H. Lustfeld, 25.2. 2004

Vorlesung Moderne Probleme der Geophysik mit Methoden der theoretischen Physik I, WS 2002/3 und WS 2003/4

# Anwendung der Theorie neuronaler Netze auf Wasserstandsvorhersagen an der deutschen Nordseeküste

### Einführung

Die Strömungsverhältnisse an der deutschen Nordseeküste sind besonders schwierig: es gibt infolge von Ebbe und Flut wegen des flachen Wassers starke Strömungen, vermehrt durch die Strömungen der Flüsse von Elbe, Weser, Ems und Rhein sowie komplizierte Uferstrukturen durch die dem Festland vorgelagerten Inseln. Dazu kommen ständig wechselnde Wetterverhältnisse. Es ist danach klar, daß eine Modellierung all dieser Einflüsse auf die Wasserstände an den verschiedenen Küstenorten sehr schwierig ist.

Unter diesen Umständen bietet sich eine andere Methode an, die immer dann erfolgversprechend ist, wenn große Datenmengen vorliegen und man die Vermutung hat, daß die Aussagen, die man erhalten möchte, *cum grano salis* eine - wenn auch komplizierte - deterministische Funktion von Eingangsgrößen ist. Einen Hinweis darauf ersieht man aus der Tatsache, daß erfahrenen Spezialisten treffsicherere Aussagen bei Kenntnis der Eingangsgrößen gelingen.

Diese andere Methode ist die der neuronalen Netze (und der damit verwandten Support Vektor Maschinen). Da Problemkreise der eben beschriebenen Art nicht nur in der speziellen Fragestellung der Geologie sondern in vielen Bereichen vorkommen, möchte ich zunächst allgemeiner auf die neuronalen Netze eingehen, um dann zum Schluß auf ihre Nutzanwendung in der Geologie, bei den Wasserständen zurückzukommen.

# Neuronale Netze: Allgemeine Aussagen

Offenbar besteht das Gehirn aus Neuronen, etwa  $10^{11}$  sind es. Die Neuronen sind miteinander gekoppelt, jedes Neuron kann durch Impulse anderer Neuronen selbst aktiviert werden und Impulse aussenden. Dabei ist schon ein Neuron eine ziemlich komplizierte Angelegenheit. Es liegen daher 2 Fragestellen nahe, i) wie funktioniert ein Neuron, ii) was ergibt sich aus der Verschaltung vieler Neuronen.

Bei den künstlichen neuronalen Netzen steht immer ii) im Vordergrund. Man möchte wissen, was die Vernetzung der Neuronen zur Folge hat. Dabei geht es um 3 Aspekte: a) Man möchte ein Verständnis dafür bekommen, wie ein Gehirn arbeitet, b) man möchte eine Schaltung mit Neuronen auf einem Chip erzeugen, der dann besonders schnell und billig bestimmte Aufgaben wahrnehmen kann, c) man möchte Ideen für neue mathematische Verfahren bekommen. Im Zusammenhang mit dieser Vorlesung interessiert besonders c).

In den künstlichen neuronalen Netzen sind die Neuronen immer folgendermaßen verbunden: Im Neuron i münden Eingänge (Dendriten von anderen Neuronen mit Werten  $\{x_{\nu}\}$ , die von anderen Neuronen kommen und durch die Synapsen verschieden gewichtet werden. Der Ausgangswert des Neurons,  $y_i$ , hängt von den Eingaben  $\{x_{\nu}\}$  und deren Gewichtungen  $\{w_{\nu}\}$  ab:  $y_i = y_i(\{w_\nu x_\nu\})$ 



(1.1)

## Neuronale Netze: Das Modell von McCulloch und Pitts

Das einfachste Modell für neuronale Netzwerke ist das von McCulloch und Pitts. In diesem Modell sind die Gewichte sowie die Eingabewerte  $x_{\nu}$  und die Ausgabewerte  $y_i$  alle 0 oder 1. Es ist also ein binäres Modell und wird auch heute noch für logische Schaltungen verwendet. Das Schema des allgemeinsten Neuronentypus dieses Modells ist in Fig.1.2





gezeigt. Die (binäre) Ausgabe  $y_i$  wird durch  $\Theta$  Funktionen

$$\Theta(x) = \begin{cases} 1 \text{ für } x \ge 0\\ 0 \text{ für } x < 0 \end{cases}$$
(1.2)

bestimmt:

$$y_{i} = \Theta(\sum_{\nu=1}^{N} x_{\nu} - m_{i}) \cdot \prod_{\mu=1}^{M} \Theta(-z_{\mu})$$
(1.3)

 $m_i$  ist die Hemmschwelle, das Neuron 'feuert' (d.h. setzt  $y_i = 1$ ), wenn diese erreicht wird, es sei denn, daß es z-Eingänge gibt mit  $z_{\mu} = 1$ . Die durch gefüllte Kreise gekennzeichneten Eingänge bilden eine absolute Hemmung. Einfache Beispiele logischer Schaltungen sind in Fig.1.3 gezeigt. Eine weitere Schaltung zeigt Fig.1.4. Nur bei Eingabevektor  $\mathbf{x} = (1, 0, 1)$ gibt es am Ausgang 1, sonst 0. Ein anderes Beispiel zeigen Fig.1.5 und Fig.1.6. Nur bei den in den Figuren angegebenen Eingabevektoren gibt es am Ausgang 1, sonst 0. Manchmal ist statt einer absoluten Hemmung (gekennzeichnet durch gefüllte Kreise) nur eine relative Hemmung (gekennzeichnet durch Kreisränder erwünscht, die die Hemmschwelle um 1 anheben kann. In Fig.1.7 ist gezeigt, daß dazu keine neuen Schaltelemente erforderlich sind.

Die vorangegangenen Beispiele machen die folgende Aussage sehr plausibel:

jede binäre Funktion, die von binären Variablen abhängt, la]ßt sich (1.4) mit dem neuronalen Netz von McCulloch-Pitts darstellen

Ein Beweis wird hier nicht gebracht, da im nächsten Abschnitt eine sehr viel allgemeinere Aussage bewiesen werden wird.



Abbildung 1.3: einfache logische Schaltungen beim McCulloch-Pitts Modell



Abbildung 1.4: einfache Schaltung. Eingetragen ist der Eingabevektor, bei dem die Ausgabe 1 ist.



Abbildung 1.5: Schaltung beim McCulloch-Pitts Modell. Eingetragen sind die Eingabevektoren, bei denen die Ausgabe 1 ist.



Abbildung 1.6: Schaltung beim McCulloch-Pitts Modell. Eingetragen ist der Eingabevektor, bei dem die Ausgabe 1 ist.



Abbildung 1.7: Relative Hemmung, gekennzeichnet durch offenen Kreis, wird im McCulloch-Pitts Modell durch Schaltung mit absoluter Hemmung erreicht

## Das universale Approximationstheorem und feed forward Netze

Es gibt verschiedene Arten von Approximationen einer Funktion. Die aus der Quantenmechanik bekannteste ist die Entwicklung nach einem Eigenfunktionensystem eines hermiteschen Operators. Eine andere ist die Entwicklung nach wavelets. Diese Entwicklungen sind systematisch in dem Sinne, daß eine genaue Vorschrift für die Berechnung der Entwicklungskoeffizienten vorliegt.

Einer großen Klasse neuronaler Netzwerke liegt ein anders gearteter Ansatz zu Grunde, der auf einer *nichtsystematischen* Entwicklung nach *sigmoiden* Funktionen

$$\varphi(s)$$
 sigmoid  $\leftrightarrow \varphi$  monoton wachsend,  $\varphi(-\infty) = 0, \ \varphi(\infty) = 1$  (2.1)

beruht. Diese nichtsystematische Entwicklung stützt sich auf den folgenden Entwicklungssatz, der in Anhang A bewiesen ist<sup>1</sup>:

Sei

$$F: \mathbb{R}^N \to \mathbb{R}$$

eine in einem abgeschlossenen Bereich A definierte stetige Funktion. Sei ferner  $\varphi(s)$ eine sigmoide Funktion, dann gilt für jedes  $\epsilon > 0$  und alle  $\mathbf{x}\epsilon A$ 

$$|F(\mathbf{x}) - \sum_{i=1}^{M} \alpha_i \varphi(\sum_{l=1}^{N} w_{il} x_l - \theta_i)| < \epsilon$$
(2.2)

für geeignet gewählte Parameter  $\alpha_i$ ,  $w_{il}$  und  $\theta_i$ , wobei  $M < \infty$ .

Gl.(2.2) ist technisch besonders gut für neuronale Netze geeignet, i) wegen der sigmoiden Form von  $\varphi$  als Funktion jedes Neurons, ii) wegen der Gleichheit aller  $\varphi$ . Ein typisches neuronales Netzwerk, das nichts weiter ist als die Realisierung des universalen Approximationstheorems, wird in Fig.2.1 gezeigt, wobei schon berücksichtigt ist, daß die Ausgabe ein Vektor der Dimension  $N_a$  sein kann. Solche Netze, die keine Querverbindungen innerhalb der Schichten und keine Rückkopplung haben, werden als *feed forward* Netze bezeichnet

Gl.(2.2) hat eine geometrische Deutung. Die sigmoiden Funktionen ändern sich in der Umgebung von 0 am stärksten. Der Ausdruck

$$\mathbf{w}_i \mathbf{x} - \theta_i = 0 \text{ mit } \mathbf{w}_i = (w_{i1}, \dots, w_{iN})$$
(2.3)

ist die Darstellung einer Hyperebene in der Hesseschen Normalform mit einem Abstand  $d_i = \theta_i / |\mathbf{w}_i|$  vom Ursprung. Senkrecht zur Hyperebene sind in der Umgebung der Hyperebene die Änderungen des jeweiligen  $\varphi$  am stärksten.

Der universale Approximationssatz sagt nichts darüber aus, wie die Parameter  $\alpha_i$ ,  $w_{il}$ und  $\theta_i$  zu bestimmen sind. Dies geschieht durch eine Kostenfunktion  $E_{Kost}$ , die folgendermaßen definiert ist.

<sup>&</sup>lt;sup>1</sup>Der Einfachheit halber ist angenommen, daß die Ausgabenfunktion F ein Skalar ist. Mit exakt der gleichen Beweisführung gilt die Aussage, wenn die Ausgabenfunktion ein  $N_a$  dimensionaler Vektor ist. Dann sind lediglich die Ersetzungen  $F \to \mathbf{F} \epsilon R^{N_a}$  und  $\alpha_i \to \alpha_i \epsilon R^{N_a}$  vorzunehmen.



Abbildung 2.1: Ein neuronales feed forward Netz, konstruiert gemäß Gl.(2.2) besteht aus Eingabeschicht (input layer), innerer Schicht (hidden layer) und Ausgabeschicht (output layer). Es ist schon berücksichtigt, daß die Ausgabe ein Vektor sein kann Die äußeren Schichten haben Verbindungen zu allen Neuronen der inneren Schicht, eingezeichnet sind nur die von  $x_1$  und die von  $F_2$ .

