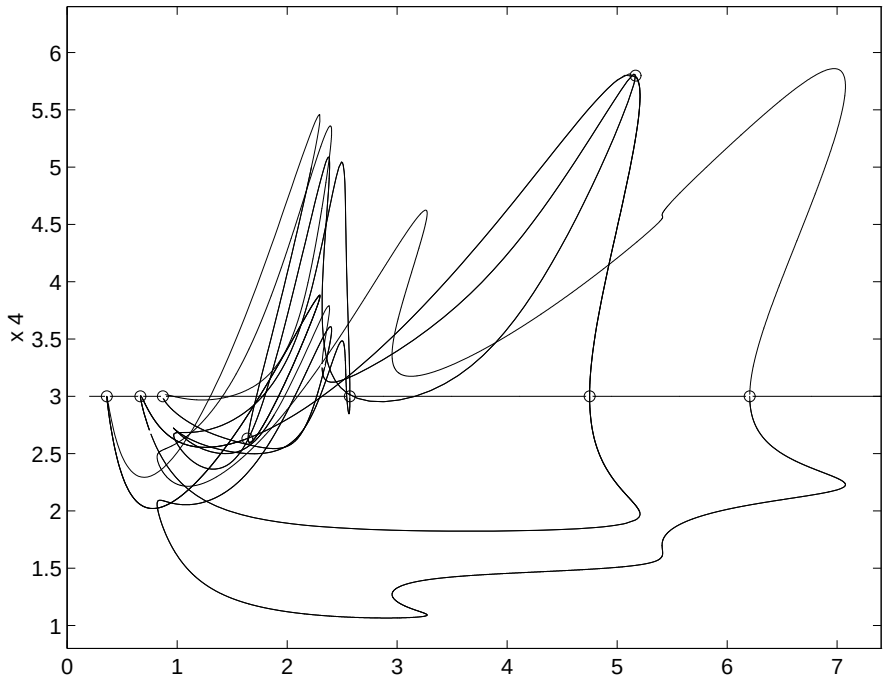


Abb.25 Der Graph der Koordinatenfunktion $x_4(\tau)$ der Lösungkurve des Brusselators mit 4 Boxen.

# A. Anhang

## A.1. Der Satz über implizite Funktionen

Man betrachte das Gleichungssystem $F(x,\tau) = 0,\ x \in \mathrm{R}^n,\ \tau \in \mathrm{R}$.
Man will nun untersuchen, unter welchen Bedingungen zu jedem $\tau$ aus einem gewissen Intervall $I_{n+1}$ genau ein $x$ existiert, sodaß $F(x,\tau) = 0$ erfüllt ist.
Dadurch ist eine Funktion $x = g(\tau)$ gegeben, für die

$$F(g(\tau),\tau) = 0$$

für alle $\tau \in I_{n+1}$ gilt.
Man sagt in diesem Fall, die Funktion $g$ werde durch die Gleichung $F(x,\tau) = 0$ implizit definiert oder auch $g$ entsteht durch Auflösen der Gleichung nach $x$.

Man kann auch ein etwas leichteres Problem betrachten.
Gegeben sei ein Gleichungssystem $H(u) = 0,\ u \in \mathrm{R}^{n+1}$.
Hier will man nur irgendeine Koordinate $u_i$ und ein Intervall $I_i$ finden, sodaß zu jedem $u_i \in I_i$ genau ein $\tilde{u} = (u_1, \ldots, \hat{u}_i, \ldots, u_{n+1})$ mit

$$H(\tilde{u}_{1:i-1}, u_i, \tilde{u}_{i:n}) = 0$$

existiert.
Im Unterschied zu vorhin ist es nun egal, nach welchen Koordinaten man auflöst.