Gegeben  $N_E$  Eingabewerte,  $\mathbf{x}_{\mu}$ ,  $\mu = 1, ..., N_E$  mit dazu bekannten, wahren Ausgabewerten  $d_{\mu}$  sowie den berechneten  $F(\mathbf{x}_{\mu})$ . Dann ist

$$E_{Kost} = E_{Kost}(\alpha_i, \mathbf{w}_i, \theta_i) = \frac{1}{2N_E} \sum_{\mu=1}^{N_E} (d_\mu - F(\mathbf{x}_\mu))^2$$
(2.4)

Die Parameter  $\alpha_i, \mathbf{w}_i, \theta_i$  ergeben sich aus der Bedingung, daß E bezüglich dieser minimal sein muß.

$$\frac{\partial}{\partial \alpha_i} E_{Kost} = 0, \ \frac{\partial}{\partial w_{ij}} E_{Kost} = 0, \ \frac{\partial}{\partial \theta_i} E_{Kost} = 0 \text{ für alle } i, j$$
(2.5)

Was ist der Vorteil des universalen Approximationstheorems, warum nimmt man nicht ein vollständiges Orthogonalsystem, bei dem die Entwicklungskoeffizienten gemäß einer Vorschrift berechnet werden können und somit das absolute Minimum von E erreicht wird? Die Antwort liegt im numerischen Aufwand; eine Entwicklung nach Orthogonalsystemen wird bei einer Dimension des Eingabevektors  $\mathbf{x}$  jenseits von N > 3 sehr CPU intensiv. Hat  $\mathbf{x}$  etwa die Dimension N = 100 - eine Dimension, die z.B. im Sprachprogramm NETtalk und im Problem der Wasserstände mit Leichtigkeit übertroffen wird -, dann wären selbst unter der Annahme, daß pro Richtung nur 2 Terme der Entwicklung nach einem Orthogonalsystem nötig sind, 2<sup>100</sup> Koeffizienten zu berechnen, die sämtlich über 100 dimensionale Integrale erhalten werden müßten. Solche Rechnungen liegen jenseits von Gut und Böse. Bei den neuronalen Netzen wird ausgenutzt, daß in vielen Problemen  $E_{Kost}$  außer dem absoluten Minimum Nebenminima hat, die nicht wesentlich höher liegen, aber numerisch um viele Größenordnungen schneller erreichbar sind.

### Probleme der Minimumsuche, Backpropagation und Lernen bei feed forward Netzen

Eine sehr robuste Methode, ein Minimum der Kostenfunktion E zu bekommen, beruht auf der folgenden Überlegung: Sei u der Vektor, dessen Komponenten sämtliche Parameter sind,

$$\mathbf{u} = (\alpha_1 ..., w_{11} ..., \theta_1 ...) \tag{2.6}$$

Dann ist

$$dE = \frac{\partial}{\partial \mathbf{u}} E d\mathbf{u}$$

dE ist am stärksten negativ, wenn

$$d\mathbf{u} = -\epsilon \cdot \frac{\partial}{\partial \mathbf{u}} E, \ \epsilon \text{ positiv und infinitesimal}$$
(2.7)

Die Minimumsuche erfolgt deshalb so: Man startet mit einem  $\mathbf{u}$ , das als Anfangswert passend erscheint und versucht, das Minimum schrittweise zu erreichen, wobei man bei jedem Schritt  $\mathbf{u}$  um  $\Delta \mathbf{u}$  ändert mit

$$\Delta \mathbf{u} = -\eta \cdot \frac{\partial}{\partial \mathbf{u}} E, \ \eta \text{ positiv und klein}$$
(2.8)

Dies Verfahren für die Änderung der Parameter  $\Delta \mathbf{u}$  ist die Gradientenmethode und läuft bei den neuronalen Netzen unter dem Namen *Backpropagationsverfahren*.

Ein Problem bei diesem Verfahren ist die Möglichkeit, in ein Nebenminimum hineinzulaufen, das weit oberhalb des absoluten Minimums liegt,vgl. Fig.2.2. Um dies zu



Abbildung 2.2: Ein möglicher und gefürchteter Kurvenverlauf der Kostenfunktion E. Es gibt eine Menge lokaler Minima, die weit oberhalb des absoluten Minimums liegen. Allerdings gibt es auch - und das nützen Methoden wie simulated annealing aus, sehr viele Nebenminima, die nur unwesentlich oberhalb des absoluten Minimums liegen.

vermeiden, werden Verfahren angewandt, die unter dem Namen simulated annealing bekannt sind und darauf beruhen, mit einer gewissen Wahrscheinlichkeit  $\Delta w$  ein  $\Delta \mathbf{u}$ , das Eum  $\Delta E$  ändert, auszuwählen, um auf diese Weise aus einem lokalen Minimum herauszuspringen:

$$\Delta u = \begin{cases} \Delta u_v \text{ mit Wahrscheinlichkeit } \Delta w = 1 \text{ bei } \Delta E < 0 \\ \Delta u_v \text{ mit Wahrscheinlichkeit } \Delta w = e^{-\beta \Delta E} \text{ bei } \Delta E > 0 \\ 0 \text{ mit Wahrscheinlichkeit } \Delta w = 1 - e^{-\Delta E} \text{ bei } \Delta E > 0 \end{cases}$$
(2.9)

Dies ist analog zu einer thermodynamischen Fluktuation mit 'Temperatur' =  $1/\beta$ , und wie dort durch Absenkung der 'Temperatur' allmälich die Fluktuationen herabgesetzt werden, so auch hier in diesem Verfahren. Daher der Name.

Bei feed forward Netzen wird eine Alternative darin gesehen, mehrere innere Schichten im Netzwerk einzurichten, obwohl dies nach dem universalen Approximationstheorem nicht nötig ist. Die *Erfahrung* hat gezeigt, daß unter Umständen dadurch lokale Minima, die weit über dem absoluten liegen, unterdrückt werden. Solche Netze sind auch unter dem Namen *multilayer perceptrons* bekannt.

Was man als *Lernen* bei den feed forward Netzen bezeichnet, ist nichts anderes als die sukkzessive Änderung der Parameter, um das Minimum zu finden, also etwas ganz prosaisches - nichts philosophisches.

# Beispiele von feed forward Netzen: das Perzeptron, ein Übungsbeispiel und das Sprachprogramm NETtalk



Abbildung 2.3: Das Perzeptron hat als innere Schicht nur 1 Neuron

• Beim *Perzeptron* nimmt man aus der Summe in Gl.(2.2) nur einen Summanden. Bei dieser rabiaten Vereinfachung können Funktionen nur noch grob oder gar nicht dargestellt werden. Es gibt jedoch Funktionen, die nur die Werte 0 oder 1 annehmen, wo nur eine *ja-nein* Entscheidung erwartet wird, etwa bei der Frage, ob sehr hohe Wasserstände vorausgesagt werden müssen. In diesen Fällen kann unter gleich zu spezifizierenden Bedingungen dann ein Summand ausreichen<sup>2</sup>

$$y = 2\varphi(\sum_{l=1}^{N} w_l x_l - \theta) - 1$$
 (2.10)

Mit den Vektoren

$$\mathbf{w} = (w_1, w_2 \cdots, w_N, \theta)$$

$$\mathbf{x} = (x_1, x_2 \cdots, x_N, -1)$$
(2.11)

ist dies

$$y(\mathbf{x}) = 2\varphi(\mathbf{w}\mathbf{x}) - 1 \tag{2.12}$$

 $\mathbf{w}/|\mathbf{w}|$  ist der Normalenvektor einer Hyperebene, die die 'ja' Werte von den 'nein' Werten trennt. Die Brauchbarkeit des Perzeptrons hängt demnach davon ab, wie gut die Werte durch eine Hyperebene separiert werden können. Zur Bestimmung des Vektors  $\mathbf{w}$  wird versuchsweise die Gradientenmethode herangezogen. Die Kostenfunktion ist

$$E = E(\mathbf{w}) = \frac{1}{2N_E} \sum_{\mu=1}^{N_E} (d_\mu - y(\mathbf{x}_\mu))^2$$
(2.13)

und ihr Gradient

$$\nabla_{\mathbf{w}} E = -\frac{2}{N_E} \sum_{\mu=1}^{N_E} \mathbf{x}_{\mu} \varphi'(\mathbf{w} \mathbf{x}_{\mu}) (d_{\mu} - y(\mathbf{x}_{\mu}))$$
(2.14)

Damit wird von der Gradientenmethode die folgende Iteration vorgeschlagen:

$$\mathbf{w}^{(n+1)} = \mathbf{w}^{(n)} + \eta \mathbf{x}_{\mu} \varphi'(\mathbf{w} \mathbf{x}_{\mu}) (d_{\mu} - y(\mathbf{x}_{\mu}))$$
(2.15)  
$$\eta > 0, \ \eta \ll 1, \qquad \mu \text{ zufällig gewählt}$$

Zwar ist  $\varphi' \ge 0$ , da  $\varphi$  eine sigmoide Funktion ist. Andererseits ist  $\varphi' \ll 1$  außer in einer Umgebung um 0, so daß die von der Hyperebene weiter entfernten  $\mathbf{x}_{\mu}$  praktisch keinen Einfluß haben. Damit entstehen eine Menge lokaler Minima. In diesem Fall ist von Hand diese Schwäche zu beseitigen:

$$\mathbf{w}^{(n+1)} = \mathbf{w}^{(n)} + \eta \mathbf{x}_{\mu} (d_{\mu} - y(\mathbf{x}_{\mu}))$$
(2.16)  
$$\eta > 0, \ \eta \ll 1, \qquad \mu \text{ zufällig gewählt}$$

Man kann zeigen, daß das Iterationsverfahren von Gl.(2.16) immer konvergiert.

<sup>&</sup>lt;sup>2</sup>wobei die ja-nein Entscheidung durch die sigmoide Funktion  $\varphi$  aufgeweicht ist.

• Ein einfaches Beispiel, in dem mehrere Summanden aus Gl.(2.2) benötigt werden, ist die Aufgabe sin(ax) durch sigmoide Funktionen darzustellen. Gewählt wird die Fermifunktion

$$\varphi(s) = \frac{1}{1 + e^{-s}} \tag{2.17}$$

Gesucht ist eine Darstellung der Form

$$\sin(ax) = \sum_{i} \alpha_{i} \varphi(w_{i}x - \theta_{i}) + \epsilon(x), \ |\epsilon(x)| < \epsilon$$
(2.18)

Diese Aufgabe ist von M. Zilly programmiert, wobei das Gradientenverfahren benutzt wurde. Die Ergebnisse sind in Fig.2.4 geplottet.



Abbildung 2.4: Plot der Approximation von  $\sin(x)$  im Intervall  $[0, 2\pi]$ . Anzahl der Neuronen ist 9,  $\eta = 0.1$ , (vgl Gl.(2.8)). Rote Linie:  $\sin(x)$ , schwarze Linie: Approximation. a)  $E_{Kost} < 10^{-2}$ , b)  $E_{Kost} < 10^{-6}$ .

• Im Gegensatz zum Perzeptron ist NETtalk ein echtes feed forward neuronales Netz, und im Gegensatz zum Übungsbeispiel hat der Eingabevektor **x** hohe Dimension. NETtalk wurde entwickelt für die Aufgabe, geschriebenen Text in Sprache umzuwandeln. Seine Eingabeschicht besteht aus 7 Gruppen von je 29 Elementen, die Dimension des Eingabevektors **x** ist also 203 - eine Dimension, in der die üblichen Entwicklungen wie Fourierzerlegung längst am Ende sind.

Die innere Schicht von NETtalk besteht aus 80 Neuronen, die Ausgabeschicht besteht aus 26 Neuronen.

NETtalk arbeitet mit derselben großen Zuverlässigkeit wie die bis zum Erscheinen von NETtalk üblichen Programme, die in hochkomplexer Weise den Silbenketten linguistische Regeln zuordnen - natürlich für jede Sprache verschiedene. Dies alles braucht NETtalk nicht.

Beim 'Lesenlernen' von NETtalk fand man die merkwürdige Analogie, daß wie bei Vorschulkindern 'Leseschwächen' auftraten, die mit fortschreitendem 'Lernprozeß' nach und nach verschwanden.

#### Kohonen's Datenanalyse und -kompression

Die hier behandelten Verfahren basieren auf Ideen von T. Kohonen. Sie heißen auch self organizing feature maps (SOFM). Die Bezeichnung 'neuronale Netze' ist bei den Verfahren von Kohonen üblich, aber nur historisch zu erklären sowie dadurch, daß diese Verfahren von der Anwendung her in Konkurrenz zu den feed forward Netzen treten. Ich gehe hier nur auf die Verfahren selbst ein, nicht darauf, wie sie historisch aus Beobachtungen des Verhaltens von rückgekoppelten Netzen und von Aktivitätsprozessen im Gehirn entstanden sind.

Es ist wichtig zu wissen, welche Dimension ein Datensatz<sup>1</sup>  $P_D$  tatsächlich hat. Z. B. werden zur Voraussage der Wasserstände an der deutschen Nordseeküste Daten zahlreicher, auch ausländischer Stationen herangzogen. Dies gibt einen Eingabevektor (bei K Stationen)

 $\mathbf{x} =$ 

 $(Pegelstand_{1}, Lufttemperatur_{1}, Wassertemperatur_{1}, Wind_{x,1}, Wind_{y,1}, Luftdruck_{1}, ..., Pegelstand_{2}, Lufttemperatur_{2}, Wassertemperatur_{2}, Wind_{x,2}, Wind_{y,2}, Luftdruck_{2}, ..., Pegelstand_{y,2}, Luftdruck_{y,2}, ...,$ 

. (3.1)

 $Pegelstand_{K}, Lufttemperatur_{K}, Wassertemperatur_{K}, Wind_{x,K}, Wind_{y,K}, Luftdruck_{K}, ..., Zeit, Sonnenstand, Mondstand)$ 

der sehr schnell eine hohe Dimension N hat. Der Datensatz  $P_D$  braucht insgesamt aber überhaupt nicht die Dimension N zu haben. Z.B. sind alle Temperaturen, alle Windgeschwindigkeiten, alle Luftfeuchtigkeiten korreliert, was die Dimension reduziert.

Auch braucht zwischen der Dimension von  $\mathbf{x}$  und der Dimension M von  $P_D$  keine Beziehung zu bestehen. Beispiel: Spirale im  $\mathbb{R}^N$ .  $\mathbf{x}$  hat die Dimension N,  $P_D$  der Spirale hat Dimension<sup>2</sup> M = 1. Die Punktmenge dieser Spirale ist also recht einfach zu beschreiben. Ganz allgemein gibt es eine *umkehrbar eindeutige und stetige* Abbildung zwischen Parameterraum  $\mathbb{R}_p^M$  und  $P_D$ 

$$\mathbf{c} : P_D \epsilon R^N \to P_D^{-1} \epsilon R_p^M, \mathbf{c} \text{ stetig}$$

$$\hat{\mathbf{x}} : P_D^{-1} \epsilon R_p^M \to P_D \epsilon R^N, \hat{\mathbf{x}} \text{ stetig}$$
es ist  $\mathbf{c} = \hat{\mathbf{x}}^{-1} \text{ auf } P_D$ 

$$(3.2)$$

<sup>&</sup>lt;sup>1</sup>streng genommen hat ein Datensatz der Geophysik immer die Dimension 0, da er aus endlich vielen Punkten (allerdings können diese in der Größenordnung von  $10^5$  und mehr liegen) besteht. Man kann aber eine Dichte  $\rho(\mathbf{x})$  einführen, eine Glättung ergibt dann eine Mannigfaltigkeit  $P_D$  mit der 'echten' Dimension des Datensatzes M. Im Folgenden wird immer in diesem Sinne vom Datensatz und seiner Dimension gesprochen werden.

<sup>&</sup>lt;sup>2</sup>hier wird die Dimension durch die Anzahl der Tangentenvektoren, die die Basis des Tangentenraumes bilden, bestimmt. Das ist stückweise durch die Dimension des Parameterraumes  $R_p^M$  festgelegt, der den Datensatz beschreibt.

Eine vernünftige Annahme M' über die Dimension M vorausgesetzt kann man  $\hat{\mathbf{x}}$  und **c** berechnen, indem man eine Kostenfunktion aufstellt, deren Minimum erreicht ist, wenn  $\hat{\mathbf{x}}$  und **c** gefunden sind. Eine solche ist

$$D_0 = \frac{1}{2} \int d^N x \,\rho(\mathbf{x}) \left[\mathbf{x} - \hat{\mathbf{x}}(\mathbf{c}(\mathbf{x}))\right]^2 \tag{3.3}$$

Diese Funktion reicht bei den in der Geophysik behandelten Problemen nicht aus, da die Daten diskret und verrauscht sind, so daß eine Unschärfefunktion entweder im  $\mathbb{R}^N$  oder im  $\mathbb{R}_p^{M'}$  eingeführt werden muß. Hier wird das letztere gewählt und man erhält

$$D = \frac{1}{2} \int d^{N} x \,\rho(\mathbf{x}) d^{M'} \nu \,\pi_{a}(\boldsymbol{\nu}) \left[\mathbf{x} - \hat{\mathbf{x}}(\mathbf{c}(\mathbf{x}) + \boldsymbol{\nu})\right]^{2}$$
(3.4)  

$$\rho(\mathbf{x}) = \text{Dichte des Datensatzes, } \rho(\mathbf{x}) \neq 0 \text{ für } \mathbf{x} \,\epsilon P_{D}$$
  

$$\pi_{a}(\boldsymbol{\nu}) = \text{um 0 konzentrierte Unschärfefunktion der Reichweite } a, z.B.$$
  

$$\pi_{a}(\boldsymbol{\nu}) \propto e^{-\nu^{2}/a^{2}}$$

Bei einer infinitesimalen Variation der Funktion  $\hat{\mathbf{x}}(\mathbf{c})$  um  $\delta \hat{\mathbf{x}}(\mathbf{c})$  (vgl. Fig.3.1) ändert sich D um  $\delta D$ . Im Minimum muß jedoch bei jeder infinitesimalen Variation  $\delta \hat{\mathbf{x}}(\mathbf{c})$  gelten:  $\delta D = 0$ . Da jede infinitesimale Variation  $\delta \hat{\mathbf{x}}(\mathbf{c})$  dargestellt werden kann als Zusammensetzung lokaler, um beliebige  $\mathbf{c}_0$  konzentrierter infinitesimaler Variationen, vgl. Fig.3.1, wird die Mininumsbedingung bezüglich einer Variation  $\delta \hat{\mathbf{x}}(\mathbf{c})$  äquivalent zu folgender Bedin-



Abbildung 3.1: globale Variation einer Funktion  $\hat{\mathbf{x}}(\mathbf{c})$  um  $\delta \hat{\mathbf{x}}(\mathbf{c})$  sowie lokale Variation in beliebig kleiner Umgebung von  $\mathbf{c}_0$ . Jede globale Variation kann aus lokalen aufgebaut werden.

gung

$$\frac{\partial}{\partial \hat{\mathbf{x}}} \hat{D}(\mathbf{c}_0) = 0 \tag{3.5}$$

$$mit$$

$$\hat{D}(\mathbf{c}_0) = \frac{1}{2} \int d^N x \,\rho(\mathbf{x}) \pi_a(\mathbf{c}_0 - \mathbf{c}(\mathbf{x})) \left[\mathbf{x} - \hat{\mathbf{x}}\right]^2 \tag{3.6}$$

und somit

$$0 = -\int d^N x \,\rho(\mathbf{x})\pi_a(\mathbf{c}_0 - \mathbf{c}(\mathbf{x})) \left[\mathbf{x} - \hat{\mathbf{x}}\right]$$
(3.7)

Man hat damit eine *explizite* Gleichung für  $\hat{\mathbf{x}}(\mathbf{c}_0)$ . Die Lösung lautet

$$\hat{\mathbf{x}}(\mathbf{c}_0) = \frac{\int d^N x \,\rho(\mathbf{x}) \pi_a(\mathbf{c}_0 - \mathbf{c}(\mathbf{x})) \,\mathbf{x}}{\int d^N x \,\rho(\mathbf{x}) \pi_a(\mathbf{c}_0 - \mathbf{c}(\mathbf{x}))}$$
(3.8)

Diese Lösung ist formal, da ja  $\mathbf{c}(\mathbf{x})$  nicht bekannt ist. Man kann aber sehen, daß  $\hat{\mathbf{x}}(\mathbf{c}_0)$ und die Ableitung selbst dann stetige Funktionen sind, wenn  $\mathbf{c}(\mathbf{x})$  unstetige Stellen hat. Dies ist eine Folge der Unschärfefunktion  $\pi_a$ .

Eine iterative Methode zur Lösung ist wieder die Gradientenmethode. Aus Gl.(3.6) ergibt sich

$$\hat{\mathbf{x}}^{(n+1)}(\mathbf{c}_0) = \hat{\mathbf{x}}^{(n)}(\mathbf{c}_0) + \eta \int d^N x \,\rho(\mathbf{x}) \pi_a(\mathbf{c}_0 - \mathbf{c}(\mathbf{x})) \left[\mathbf{x} - \hat{\mathbf{x}}^{(n)}(\mathbf{c}_0)\right]$$
(3.9)

Die Variation von D um  $\delta \mathbf{c}$  muß im Minimum ebenfalls verschwinden. Analog kann lokale Variation an  $\mathbf{c}$  um  $\mathbf{x}_0$  vorgenommen werden. Dann ist in Analogie zu  $\hat{D}$  der Ausdruck

$$\check{D} = \frac{1}{2} \int d^{M'} \nu \,\pi_a(\boldsymbol{\nu}) \left[ \mathbf{x}_0 - \hat{\mathbf{x}}(\mathbf{c} + \boldsymbol{\nu}) \right]^2 \tag{3.10}$$

bezüglich  $\mathbf{c}$  zu minimieren. Leider gibt die Bedingung

$$\frac{\partial}{\partial \mathbf{c}}\check{D}(\mathbf{x}_0) = 0 \tag{3.11}$$

keinen expliziten Ausdruck für  $\mathbf{c}(\mathbf{x}_0)$ . Kohonen machte an dieser Stelle aus der Not eine Tugend, indem er für  $\mathbf{c}$  diskrete Werte einsetzte:

$$\begin{aligned} \mathbf{c}(\mathbf{x}) &\to \mathbf{i}(\mathbf{x}) \\ \mathbf{c}_0 &\to \mathbf{j} \end{aligned}$$
 (3.12)

Dann gilt für das Minimum

$$\mathbf{i}(\mathbf{x}) \qquad \text{zu bestimmen aus}$$
(3.13)  
$$\mathbf{i}(\mathbf{x}_0) = \min_{\mathbf{i}} \check{D} = \frac{1}{2} \sum_{\boldsymbol{\nu}} \pi_a(\boldsymbol{\nu}) \left[ \mathbf{x}_0 - \hat{\mathbf{x}}(\mathbf{i} + \boldsymbol{\nu}) \right]^2$$

wobei wegen der Diskretheit von

$$\mathbf{i} = (i_1, i_2, \cdots, i_{M'})$$
 (3.14)

eine infinitesimale Variation nicht möglich ist. Damit bekommt man das

Verfahren von Kohonen:

# ${\bf I}$ Lernverfahren

i) Numeriere alle Daten  $\mathbf{x}_l$ , l ganze Zahl

16

- ii) Wähle eine Dimension M und lege im ganzzahligen Raum<sup>3</sup>  $I^M$  ein einfach zusammenhängendes Gebiet<sup>4</sup>, bestehend aus  $N_R$  Punkten fest. Ordne jedem  $\mathbf{j}$  der  $N_R$  Punkte einen Vektor  $\hat{\mathbf{x}}(\mathbf{j})$  zu. Diese Vektoren werden zu Repräsentanten der  $\{\mathbf{x}_l\}$ . Sie werden iteriert und ändern sich bei jeder Iteration  $\hat{\mathbf{x}}^n(\mathbf{j}) \to \hat{\mathbf{x}}^{n+1}(\mathbf{j})$ . Als Anfangswert  $\hat{\mathbf{x}}^0(\mathbf{j})$  wird für jedes  $\mathbf{j}$  ein Vektor  $\mathbf{x}_l$  ausgesucht, wobei die l durch einen Zufallszahlengenerator bestimmt werden. Damit hat man dann anfangs  $N_R$  Vektoren  $\hat{\mathbf{x}}^{(0)}(\mathbf{j})$ .
- iii) Wähle mit Zufallszahlengenerator ein  $\mathbf{x}_{l_0}$  aus dem Datensatz aus und bestimme das zugehörige i durch

$$\mathbf{i}(\mathbf{x}_{l_0}) = \min_{\mathbf{i}} \check{D} = \frac{1}{2} \sum_{\boldsymbol{\nu}} \pi_a(\boldsymbol{\nu}) \left[ \mathbf{x}_{l_0} - \hat{\mathbf{x}}(\mathbf{i} + \boldsymbol{\nu}) \right]^2$$
(3.15)

iv) Bestimme die neuen  $\hat{\mathbf{x}}^{(n+1)}(\mathbf{j})$  aus<sup>56</sup>

$$\hat{\mathbf{x}}^{(n+1)}(\mathbf{j}) = \hat{\mathbf{x}}^{(n)}(\mathbf{j}) + \eta \pi_a(\mathbf{j} - \mathbf{i}(\mathbf{x}_{l_0})) \left[\mathbf{x}_{l_0} - \hat{\mathbf{x}}^{(n)}(\mathbf{j})\right], \ \eta \ll 1$$
(3.16)