Ein dritte Problemstellung betrachtet ebenfalls das Gleichungssystem
$H(u) = 0,\ u \in \mathrm{R}^{n+1}$, man will aber nur mehr ein Intervall $I$ und eine Funktion $u : I \longrightarrow \mathrm{R}^{n+1} :\ s \mapsto u(s)$ finden, sodaß

$$H(u(s)) = 0 \qquad \forall s \in I$$

gilt. Man läßt also die Forderung, daß man nach einer Koordinate parametrisiert, weg.

Es folgt nun der Satz über implizite Funktionen (siehe etwa [1]), der die Lösung für obige Probleme liefert.

**Satz A.1:** (Satz über implizite Funktionen)
Seien $U_1 \subseteq \mathrm{R}^n, U_2 \subseteq \mathrm{R}^m$ offene Mengen und
$F : U_1 \times U_2 \longrightarrow \mathrm{R}^n : \ (x, y) \mapsto F(x, y)$ eine stetig differenzierbare Abbildung.
Sei $(a, b) \in U_1 \times U_2$ ein Punkt so, daß $F(a, b) = 0$ und die $(n \times n)$-Matrix $\frac{\partial F}{\partial x}(a, b)$ invertierbar ist.
Dann gibt es offene Umgebungen $V_2 \subseteq U_2$ von $b$ und $V_1 \subseteq U_1$ von $a$ und eine stetig differenzierbare Abbildung $g : V_2 \longrightarrow V_1$ mit $F(g(y), y) = 0 \quad \forall y \in V_2$.
Ist $(x, y) \in V_1 \times V_2$ ein Punkt mit $F(x, y) = 0$, so folgt $x = g(y)$.

Die letzten zwei Problemstellungen sind nach dem Satz über implizite Funktionen (setze $m = 1$) lösbar, wenn die Jacobimatrix von $H$ $n$ linear unabhängige Spalten enthält. Das bedeutet gerade, daß bei allen regulären Punkten ein lokal eindeutig bestimmter Homotopieweg existiert.

Das erste Problem ist lösbar, falls $F_x$ invertierbar ist, also bei allen regulären Punkten mit Ausnahme der Umkehrpunkte. Das Wegfallen der Bedeutung von $\tau$ als Parameter ermöglicht es also, eine größere Anzahl von Gleichungssystemen zu lösen.

## A.2. Die QR-Zerlegung

**Satz A.2:** Sei $A$ eine $(m \times n)$-Matrix mit Rang $k$. Dann läßt sich A folgendermaßen in 3 Matrizen $Q, R$ und $P$ zerlegen:

$$A = QRP, \text{ wobei gilt:}$$

$Q$ ist eine orthogonale $(m \times m)$-Matrix, P eine $(n \times n)$-Permutationsmatrix und $R$ eine obere Dreiecksmatrix.

$$\begin{pmatrix} R_{11} & * & & & & * \\ 0 & \ddots & & & & \\ & & R_{kk} & * & \dots & * \\ & & & 0 & \dots & 0 \\ & & & & \ddots & \vdots \\ 0 & & & & & 0 \end{pmatrix}$$

Die QR-Zerlegung kann mit Hilfe des Schmidtschen Orthonormalisierungsverfahrens (siehe etwa [2]) gewonnen werden, besser ist jedoch ein Verfahren, das mit Householder-Matrizen arbeitet (siehe [4]). Man kann an der QR-Zerlegung einer Matrix ihren Rang

ablesen. Er ist der größte Index $k \leq m, n$, für den das Diagonalelement $R_{k,k}$ nicht Null ist.
Bei einer numerisch berechneten QR-Zerlegung wird $R_{k,k} = 0$ aufgrund von Rundungsfehlern nicht vorkommen, aber wenn der Rang nicht maximal ist, wird es einen Index $k+1$ geben, für den $R_{k+1,k+1}$ „viel kleiner“ als $R_{k,k}$ ist. Man vergleicht daher sukzessive aufeinanderfolgende Diagonalelemente von $R$ und bestimmt, wann diese Situation auftritt. Genauer gesagt gibt man eine Schranke *tolrang* vor und bestimmt den Rang folgendermaßen:

$$\mathrm{Rg}(A) = \begin{cases} \min\{k \leq m, n \mid \frac{R_{k+1,k+1}}{R_{k,k}} \leq tolrang\}, \text{ falls so ein } k \text{ existiert} \\ \min\{m, n\} \text{ sonst.} \end{cases}$$