- v) Fahre fort mit iii) und iv), wobei  $\eta$  und *a* verkleinert werden, bis (stochastische) Konvergenz eingetreten ist.
- **II** Durchführung
  - i) Bestimme zu einem gegebenen  $\mathbf{x}$  das entsprechende  $\mathbf{i}_0$  durch

$$\mathbf{i}_0(\mathbf{x}) = \min_{\mathbf{i}} \left[ \mathbf{x} - \hat{\mathbf{x}}(\mathbf{i}) \right]^2 \tag{3.17}$$

Der gesuchte Repräsentant ist dann  $\hat{\mathbf{x}}(\mathbf{i}_0)$ 

Das Kohonen Verfahren in dieser Form erreicht Datenkompression und Reduktion der Dimension: statt des Datensatzes  $\{\mathbf{x}_l\}$  tauchen nur die Repräsentanten  $\{\hat{\mathbf{x}}(\mathbf{i})\}$  auf. Wie sich die Datenkompression auswirkt, ist schematisch in Fig.3.2 gezeigt. Die Repräsentanten erzielen unabhängig von der Dimension der diskreten **i** Vektoren eine Überdeckung des Datensatzes.

Im kontinuierlichen Fall folgt aus der Stetigkeit der Abbildungen  $\hat{\mathbf{x}}(\mathbf{c})$  und  $\mathbf{c}(\mathbf{x})$  automatisch die Topologieerhaltung im folgenden Sinn:

$$|\hat{\mathbf{x}}(\mathbf{c}) - \hat{\mathbf{x}}(\mathbf{c}_0)| \to 0 \Leftrightarrow |\mathbf{c} - \mathbf{c}_0| \to 0 \text{ falls } \mathbf{c} \in R^{M'}, \text{ und } \mathbf{c}_0 \in R^{M'} M' = M$$
 (3.18)

Ist die Dimension M' von **c** und **c**<sub>0</sub> zu klein gewählt, dann gilt nur noch statt Gl.(3.18)

$$|\mathbf{c} - \mathbf{c}_0| \to 0 \Rightarrow |\hat{\mathbf{x}}(\mathbf{c}) - \hat{\mathbf{x}}(\mathbf{c}_0)| \to 0, \text{ falls } \mathbf{c} \in R^{M'}, \ \mathbf{c}_0 \in R^{M'}, \ M' < M$$
(3.19)

 $<sup>^{3}\</sup>mathrm{In}~I^{M}$  bestehen die Komponenten aller Vektoren aus ganzen Zahlen.

<sup>&</sup>lt;sup>4</sup>Ein einfach zusammenhängendes Gebiet ist bei M = 1 ein Intervall, bei M = 2 z.B. ein Quadrat, bei M = 3 z.B. ein Würfel usw.

<sup>&</sup>lt;sup>5</sup>**j** durchläuft alle Werte der durch  $\pi_a$  definierten Umgebung von  $\mathbf{i}(\mathbf{x}_{l_0})$ .

<sup>&</sup>lt;sup>6</sup>Gl.(3.16) ist nicht ganz so unschuldig, wie sie aussieht, vgl. Anhang B.



Abbildung 3.2: Schematische Anordnung der Repräsentanten auf einem Quadrat, das gleichmäßig mit  $\{\mathbf{x}_l\}$  überdeckt ist. In a) ist  $i(\mathbf{x}_l)$  1 dimensional, in b) ist  $\mathbf{i}(\mathbf{x}_l)$  2 dimensional. In a) sind im *i* Raum 2 der Nachbarn von  $i_0 = 13$ , nämlich 12 und 14, richtig erfaßt. Dagegen sind die weiteren Nachbarn 4 und 20 im eindimensionalen *i* Raum nicht als Nachbarn gesehen. In b) sind alle 4 Nachbarn von  $\mathbf{i}_0 = (2, 4)$ , nämlich (2, 5), (2, 3), (3, 4) und (1, 4), im zweidimensionalen **i** Raum richtig erkannt. Daß auch eine zu niedrig dimensionierte *i* Abbildung eine Überdeckung des Datensatzes liefert, ist typisch.

Die gleichen Zusammenhänge gelten auch für die diskreten Abbildungen, (vgl. Fig.3.2)

 $\hat{\mathbf{x}}(\mathbf{i})$  und  $\hat{\mathbf{x}}(\mathbf{i}_0)$  nächste Nachbarn  $\Leftrightarrow |\mathbf{i} - \mathbf{i}_0| = 1$  falls  $\mathbf{i} \in I^{M'}$ ,  $\mathbf{i}_0 \in I^{M'} M' = M$  (3.20)

Ist die Dimension von i und  $i_0$  zu klein gewählt, dann gilt nur noch statt Gl.(3.20)

 $|\mathbf{i} - \mathbf{i}_0| = 1 \Rightarrow \hat{\mathbf{x}}(\mathbf{i}) \text{ und } \hat{\mathbf{x}}(\mathbf{i}_0) \text{ nächste Nachbarn , falls } \mathbf{i} \in I^{M'}, \ \mathbf{i}_0 \in I^{M'}, \ M' < M$  (3.21)

### Anwendung des Kohonen Formalismus auf Funktionen

Sei eine Funktion auf  $P_D$  definiert

$$\mathbf{f}: P_D \to R^Q \tag{3.22}$$

Z.B kann **f** die Funktion sein, die den Wetterdaten, die in **x** zusammengefaßt sind, (vgl. Gl.(3.1), die 6 Stunden später auftretenden Wasserstände zuordnet. Q ist dann die Anzahl der Orte, an denen die Wasserstände für die nächsten Stunden vorauszusagen sind.

Um das Kohonen Verfahren anzuwenden, wird eine leichte Verallgemeinerungen am Formalismus vorgenommen:

Für die Lernphase wird definiert

$$R^{N+Q} = R^N \oplus R^Q$$

$$\mathbf{z} = \mathbf{x} \oplus \mathbf{f}(\mathbf{x})$$
(3.23)

wobei  $\mathbf{f}(\mathbf{x})$  in der Lernphase die tatsächlichen Werte sind, die sich ergeben haben. Es wird jedoch nicht überall  $\mathbf{x}$  durch  $\mathbf{z}$  ersetzt. Denn das Anordnen der Indexvektoren  $\mathbf{i}$  darf nicht von  $\mathbf{f}(\mathbf{x})$  beeinflußt werden. Damit bekommt man das

Verfahren von Kohonen bei Funktionen:

I Lernverfahren

- i) Numeriere alle Daten  $\mathbf{z}_l$ , l ganze Zahl
- ii) Bestimme  $N_R$  Repräsentanten aus  $\mathbf{z}_l$  durch Zufallszahlen und setze für den ganzzahligen Vektor **j** irgendwelche Werte ein. Damit hat man dann  $N_R$  Vektoren  $\hat{\mathbf{z}}^{(0)}(\mathbf{j})$
- iii) Wähle mit Zufallszahlengenerator ein  $\mathbf{z}_{l_0}$ aus dem Datensatz aus und bestimme $\mathbf{i}$ durch

$$\mathbf{i}(\mathbf{x}_{l_0}) = \min_{\mathbf{i}} \check{D} \\
 mit 
 \tag{3.24}$$

$$\check{D} = \frac{1}{2} \sum_{\boldsymbol{\nu}} \pi_a(\boldsymbol{\nu}) \left[ \mathbf{x}_{l_0} - \hat{\mathbf{x}} (\mathbf{i} + \boldsymbol{\nu}) \right]^2$$
(3.25)

iv) Bestimme das neue  $\hat{\mathbf{z}}^{(n+1)}(\mathbf{j})$  aus

$$\hat{\mathbf{z}}^{(n+1)}(\mathbf{j}) = \hat{\mathbf{z}}^{(n)}(\mathbf{j}) + \eta \pi_a(\mathbf{j} - \mathbf{i}(\mathbf{x}_{l_0})) \left[\mathbf{z}_{l_0} - \hat{\mathbf{z}}^{(n)}(\mathbf{j})\right], \ \eta \ll 1$$
(3.26)

v) Fahre fort mit iii) und iv), wobe<br/>i $\eta$  und a verkleinert werden, bis Konvergenz<br/> eingetreten ist.

## II Durchführung

0) 0. Näherung:

Bestimme zu einem gegebenen  $\mathbf{x}$  das entsprechende  $\mathbf{i}_0$  durch

$$\mathbf{i}_0(\mathbf{x}) = \min_{\mathbf{i}} \left[ \mathbf{x} - \hat{\mathbf{x}}(\mathbf{i}) \right]^2$$
(3.27)

Der gesuchte Repräsentant ist dann  $\hat{\mathbf{x}}(\mathbf{i}_0)$  und der gesuchte Funktionswert  $\mathbf{f}(\hat{\mathbf{x}}(\mathbf{i}_0))$ .

1) 1. Näherung:

Gehe vor wie in 0). Alsdann Bestimme mit Hilfe der nächsten Nachbarn von  $\mathbf{i}_0$ ,  $\mathbf{i}_{\alpha}$ , die approximierten Tangentenvektoren von  $P_D$  in  $\hat{\mathbf{x}}_{\mathbf{i}_0}$ 

$$\mathbf{s}_{\alpha} = \hat{\mathbf{x}}(\mathbf{i}_{\alpha}) - \hat{\mathbf{x}}(\mathbf{i}_{0}) \tag{3.28}$$

Wähle  $\gamma_{\alpha}$  so, daß der folgende Vektor minimiert wird

$$\delta \mathbf{v} = \mathbf{x} - \hat{\mathbf{x}}(\mathbf{i}_0) - \sum \gamma_\alpha \mathbf{s}_\alpha, \ |\gamma_\alpha| \le 1$$
(3.29)

Liegt  $|\delta \mathbf{v}|$  oberhalb der Rauschgrenze, dann ist die Dimension von **i** zu klein eine Interpolation ist nicht möglich, weil die Abbildung  $\mathbf{i}(\mathbf{x})$  nicht topologieerhaltend ist<sup>7</sup>. Andernfalls hat man 2 Ziele erreicht:

<sup>&</sup>lt;sup>7</sup>An dieser Stelle zeigt sich die Wichtigkeit der Topologieerhaltung.

\* lineare Interpolation: Setze

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{x}_{\mathbf{i}_0}) + \sum \gamma_{\alpha} [\mathbf{f}(\mathbf{x}) - \mathbf{f}(\hat{\mathbf{x}}(\mathbf{i}_{\alpha}))]$$
(3.30)

\* Paramterdarstellung von  $P_D$ : Die  $\gamma_{\alpha}$  in Gl.(3.29) erlauben den Übergang vom diskreten Parameterraum  $I^{M'}$  zum kontinuierlichen  $R^{M'}$ . Sei

$$\boldsymbol{\gamma} = (\gamma_1, \gamma_2 \cdots \gamma_{M'}) \tag{3.31}$$

Dann ist in eindeutiger Weise

$$\mathbf{x} = \hat{\mathbf{x}}_{\mathbf{i}_0} + \sum \gamma_{\alpha} \mathbf{s}_{\alpha} = \mathbf{x}(\mathbf{c})$$
  
mit  
$$\mathbf{c} = \hat{\mathbf{x}}_{\mathbf{i}_0} + \boldsymbol{\gamma}$$
 (3.32)

Das Kohonenverfahren benutzt das Gradientenverfahren, allerdings modifiziert durch Einführen der Unschärfefunktion  $\pi_a$ . Dies ist analog zur Einführung von Fluktuationen, die im Verlauf des Iterationsverfahrens abnehmen durch abnehmendes a. Trotzdem kann es vorkommen, daß die Iterationsmethode von Kohonen nicht funktioniert. Das bedeutet aber noch keineswegs, daß der Ansatz von Kohonen falsch ist. Man muß nur modifizierte Verfahren in Richtung von simulated annealing anwenden, um konsistente **i**- und  $\hat{\mathbf{x}}$ Abbildungen zu erhalten.

Wenn die Dimension M' des diskreten Raumes  $I^{M'}$  richtig gewählt ist, bietet dies Verfahren, wie gezeigt, eine Parameterdarstellung von  $P_D$ . Ist M' kleiner als die tatsächliche Dimension M von  $P_D$ , dann besteht das Verfahren von Kohonen im Folgenden. Bekannte Eingaben  $\mathbf{x}$  und die dazu bekannten Ergebnisse  $\mathbf{f}(\mathbf{x})$  werden in einem (N + Q)dimensionalen Raum eingetragen. Da die Anzahl der Datenpunkte zu hoch ist, werden Repräsentanten berechnet. Das ist die Lernphase. In der Ausführungsphase wird zu einem aktuellen Datenpunkt  $\mathbf{x}$  der nächstliegende Repräsentant  $\hat{\mathbf{x}}(\mathbf{j})$  ermittelt, der zugehörige Wert  $\mathbf{f}(\hat{\mathbf{x}}(\mathbf{j}))$  berechnet und dieser Wert als Ergebnis genommen.

### Wasserstandsvorhersage mit Hilfe neuronaler Netze

Das Bundesamt für Seeschiffahrt und Hydrographie in Hamburg hat u.a. die Aufgabe, die Wasserstände an der Deutschen Nordseeküste, insbesondere an den Mündungen von Ems, Jade, Weser und Elbe vorauszusagen, und zwar die Hochwasser- und Niedrigwasserpegel für die nächsten 5 Perioden (1 Periode beträgt etwa 12 Stunden). Gleichzeitig werden die Daten für Pegelstand, Lufttemperatur, Wassertemperatur, Wind<sub>x</sub>, Wind<sub>y</sub>, Luftdruck, Luftfeuchtigkeit usw im Stundentakt gespeichert und liegen von etwa 20 Stationen für die letzten Jahrzehnte vor.

Gleichfalls liegen vor die Voraussagen für die Wasserstände, so daß man die Trefferquote, die mit den bisher oder seinerzeit üblichen Methoden erreicht wurde, bestimmen und mit neuen Methoden vergleichen kann. Das bedeutet, daß man sich bei der Entwicklung eines neuen Modells auf Daten in der Größenordnung  $10^5$  bis  $10^6$  stützen kann.

Wie aus Gl.(3.1) hervorgeht, ist die Dimension N des Datenraumes sehr groß und liegt etwa - je nachdem, welche Daten und Stationen als irrelevant eingeschätzt werden, zwischen 50 und 200.

Aus der Analyse der aktuellen Daten soll eine Voraussage der Wasserstände an der Deutschen Nordseeküste, insbesondere wieder an den Mündungen von Ems, Jade, Weser und Elbe erstellt werden - für die nächsten 5 Perioden, wovon die erste natürlich die wichtigste ist. Die Dimension Q der Funktion **f** ist daher auch groß, sie liegt bei 100.

Von der Dimension der Ausgabenfunktion braucht man sich nicht besonders beeindrucken lassen. Es genügt der Nachweis, daß das Verfahren für die nächsten 5 Perioden an *einem* Ort richtige Voraussagen trifft, um die Verwenbarkeit der Methode nachweisen und mit früheren Voraussagen vergleichen zu können. Das reduziert die Dimension auf Q < 12, wenn man will, sogar auf Q = 1.

Bei der Dimension des Eingabevektors  $\mathbf{x}$  sieht es etwas anders aus. Man kommt schnell auf hohe Dimensionen, die sich noch vervielfachen, wenn man die Werte der vorhergehenden Stunden mit berücksichtigt. Das ergibt dann Dimensionen der Größenordnung 1000, die nicht einfach reduziert werden können.

Welcher Art von neuronalen Netzen der Vorzug gegeben werden sollte, ist nicht so leicht zu entscheiden. Nicht in Frage kommen das McCulloch-Pitts Modell sowie das einfache Perzeptron, da diese zu einfache Strukturen voraussetzen.

Man hat mit feed forward Netzen Voraussagen versucht. Als Nachteil stellte sich die schlechte und z.Teil offenbar nicht kontrollierbare Art von Konvergenz/Nichtkonvergenz der Kostenfunktion  $E_{Kost}$  heraus. Klar ist, daß die meisten Komponenten von  $\mathbf{x}$  sehr stark korreliert sind. Z.B. Luftdruck und Windgeschwindigkeit sind intern korreliert, Windgeschwindigkeiten und Temperatur usw. sind extern (d.h. bei den verschiedenen Stationen) ebenfalls sehr stark korreliert. Die tatsächliche Dimension M des Parametersatzes  $P_D$  wird daher viel niedriger als die Dimension N von  $\mathbf{x}$  sein. Das spricht für den Kohonenansatz. Dieser war in der Tat auch erfolgreich und wird nun beschrieben.

Das Vorgehen sah folgendermaßen aus:

• Der Eingabevektor  $\mathbf{x}$  mußte natürlich dimensionslos gemacht werden. Dabei wurde jede dimensionslose Komponente so eingestellt, daß sie zwischen -1 und 1 lag.

Danach wurde sie mit einem Skalenfaktor multipliziert, der ihr Gewicht, verglichen mit anderen Komponenten, festlegte, da er bei der Berechnung von  $[\mathbf{x} - \hat{\mathbf{x}}]^2$  zum Tragen kommt. Hier gingen die Bedeutung einer Größe, der Fehler, mit der sie erfaßt wird und der zeitliche sowie evtl der räumliche Abstand vom Voraussagepunkt ein.

- Die Anfangspositionen der  $\{\hat{\mathbf{x}}\}$  mußte festgelegt werden. Dies ist nach der Normierung relativ einfach, da die Schwankungsbreite der einzelnen Werte auf 1 gesetzt ist und somit der in Frage kommende Wertebereich abgeschätzt werden kann.
- Festzulegen war gleichzeitig die Dimension M' des diskreten Raumes  $I^{M'}$ . Hier hat man sich auf

$$M' = 2$$
 bei Wasserstandsvorhersage (4.1)

 $festgelegt^1$ .

• Die Lernphase verwendete Daten aus den Jahren 1985 bis 1993. Die Anzahl der Datenpunkte war etwa 10<sup>5</sup>. Benutzt wurde der Kohonen Formalismus für Funktionen.

Die Voraussagen wurden mit Daten aus 1993, die dem Programm in der Lernphase nicht zur Verfügung gestellt worden waren, gebildet und mit anderen Voraussagen verglichen. Dabei ergab sich, daß das auf dem Ansatz von Kohonen beruhende Programm die genauesten Voraussagen machte. Die Fehler liegen bei wenigen cm.

Wie ausdrücklich hervorgehoben werden muß, ist nur im Normalfall mit dieser Methode eine richtige Voraussage zu erwarten. Z.B. katastrophale Wasserstände in Folge von extrem starken Sturmfluten können nicht gut oder gar nicht vorausgesagt werden, weil einfach zu wenig Daten während des Lernprozesses vorliegen und damit die Repräsentanten  $\hat{\mathbf{x}}(\mathbf{j})$  entsprechend entfernt von diesen Datenpunkten liegen<sup>2</sup>.

<sup>&</sup>lt;sup>1</sup>Das wird merkwürdiger Weise nicht weiter begründet als mit der Feststellung, daß die meisten Anwendungen mit Kohonen Netzen der Dimension M' = 2 arbeiten. Immerhin wird auch durch diese Wahl eine repräsentative Überdeckung von  $P_D$  mit den  $\{\hat{\mathbf{x}}\}$  erreicht, aber z.B. eine Interpolation oder Parameterdarstellung ist nicht mehr möglich. Auch eine Auskunft darüber, wieviele relevante Größen überhaupt mitspielen, geht verloren.

<sup>&</sup>lt;sup>2</sup>Man könnte natürlich diese Extrem Daten besonders hervorheben, indem man z.B. um einen kleinen Bereich eines Datenpunktes weitere künstlich generiert, dies wurde jedoch nicht versucht.

# Mögliche Instabilität des Golfstroms

## Einführung

Der Golfstrom ist eine riesige Oberflächenströmung - eine der mächtigsten -, die warmes Wasser vom Golf von Mexiko an die Küsten Europas bringt. Im Golf von Mexiko hat er nur eine Breite von etwa 100 km und eine Geschwindigkeit von 5 km/h bei einer Wassertemperatur von 25C. Im Nordatlantik wird er 10 mal breiter mit einer Strömungsgeschwindigkeit von nur noch 1 km/h. Korrespondierend zur warmen Oberflächenströmung ist eine kalte Grundströmung, die nach Süden zurückfließt,vgl Fig.5.1.



Abbildung 5.1: Strömungsverlauf des Golfstroms (nach Rahmstorf)

Die als Wärme gespeicherte Energie der Oberflächenströmung wird in gewaltigen Mengen transportiert und führt zu einer Leistungsabgabe, die umgerechnet 10<sup>15</sup> Watt beträgt. Die Konsequenz ist ein, im Vergleich zu anderen Gebieten derselben geographischen Breite, sehr mildes Klima West- und Nordeuropas. Der Wegfall des Golfstromes würde also für Europa eine Naturkatastrophe darstellen, die alle anderen Naturkatastrophen, die wir in geschichtlicher Zeit in Europa kennen, in den Schatten stellen würde. Es gibt aus der geschichtlichen Zeit keinerlei Berichte oder Hinweise irgendwelcher Art, die auf eine Instabilität des Golfstromes hindeuten könnten. Insofern wird man sich keine Sorge über die Zukunft des Golfstromes machen wollen - warum sollte sich auf einmal etwas ändern? Zu diesem nicht unplausiblen Argument muß man aber zwei Anmerkungen machen. Die erste bezieht sich auf die vorgeschichtliche Zeit: Wir haben erdrückende Hinweise, daß während der letzten Eiszeit vor 12000 Jahren die nach Norden führende Strömung des Golfstroms nicht existierte. Die zweite bezieht sich auf die durch die Menschheit selbst herbeigeführte Klimaänderung. Diese ist immerhin so groß bzw wird so groß werden, daß eine neue Lage für unser Ökosystem entsteht: Die Bedingungen für stationäres Verhalten beginnen sich zu verschieben.

Man wird zu Recht davon ausgehen, daß eine normale Strömung, etwa die eines Flusses, bei Anderung der Umgebungsparameter sich stetig ändert. Beim Golfstrom handelt es sich aber um keine normale Strömung. Sondern diese kommt dadurch zu Stande, daß zumächst einmal wegen der Corioliskräfte, der Passate aus östlichen Richtungen in den Tropen und der Westwinde in mittleren Breiten, der Küstenlinien und der Gestaltung des Meeresbodens, sowohl eine Oberflächenströmung aus dem Golf von Mexika nach Norden wie auch eine Meeresgrundströmung von Grönland nach Süden begünstigt wird. Die Verbindung wird durch folgendes Phänomen hergestellt, das dann den Golfstrom auslöst: In den Regionen am Golf von Mexiko wird das Oberflächenwasser stark erwärmt, wodurch es sich ausdehnt und gleichzeitig sein Salzgehalt wegen der starken Verdunstung ansteigt. Durch den Anstieg des Salzgehaltes nimmt das spezifische Gewicht zu, durch die thermische Ausdehnung nimmt das spezifische Gewicht ab, netto überwiegt in den tropischen und subtropischen Gewässern die Abnahme: Die Oberflächenströmung mischt sich nicht mit den tieferen Wasserschichten. Das ändert sich vollkommen in den Gewässern des Nordatlantik vor Grönland. Island und weiter nördlich: Jetzt schlägt der höhere Salzgehalt des Oberflächenwassers voll durch, das bisherige Oberflächenwasser sinkt auf den Grund und stellt damit die Verbindung zwischen nördlicher Oberflächenströmung und südlicher Tiefenströmung her. Wegen dieses Effekts wird der Golfstrom als thermohaline Strömung bezeichnet.