## A.3. Das Newtonverfahren

Es sei $f : D \subseteq \mathrm{R}^n \longrightarrow \mathrm{R}^n$ gegeben, und man sucht eine Nullstelle $a \in \mathrm{R}^n$ mit $f(a) = 0$.

Dazu betrachtet man die Taylorreihenentwicklung von $f$ um $x_0$

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + O(\|x - x_0\|^2)$$

Wenn man für $x$ eine Nullstelle $a$ einsetzt und annimmt, daß $x_0$ schon nahe bei $a$ liegt, erhält man

$$0 = f(a) = f(x_0) + f'(x_0)(a - x_0) + O(\|a - x_0\|^2) \approx f(x_0) + f'(x_0)(a - x_0)$$

und daraus

$$a = x_0 + (a - x_0) \approx x_0 - f'(x_0)^{-1} f(x_0)$$

Man hofft nun, daß $x_0 - f'(x_0)^{-1} f(x_0)$ näher bei $a$ liegt als $x_0$.

Die Newtoniteration lautet also: Wähle $x_0$. Berechne $\Delta x_k$ aus dem Gleichungssystem

$$f'(x_k)\Delta x_k := -f(x_k)$$

$$x_{k+1} := x_k + \Delta x_k$$

Man gibt die Genauigkeit TOL vor und läßt das Newtonverfahren so lange laufen, bis $\|f(x_k\| \leq \mathrm{TOL}$ ist.

Natürlich muß noch bewiesen werden, daß das Verfahren konvergiert. Dazu folgender Satz:

**Satz A.3:** Sei $a$ wie oben eine Lösung von $f(x) = 0$, $f$ in einer Umgebung von $a$ 3-mal differenzierbar, und $e_k := x_k - a$ sei der Fehler nach dem k-ten Schritt.
Dann gilt:

$$e_{k+1} = \frac{1}{2} f'(x_k)^{-1} f''(x_k)(e_k, e_k) + O(\|e_k\|^2),$$

wobei $f''(x)(c,d) = \left(\sum_{j,l} \frac{\partial^2 f_i}{\partial x_j \partial x_l}(x) c_j d_l\right)_{i=1,..,n}$

**Beweis**: Man betrachtet die Taylorreihenentwicklung von $f$ um den Punkt $x_k$ und setzt dann $a$ ein:

$$0 = f(a) = f(x_k) + f'(x_k)(a - x_k) + \frac{1}{2} f''(x_k)(a - x_k, a - x_k) + O(\|a - x_k\|^3)$$

$$= f(x_k) + f'(x_k)(x_{k+1} - x_k - x_{k+1} + a) + \frac{1}{2} f''(x_k)(e_k, e_k) + O(\|e_k\|^3)$$