Bei einem solchen komplexen Strömungsmechanismus ist durchaus von vornherein nicht klar, ob wirklich eine stetige Änderung der äußeren Parameter nur eine stetige Änderung des Golfstromes nach sich zieht. Hier wird gezeigt, daß eine thermohaline Strömung in der Tat intrinsische Instabilitäten aufweist, daß also eine stetige Änderung der äußeren Parameter eine dramatische Änderung der Strömung zur Folge haben kann. Gezeigt werden wird dies an einem einfachen Modell mit einem Instabilitätsmechanismus, der zu einem Zustand verschwindender Stömungsgeschwindigkeit führen kann. Dieser Zustand bleibt dann seinerseits sehr stabil über einen weiten Parameterbereich, so daß erst bei massiven Änderungen, wie sie etwa bei globalen Katastrophen auftreten, die vorherige Situation wiederhergestellt werden kann. In Analogie zur Festkörperphysik kann man das Auftreten der Instabilität als Phasenübergang 1. Ordnung bezeichnen, begleitet von deutlicher Hysterese.

Nach den heutigen Abschätzungen ist ein solcher Vorgang, was den Golfstrom anbetrifft, zwar sehr unwahrscheinlich, aber man hat eines der Probleme, wie man sie von Kernkraftwerken her kennt: low probability, high risk.

In den nächsten Abschnitten werden die Ursache für die Existenz des Golfstroms sowie

das Vorhandensein einer Instabilität des Golfstroms an dem erwähnten einfachen Modell untersucht werden.

### Modell für den Golfstrom

#### Boxmodell von Stommel, vgl. Fig.6.1

Am Golf von Mexiko ist das Ozeanwasser sehr warm mit Temperatur  $T_1$  und Salinität  $S_1$ . In dieser Umgebung liegt das Reservoir des Golfstroms (Reservoir 1). Da hier Wasser zuund abströmt sind im Strömungsgebiet Temperatur und Salinität nicht  $T_1$  und  $S_1$  sondern verschieden davon. Unter Vernachlässigung aller Temperatur- und Salinitätsprofile nehme ich an, daß im Strömungsgebiet ein einheitlicher Zustand (Zustand 1) besteht, d.h. daß Temperatur und Salinität hier räumlich konstant sind mit Werten  $\tau_1$  und  $\sigma_1$ .

Bekanntlich fließt der Golfstrom, der ein Oberflächenstrom ist, nach Norden. Dort stößt er bei Kanada, Grönland und Skandinavien auf kaltes Wasser. Dies ist ein Kältereservoir (Reservoir 2) mit Temperatur  $T_2$  und Salinität  $S_2$ . Das darin befindliche Strömungsgebiet beschreibe ich wie oben durch einen Zustand (Zustand 2) mit den Größen  $\tau_2$  und  $\sigma_2$ . Das kalte Wasser hat eine größere Dichte, sinkt nach unten und fließt zurück.



Abbildung 6.1: Einfaches Boxmodell von Stommel für den Golfstrom

Wie ändern sich nun Temperaturen und Salinitäten in Zustand 1? Hier gibt es 2 Terme,  $\tau_1$  ändert sich  $\propto (T_1 - \tau_1)$  wegen Einbettung in Reservoir 1 (konvektiver Term). Eine weitere Änderung erfolgt durch die Strömung und ist  $\propto v_S$ , wobei  $v_S$  die Strömungsgeschwindigkeit des warmen/kalten Wassers ist, das her- bzw. wegtransportiert wird. Damit erhält man

$$\frac{d}{dt}\tau_1 = c(T_1 - \tau_1) - g \mid v_s \mid (\tau_1 - \tau_2)$$
(6.1)

Beachte, daß der Strömungsterm nicht vom Vorzeichen der Strömung abhängt. (Nehme ich in Gedanken an, daß der Golfstrom mit gleicher Strömungsstärke, aber nach Süden fließen würde, dann würde immer noch die gleiche Menge kaltes Wasser vom Norden nach Süden fließen - nur jetzt als Oberflächenstrom.) Die Größe  $gv_s$  gibt die relative Änderung q des Reservoirinhalts an. Dies ist eine sehr viel anschaulichere und relevantere Größe. Ich setze daher

$$q = gv_s \tag{6.2}$$

und erhalte dann für Temperatur und entsprechend für Salinität die Gleichungen

$$\frac{d}{dt}\tau_{1} = c(T_{1} - \tau_{1}) - |q|(\tau_{1} - \tau_{2})$$

$$\frac{d}{dt}\sigma_{1} = d(S_{1} - \sigma_{1}) - |q|(\sigma_{1} - \sigma_{2})$$
(6.3)

(Beachte, daß hier das gleiche q auftritt.) Ebenso bekommt man die Gleichungen für den Zustand des anderen Reservoirs:

$$\frac{d}{dt}\tau_{2} = c(T_{2} - \tau_{2}) - |q| (\tau_{2} - \tau_{1})$$

$$\frac{d}{dt}\sigma_{2} = d(S_{2} - \sigma_{2}) - |q| (\sigma_{2} - \sigma_{1})$$
(6.4)

Wann kommt nun eine Strömung selbst zustande? Der springende Punkt ist die Möglichkeit des Oberflächenwassers, nach unten zu sinken, wenn es auf kältere Wasserschichten stößt. Die Ursache dafür liegt in der unterschiedlichen Dichte  $\rho$  in Reservoir 1 und 2: Wasser strömt nach Norden und im Norden nimmt die Dichte ab, wodurch es zum Absinken des Wassers kommen kann. Der dadurch entstehende Strom hängt ab vom Dichteunterschied, in linearer Näherung kann man schreiben

$$q = \frac{1}{k}(\rho_2 - \rho_1) \tag{6.5}$$

In Analogie zum elektrischen Strom kann man k als spezifischen Widerstand ansehen.

Im Norden sinkt das kalte - dichtere- Wasser auf den Grund und hat die Tendenz, alles wenigere dichte Wasser auf dem Grund zu verdrängen. Das unterstützt eine Grundströmung nach Süden.

Die Dichte ändert sich sowohl infolge der Temperatur als auch der Salinität, und zwar bekommt man in linearer Näherung - die bei den kleinen Dichteschwankungen völlig ausreicht:

$$\rho = \rho_0 [1 - \alpha (T - T_0) + \beta (S - S_0)]$$

wobei der Zustand  $(\rho_0, T_0, S_0)$  ein Referenzzustand ist, der frei gewählt werden kann. Somit erhält man

$$\rho_1 = \rho_0 [1 - \alpha(\tau_1 - T_0) + \beta(\sigma_1 - S_0)] 
\rho_2 = \rho_0 [1 - \alpha(\tau_2 - T_0) + \beta(\sigma_2 - S_0)]$$
(6.6)

29

und

$$q = \frac{\rho_0}{k} \left[ \alpha(\tau_1 - \tau_2) - \beta(\sigma_1 - \sigma_2) \right]$$
(6.7)

Mit den Gleichungen Gl.(6.3), Gl.(6.4) und Gl.(6.7) hat man nun ein in sich geschlossenes System von 4 gekoppelten gewöhnlichen nichtlinearen Differentialgleichungen.

### Diskussion der Modell Gleichungen

Der nichtlineare q Term verhindert eine einfache Lösung der Differentialgleichungen. Er tritt jedoch mit entgegengesetztem Vorzeichen auf, d.h. er fällt weg bei Addition der entsprechenden Gleichungen. Das legt eine Substitution

$$X_{T} = \frac{1}{2}(\tau_{1} + \tau_{2})$$

$$Y_{S} = \frac{1}{2}(\sigma_{1} + \sigma_{2})$$
(7.1)

nahe. Ferner treten im nichtlinearen Termnur Differenzen auf, was die Substitution nahelegt

$$x_T = \tau_1 - \tau_2 \tag{7.2}$$
$$y_S = \sigma_1 - \sigma_2$$

Dann erhalte ich die Gleichungen

$$\frac{d}{dt}X_{T} = c \left[\frac{1}{2}(T_{1} + T_{2}) - X_{T}\right] 
\frac{d}{dt}Y_{S} = d \left[\frac{1}{2}(S_{1} + S_{2}) - Y_{S}\right]$$
(7.3)

und

$$\frac{d}{dt}x_{T} = c(T_{1} - T_{2} - x_{T}) - 2 |q| x_{T}$$

$$\frac{d}{dt}y_{S} = d(S_{1} - S_{2} - y_{S}) - 2 |q| y_{S}$$

$$q = \frac{\rho_{0}}{k} [\alpha x_{T} - \beta y_{S}]$$
(7.4)

Vor einer Diskussion dieser Gleichungen führe ich noch dimensionslose Größen und die relevanten Parameter ein. 3 Skalen kann man unabhängig voneinander einstellen: die Zeit, die Temperatur und die Salinität. Die Zeitskala mache ich dimensionslos durch die Skala

$$s = ct \tag{7.5}$$

Für Temperatur und Salinität setze ich jeweils zwei Referenzwerte fest. Die typische Skala, auf der sich  $X_T$  und  $Y_S$  bewegen, ist

$$T_{0} = \frac{1}{2}(T_{1} + T_{2})$$

$$S_{0} = \frac{1}{2}(S_{1} + S_{2})$$
(7.6)

Die typische Skala von  $x_T$  und  $y_S$  sind<sup>1</sup>

$$\Delta_T = T_1 - T_2 \tag{7.7}$$
$$\Delta_S = S_1 - S_2$$

Nun definiere ich die dimensionslosen Größen

$$X = \frac{X_T}{T_0}$$

$$Y = \frac{Y_S}{S_0}$$
(7.8)

und

$$x = \frac{x_T}{\Delta_T}$$

$$y = \frac{y_S}{\Delta_S}$$

$$\kappa = \frac{2q}{c}$$
(7.9)

Kombinationen der bisher eingeführten Parameter ergeben die relevanten Parameter. Diese werden zweckmäßiger Weise definiert durch

$$p_{1} = \frac{ck}{2\rho_{0}\alpha\Delta T}$$

$$p_{2} = \frac{\beta\Delta_{S}}{\alpha\Delta_{T}}$$

$$d_{c} = \frac{d}{c}$$

$$(7.10)$$

Das ergibt die dimensionslosen Gleichungen

.

$$\frac{d}{ds}X = 1 - X$$

$$\frac{d}{ds}Y = d_c(1 - Y)$$
(7.11)

und

$$\frac{d}{ds}x = 1 - x - |\kappa| x$$

$$\frac{d}{ds}y = d_c(1 - y) - |\kappa| y$$

$$\kappa = \frac{1}{p_1} [x - p_2 y]$$
(7.12)

<sup>1</sup>Ich nehme an  $T_1 > T_2, S_1 > S_2$ . Diese Annahme entspricht den Gegebenheiten beim Golfstrom.

Die Lösungen von Gl.(7.11) lassen sich sofort angeben:

$$X(s) = 1 + [X(0) - 1]e^{-s}$$
  

$$Y(s) = 1 + [Y(0) - 1]e^{-d_cs}$$
(7.13)

Zu lösen bleibt demnach das System Gl.(7.12), aber das besteht nur noch aus den 2 gekoppelten Größen (x, y), die von 3 Parametern abhängen,  $p_1$ ,  $p_2$  und  $d_c$ .

### Lösung der Modell Gleichungen

Interessant sind nicht die allgemeinen Lösungen der Modellgleichungen sondern gemäß der Fragestellungen *stationäre Lösungen* und deren Stabilität. Infolgedessen sind die Fixpunkte der Modellgleichungen zu bestimmen. Die Fixpunkte von Gl.(7.11) sind

$$\begin{aligned} X_F &= 1\\ Y_F &= 1 \end{aligned} \tag{8.1}$$

Die Fixpunkte von Gl.(7.12) erhält man aus den algebraischen Gleichungen

$$\begin{array}{rcl}
0 &=& 1 - x_F - \mid \kappa \mid x_F \\
0 &=& d_c (1 - y_F) - \mid \kappa \mid y_F \\
\kappa &=& \frac{1}{p_1} \left[ x_F - p_2 y_F \right]
\end{array}$$
(8.2)

Daraus erhalte ich

$$x_{F} = \frac{1}{1+|\kappa|}$$

$$y_{F} = \frac{1}{1+|\kappa|/d_{c}}$$

$$p_{1}\kappa = \varphi(\kappa, p_{2}, d_{c}) = \left[\frac{1}{1+|\kappa|} - \frac{p_{2}}{1+|\kappa|/d_{c}}\right]$$
(8.3)

In dieser Schreibweise sind die Gleichungen nicht mehr gekoppelt, und es genügt, die (kubische) Gleichung für  $\kappa$  zu bestimmen. Es gibt 3 Lösungen (eine reelle und zwei komplexe oder reelle). Es gibt also entweder 1 oder 3 reelle Fixpunkte. Diese hängen ab von nur noch 3 Parametern:  $p_1$ ,  $p_2$  und  $d_c$ . Die Lösungen kann man aus der kubischen Gleichung exakt bestimmen. Anschaulicher ist jedoch ein graphisches Verfahren, indem man den Schnittpunkt zweier Kurven berechnet, nämlich

$$\begin{aligned}
K_1 : \kappa &\to p_1 \kappa \\
K_2 : \kappa &\to \varphi(\kappa, p_2, d_c)
\end{aligned}$$
(8.4)

#### Stabilitätsanalyse stationärer Lösungen

Hat man die stationären Lösungen, d.h. die Fixpunkte gefunden, dann ist die Frage, ob sie stabil sind, d.h. ob das System bei einer kleinen Störung zum jeweiligen Fixpunkt zurückkehrt - oder nicht. Bei einer kleinen Störung $\delta \mathbf{x}$ kann man entwickeln. Sei

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}) \text{ DGL mit Fixpunkt } \mathbf{x}_F$$
(8.5)

Dann ist

$$\frac{d}{dt}[\mathbf{x}_F + \delta \mathbf{x}] = \frac{d}{dt}\delta \mathbf{x} = \mathbf{f}(\mathbf{x}_F + \delta \mathbf{x}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_F}\delta \mathbf{x} + \mathcal{O}(\delta \mathbf{x}^2)$$

d.h. bei Beschränkung auf den linearen Term

$$\frac{d}{dt}\delta \mathbf{x} = \mathcal{A}\,\delta \mathbf{x}\,\,\mathrm{mit}\,\,\mathcal{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_F} \tag{8.6}$$

Der Fixpunkt  $\mathbf{x}_F$  ist stabil, wenn  $\delta \mathbf{x} \to 0$  für  $t \to \infty$ . Wann das genau der Fall ist, wird in Anhang C diskutiert, vgl. Gl.(C.16).

34

#### Instabilitäten des Modells - Folgerungen für den realen Golfstrom

Das Modell läßt außer den - vom geophysikalischen Gesichtspunkt aus betrachteten - vernünftigen Lösungen auch solche mit negativem  $\kappa$  zu. Darum sind Modifizierungen nötig, da ein verkehrt herum, also nach Süden fließender Golfstrom auszuschließen ist. Diese werden unten diskutiert werden.

Es gibt 3 deutlich verschiedene Lösungsbereiche:

1)

$$d_c \approx 1$$

In diesem Fall führt nur  $p_2 < 1$  zu einer positiven Lösung  $\kappa_1$ . Diese ist eindeutig



Abbildung 9.1: Kurven  $p_1\kappa$  und  $\varphi(\kappa, p_2, d_c)$  für  $p_1 = 0.8$ ,  $p_2 = 0.5 d_c = 1$ . Es gibt nur einen Schnittpunkt der Kurven. Dieser kennzeichnet einen stationären, stabilen, stetig von den Parametern abhängigen Zustand,  $\kappa_1$ 

und stabil und hängt stetig von den Parametern  $p_1$ ,  $p_2$  und  $d_c$  ab. Es passiert also nichts Dramatisches, wenn diese Parameter variieren, vgl. Fig.9.1

2)

 $d_c \gg 1$ 

In diesem Fall führt ebenfalls nur  $p_2 < 1$  zu einer positiven  $\kappa$  Lösung. Diese ist wiederum eindeutig und stabil und hängt stetig von den Parametern  $p_1$ ,  $p_2$  und  $d_c$  ab. Es passiert also nichts Dramatisches, wenn diese Parameter variieren. Bemerkenswert ist, daß  $\kappa$  endlich bleibt bei  $p_1 \rightarrow 0$ , vgl. Fig.9.2.

3)

 $d_c \ll 1$ 



Abbildung 9.2: Kurven  $p_1\kappa$  und  $\varphi(\kappa, p_2, d_c)$  für  $p_1 = 0.8$ ,  $p_2 = 0.5 d_c = 10$ . Es gibt wieder nur einen Schnittpunkt der Kurven. Dieser kennzeichnet einen stationären, stabilen, stetig von den Parametern abhängigen Zustand,  $\kappa_1$ 

Dies ist der Fall mit realistischen Parametern. Betrachte zuerst  $p_2 = 1$ . Dann findet man

$$\kappa_0 = 0$$
 einziger und stabiler Fixpunkt bei  $p_1 > p_{1,0} = \frac{1}{d_c} - 1$ 

Bei  $p_1 < p_{1,0}$  gibt es nun zwei *stabile* Lösungen:  $\kappa_0 = 0$  und  $\kappa_1 > 0$ , vgl. Fig.9.3. Auf den Golfstrom übertragen heißt das, daß sowohl der stromlose Zustand als auch der stromtragende Zustand beide möglich und stabil gegen Störungen sind. Welcher von beiden angenommen wird, hängt von der Vorgeschichte ab. Außerdem nimmt für  $p_1 \rightarrow p_{1,0}$ ,  $p_1 < p_{1,0} \kappa_1$  dramatisch schnell ab und fällt bei  $p_1 = p_{1,0}$  mit  $\kappa_0 = 0$ zusammen.

Noch dramatischer wird die Situation bei  $p_2 > 1$ . Allerdings wird dann eine Modifikation des Gleichungssystems nötig, da sonst der Fixpunkt bei negativen Strömungen eine Rolle spielt. Um im Modell den Bezug zum Golfstrom zu erhalten, bei dem eine negative Strömung auszuschließen ist, werden die Gleichungen modifiziert durch

ersetze in Gl.(7.4): 
$$|q|x_T \rightarrow q\Theta(q)x_T$$
  
ersetze in Gl.(7.12):  $|\kappa|x \rightarrow \kappa\Theta(\kappa)x$  (9.1)  
ersetze in Gl.(7.4):  $|q|y_T \rightarrow q\Theta(q)y_T$   
ersetze in Gl.(7.12):  $|\kappa|y \rightarrow \kappa\Theta(\kappa)y$ 

Die Modifikationen Gl.(9.1) verlagern den negativen stabilen Fixpunkt  $\kappa_0 < 0$  zum stabilen  $\kappa_0 = 0$ . Das Ergebnis ist dann, vgl. Fig.9.4: Es gibt drei Fixpunkte,  $\kappa_0 = 0$ ,  $\kappa_1 > 0$  und  $\kappa_2 > \kappa_1$ .  $\kappa_1$  ist instabil und gibt die Grenze an, bis zu welchem Wert



Abbildung 9.3: Kurven  $p_1\kappa$  und  $\varphi(\kappa, p_2, d_c)$  für  $p_1 = 0.8$ ,  $p_2 = 1$   $d_c = 0.17$ . Es gibt zwei Schnittpunkte der Kurven. Diese kennzeichnen stationäre, stabile, stetig von den Parametern abhängige Zustände.  $\kappa_0 = 0$  ist stromlos,  $\kappa_2$  trägt Strom, solange  $p_1$  nicht zu groß ist.

eine anfängliche Störung  $\kappa(t_0)$  von  $\kappa_2$  abweichen darf. Es gilt:

$$\kappa(t_0) > \kappa_1, \qquad \lim_{t \to \infty} \kappa(t) = \kappa_2 \text{ bei } d_c \gg 1, \ p_2 > 1 \tag{9.2}$$
$$\kappa(t_0) < \kappa_1, \qquad \lim_{t \to \infty} \kappa(t) = \kappa_0 = 0 \text{ bei } d_c \gg 1, \ p_2 > 1$$

Ist nun  $p_1$  so groß, daß  $\kappa_1$  und  $\kappa_2$  dicht beieinander liegen, dann genügt eine geringe Störung, um die Bedingung  $\kappa(t_0) < \kappa_1$  zu erreichen. Daraufhin nimmt im Verlauf der Zeit  $\kappa(t)$  immer weiter ab, d.h. die Strömung wird immer schwächer und verschwindet schließlich. Das gleiche Phänomen tritt auf, wenn infolge Klimaänderung die Parameter sich so verändern , daß die Fixpunkte  $\kappa_1$  und  $\kappa_2$  nicht mehr existieren, vgl. Fig.9.5, denn der stabile Fixpunkt  $\kappa_0 = 0$  existiert nach wie vor.

Das Modell macht eine weitere Aussage:

Kommt man, nachdem der Golfstrom versiegt ist, auf die löbliche Idee, mit der Natur etwas pfleglicher umzugehen, derart, daß  $\kappa_1$  und  $\kappa_2$  wieder existieren und nicht nahezu zusammenfallen, dann hilft das in Bezug auf den Golfstrom gar nichts mehr. Denn um den Strom bei  $\kappa_2$  zu stabilisieren, benötigt man einen Anfangsstrom  $\kappa(t_0) > \kappa_1$ . Eine derartige Änderung kann jedoch nur infolge von Naturkatastrophen biblischen Ausmaßes erreicht werden. D.h. ist der Strom einmal instabil geworden, dann hat man einen praktisch irreversiblen Vorgang eingeleitet, der zum Versiegen des Stromes führt.



Abbildung 9.4: Kurven  $p_1\kappa$  und  $\varphi(\kappa, p_2, d_c)$  für  $p_1 = 0.7$ ,  $p_2 = 1.5$ ,  $d_c = 0.17$ . Es gibt zwei Schnittpunkte der Kurven. Der erste kennzeichnet einen instabilen stationären, der zweite einen stabilen, stetig von den Parametern abhängigen Zustand.  $\kappa_0 = 0$  ist ein weiterer stromloser stabiler Zustand, siehe Text.  $\kappa_2$  trägt immer Strom, kann aber abrupt instationär werden, sobald  $p_1$  zu groß wird.



Abbildung 9.5: Kurven  $p_1\kappa$  und  $\varphi(\kappa, p_2, d_c)$  für  $p_1 = 0.8$ ,  $p_2 = 1.5$ ,  $d_c = 0.17$ . Es gibt nur noch den stabilen stationären Zustand bei  $\kappa_0 = 0$ , siehe Text.  $\kappa_2$  ist gerade instabil geworden und das System • bewegt sich in Richtung  $\kappa_0$ .

### Hinweise zu den numerischen Beispielen in C<sup>++</sup>

Übungsbeispiele sind in den Verzeichnissen Vws03/prog vorhanden. Im Unterverzeichnis ueb01 treten sehr einfache C<sup>++</sup> Programme auf. In ueb02 kommen zum ersten Mal Klassen vor, die sich aber noch nicht von den Strukturen struct in C unterscheiden. In ueb03 treten echte Klassen auf, in ueb04 auch templates.