Wenn man die Iteration einsetzt, erhält man also

$$0 = -f'(x_k)(e_{k+1}) + \frac{1}{2} f''(x_k)(e_k, e_k) + O(\|e_k\|^3)$$

□

# Symbolverzeichnis

| | |
|---|---|
| $\mathbb{N} = \{1, 2, 3, \dots\}$ | Menge der natürlichen Zahlen |
| $\mathbb{R}$ | Körper der reellen Zahlen |
| $\mathbb{C}$ | Körper der komplexen Zahlen |
| $\mathbb{K}$ | beliebiger Körper |
| $f'(u) = f'(x, y)$ | Ableitung der Abbildung $f$ nach allen Komponenten $u = (x, y) = (x_1, \dots, x_N, y_1, \dots, y_M)$ |
| $f_x(u)$ | Ableitung von $f$ nur nach $x = (x_1, \dots, x_n)$ |
| $\frac{\partial f}{\partial x_k}$ | Ableitung von $f$ nach der $k$-ten Komponente |
| $f_{xx}$ | zweifache Ableitung von $f$ nach $x$ |
| $f''(u)$ | zweifache Ableitung von $f$ nach allen Komponenten |
| $f''(u)(v, v)$ | $\sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial^2 H_i}{\partial u_j \partial u_k} v_k v_j$ |
| $O(h^k)$ | $f(h) = O(h^k) \iff h^{-k} f(h)$ ist beschränkt für $h \to 0$ |
| max | Maximum |
| $\dim(V)$ | Dimension des Vektorraumes $V$ |
| $\mathrm{Ker}(f)$ | Kern der Abbildung $f$ |
| $\mathrm{Bi}(f)$ | Bild der Abbildung $f$ |
| $\mathrm{Rg}(A)$ | Rang der Matrix $A$ |
| $I_n$ | $n$-dimensionale Einheitsmatrix |
| $A^T$ | die zu $A$ transponierte Matrix |
| $A_{-,k}$ | $k$-te Spalte der Matrix $A$ |
| $A_{a:b,k}$ | Zeilen $a$ bis $b$ der $k$-ten Spalte von $A$ |
| $\oplus$ | direkte Summe |
| $\forall$ | für alle |
| $\varnothing$ | leere Menge |
| $\lvert \cdot \rvert : \mathbb{K} \to \mathbb{R}$ | Absolutbetrag |
| $\lVert \cdot \rVert : \mathbb{R}^n \to \mathbb{R}$ | beliebige Norm auf dem $\mathbb{R}^n$ |
| $[a, b] \subseteq \mathbb{R}$ | abgeschlossene Intervalle |

# Literaturverzeichnis


[1] Forster O. (1984). *Analysis 2*. Friedr. Vieweg & Sohn, Braunschweig, 5. Aufl.

[2] Jänich K. (1991). *Lineare Algebra*. Springer-Verlag, 4.Aufl.

[3] Cox D., Little J., O'Shea D. (1992) *Ideals, Varieties and Algorithms*. Springer-Verlag.

[4] Ostermann A. *Numerik 2*. Vorlesungsskriptum aus dem Sommersemester 96.

[5] Pauer, Franz. *Polynome in mehreren Variablen*. Vorlesungsskriptum aus dem Sommersemester 96.

[6] Pauer, Franz. *Algebraische Geometrie*. Vorlesungsskriptum aus dem Wintersemester 96/97.

[7] Prigogine I., Lefever R. (1968) *Symmetry Breaking Instabilities in Dissipative Systems.II*. J.Chem.Phys., 48, 1695-1701.

[8] Deuflhard P., Hohmann A. (1991) *Numerische Mathematik*. W. de Gruyter Lehrbuch.

# Lebenslauf

**Persönliche Daten**

| | |
|---|---|
| Name | Günther Eibl |
| Geburtsdaten | 14. Mai 1971 in Salzburg |
| Eltern | Prof. Mag. Hans Eibl und Hedda Eibl, geb. Vogl |
| Staatsbürgerschaft | Österreich |
| Familienstand | ledig |

**Ausbildung**

| | |
|---|---|
| 1977-1981 | Besuch der Volksschule Pradl Ost I in Innsbruck |
| 1981-1989 | Besuch des Bundesgymnasium und Bundesrealgymnasium Reithmannstraße in Innsbruck |
| 30. Mai 1989 | Matura |
| Oktober 89 - September 90 | Präsenzdienst |
| Oktober 1990 | Beginn des Doppelstudiums Mathematik (Stzw. Mathematik) und Physik (Stzw. Physik) |
| WS 94/95 - WS 95/96 | Abhaltung von Fachtutorien |
| WS 95/96 - SS 97 | Mitarbeit in der Studienrichtungsvertretung Mathematik |