Beispielrechnungen zum Golfstrom sind von Herrn Ralf Gehrke im Unterverzeichnis gehrke und mir. Das Programm von Herrn Gehrke ist kurz und bündig.

Zu feed-forward Netzen hat Herr Matias Zilly ein Programm geschrieben, das ebenfalls sehr gut verständlich ist und die Vorteile von  $C^{++}$  zeigt. Es steht im Unterverzeichnis zilly. Dort läßt sich auch gut studieren, wie ein Makefile aufgebaut ist.

Zum Golfstromproblem habe ich ein Programm geschrieben, das auf dem von Herrn Gehrke aufbaut, aber zusätzlich die Stabilität der Fixpunkte untersucht (Untervezeichnis lfeld). Hier sind die headerfiles im Verzeichnis include und die Quellunterprogramme im Verzeichnis c0 zusammengefaßt. Hauptprogramm und Datensätze stehen im Verzeichnis c1. Um die Übersicht nicht zu verlieren, wird extra ein Programmverzeichnis Run mit CRun erzeugt (geschrieben für tc Shell, einfach zu ändern für andere Shells). Durch softlinks erscheinen alle Programme und header files in Run, ebenso der aktuelle Datensatz und Makefile. Am Makefile kann man wiederum den typischen Aufbau erkennen.

Wer alle die genannten Programme durchgearbeitet und selbst kompiliert hat, ist über das Anfängerstadium von  $C^{++}$  hinaus.

Fragen an: h.lustfeld@fz-juelich.de

### Hinweise zur Literatur

Einiges aus der Literatur, was ich für die Vorlesung benutzt habe, steht im Verzeichnis Vws03/LitV. Bei den Artikeln habe ich immer nur die erste Seite aufgenommen. Sie können jedoch die vollständigen Artikel per e-mail anfordern (h.lustfeld@fz-juelich.de), wenn Sie Hörer dieser Vorlesung sind oder gewesen sind. In dem Fall muß Ihre e-mail folgenden Satz enthalten: Ich versichere, daß ich die erhaltene Literatur persönlich verwenden und nicht an Dritte in irgendeiner Form, die das copyright verletzt, weitergeben werde.

### Appendix A: Beweis des universalen Approximationstheorems, Gl.(2.2)

Beweis in 3 Schritten:

I) Fourierdarstellung

Sei eine stetige Funktion F in einem N dimensionalen endlichen Bereich B gegeben. Dann kann man diese Funktion stetig fortsetzen in einem Würfel des  $\mathbb{R}^N$ , der B enthält und selbst endlich ist. Es gibt daher zu jedem  $\epsilon > 0$  eine N dimensionale Fourierdarstellung mit

$$H(\mathbf{r}) = \sum_{\mu=1}^{M_1} A(\mathbf{k}_{\mu}) e^{i\mathbf{k}_{\mu}\mathbf{r}}, \ M_1 < \infty$$
  
und  
$$|F(\mathbf{r}) - H(\mathbf{r})| < \epsilon$$
 (A.1)

DaFreell ist, kann auch Hreell gewählt werden, man kann deshalb statt $\operatorname{Gl.}(\operatorname{A.1})$  auch schreiben

$$H(\mathbf{r}) = \sum_{\mu=1}^{M_1} Re\{A(\mathbf{k}_{\mu})e^{i\mathbf{k}_{\mu}\mathbf{r}}\}, \ M_1 < \infty$$

 $\operatorname{oder}$ 

$$H(\mathbf{r}) = \sum_{\nu=1}^{M_2} A(\mathbf{k}_{\nu}) \cos(\mathbf{k}_{\nu}\mathbf{r} - \theta_{\nu}), \ M_2 < \infty$$
  
und  
$$F(\mathbf{r}) - H(\mathbf{r})| < \epsilon$$
 (A.2)

II) Beweis des universalen Approximationstheorems für die speziellen sigmoide Funktionen  $\varphi_{\cos}$ Aus den cos Funktionen werden sigmoide Funktionen  $\varphi_{\cos}$  konstruiert in folgender Weise:

$$\varphi_{\cos}(s) = \begin{cases} 0 \text{ für } -\infty \le s \le -\pi \\ \frac{1}{2} [1 + \cos(s)] \text{ für } -\pi \le s \le 0 \\ 1 \text{ für } s > 0 \end{cases}$$
(A.3)

Hiermit kann man wiederum den cos zurückgewinnen: Aus der Relation

$$\varphi_{\cos}(s) + \varphi_{\cos}(-s) - 1 = \begin{cases} 0 \text{ für } -\infty \le s \le -\pi \\ \frac{1}{2}[1 + \cos(s)] \text{ für } -\pi \le s \le \pi \\ 0 \text{ für } s > 0 \end{cases}$$
(A.4)

folgt

$$\cos(s) = -1 + 2\sum_{\nu=n}^{m} (\varphi_{\cos}(s+\nu 2\pi) + \varphi_{\cos}(-s-\nu 2\pi) - 1)$$
  
für  
$$s \quad \epsilon \quad [(2n-1)\pi, (2m+1)\pi]$$
(A.5)

Da in Gl.(A.2) **r** im endlichen Gebiet *B* des  $R^N$  bleibt, bleibt auch das Argument jedes cos in einem endlichen Intervall. Da die Anzahl der Summanden endlich ist, gibt es ein endliches Intervall, charakterisiert durch  $n_{min}$  und  $m_{max}$ , in dem *alle* cos Argumente bleiben. Mit diesem  $n_{min}$  und  $m_{max}$  kann man mit Gl.(A.5) in Gl.(A.2) die cos Funktionen durch sigmoide ersetzen. Damit ist das universale Approximationstheorem, Gl.(2.2) für { $\varphi_{cos}$ } bewiesen.

III) Darstellung der  $\varphi_{cos}$  Funktionen durch beliebige sigmoide Funktionen. Es sei  $\varphi_b$  eine beliebige sigmoide Funktion. Dann gibt es ein  $P_b$  mit

$$\begin{aligned} 1 - \varphi_b &< \eta \text{ für } s > P_b \\ \varphi_b &< \eta \text{ für } s < P_b \end{aligned} \tag{A.6}$$

Nun gebe ich eine natürliche Zahl G vor und ich unterteile das Intervall  $I = [-\pi, \pi]$ in G gleiche Teile und führe die folgenden Größen und linearen Abbildungen ein:

$$s_0 = -\pi, \ s_1 = -\pi + \frac{2\pi}{G}, \ s_2 = -\pi + 2\frac{2\pi}{G}, \ \dots s_G = \pi$$
 (A.7)

$$\beta_j = p_{\cos}(s_j) - p_{\cos}(s_{j-1}), \ |\beta_j| \le \frac{2\pi}{G}$$
 (A.8)

lineare Abbildung 
$$A_j(s)$$
 mit (A.9)  
 $A_j(s_{j-1}) = -P_b$   
 $A_j(s_j) = P_b$ 

Dann gilt für beliebige simoide Funktionen  $\varphi_b$ :

$$\varphi_{\cos}(s) = \sum_{\nu=1}^{G} \beta_{\nu} \varphi_{b}(A_{j}(s)) + F_{e}$$
  
mit  $|F_{e}| < \frac{2\pi}{G} + \frac{2\pi}{G}G\eta$ 

Setze

$$\eta = \frac{3}{G}, \text{ dann ist } |F_e| < \frac{8\pi}{G}$$
 (A.10)

G darf beliebig groß gewählt werden. Dies bedeutet, daß die Darstellung Gl.(A.10) für  $\varphi_{cos}$  mit beliebiger Genauigkeit eingesetzt werden kann. Da jedoch das universale Approximationstheorem für  $\varphi_{cos}$  schon bewiesen ist, ist damit die Gültigkeit des universalen Approximationstheorems für alle sigmoiden Funktionen bewiesen.

### Appendix B: Herleitung von Gl.(3.16) aus Gl.(3.9)

Beim Übergang vom Integral zur Summe in Gl.(3.9) erhält man

$$\hat{\mathbf{x}}^{(n+1)}(\mathbf{j}) = \hat{\mathbf{x}}^{(n)}(\mathbf{j}) + \eta \sum_{\mathbf{x}_{l_0}} \pi_a(\mathbf{j} - \mathbf{i}(\mathbf{x}_{l_0})) \left[\mathbf{x}_{l_0} - \hat{\mathbf{x}}^{(n)}(\mathbf{j})\right], \ \eta \ll 1$$
(B.1)

Die Unhandlichkeit dieser Formel besteht in der Notwendigkeit, für jedes **j** die Summe über den gesamten Datensatz wirklich durchführen zu müssen, da man nicht weiß, welche  $\mathbf{i}(\mathbf{x}_l)$  in der Umgebung von **j** liegen. Äquivalent dazu aber viel weniger aufwendig ist das folgende Verfahren:

i) Setze

$$\Delta \hat{\mathbf{x}}^{(n)}(\mathbf{j}) = 0$$
 für alle  $N_r$  Werte von  $\mathbf{j}$ 

ii) Berechne für jedes  $\mathbf{x}_{l_0}$ :

 $- \mathbf{i}(\mathbf{x}_{l_0})$  aus Gl.(3.17)

- setze bei allen **j** in der Umgebung von  $\mathbf{i}(\mathbf{x}_{l_0})$ 

$$\Delta \hat{\mathbf{x}}^{(n)}(\mathbf{j}) \to \Delta \hat{\mathbf{x}}^{(n)}(\mathbf{j}) + \eta \pi_a(\mathbf{j} - \mathbf{i}(\mathbf{x}_{l_0})) \left[\mathbf{x}_{l_0} - \hat{\mathbf{x}}^{(n)}(\mathbf{j})\right], \ \eta \ll 1$$
(B.2)

iii) Setze für alle  $N_R$  j Werte:

$$\hat{\mathbf{x}}^{(n+1)}(\mathbf{j}) = \hat{\mathbf{x}}^{(n)}(\mathbf{j}) + \Delta \hat{\mathbf{x}}^{(n)}(\mathbf{j})$$
(B.3)

Der Vorteil dieses Verfahrens liegt darin, daß die  $\{\hat{\mathbf{x}}\}$  wirklich konvergieren und jedes  $\mathbf{x}_l$  wirklich aufgerufen wird. Der Nachteil besteht darin, daß a) eine stochastische Wahl der  $\mathbf{x}_l$  Daten nicht möglich ist, b) die Korrekturen jeweils erst nach Durchlauf des gesamten Datensatzes eingesetzt werden. Diese Nachteile sind in Gl.(3.16) nicht gegeben und beide Verfahren unterscheiden sich nur um  $\mathcal{O}(\eta^2)$ . Darum wählt man das hier beschriebene Verfahren im allgemeinen nicht.

### Appendix C: Eigenschaften autonomer<sup>1</sup>, linearer gewöhnlicher Differentialgleichungen

Autonome lineare homogene N dimensionale gewöhnliche Differentialgleichungen haben die Form

$$\dot{\mathbf{x}} = \mathcal{A}\mathbf{x}$$
(C.1)  
oder ausgeschrieben  
$$\dot{x}_i = \sum_{k=1}^N A_{ik} x_k$$
$$\mathbf{x}(t) \qquad \text{ist } N \text{ dimensionaler Vektor (die Lösungstrajektorie)}$$
$$\mathcal{A} \qquad \text{ist } N \times N \text{ konstante Matrix}$$

Die Lösung dieser DGL im 1 dimensionalen Fall ist

$$x(t) = e^{At} \cdot x(0) \tag{C.2}$$

oder als Potenzreihe

$$x(t) = \sum_{\nu=0}^{\infty} \frac{A^{\nu} t^{\nu}}{\nu!} x(0)$$
 (C.3)

Da im mehrdimensionalen die Matrix  $\mathcal{A}$  mit sich selbst vertauscht, kann man häufig mit ihr wie mit Zahlen rechnen und erhält als Lösung von Gl.(C.1)

$$\mathbf{x}(t) = \sum_{\nu=0}^{\infty} \frac{\mathcal{A}^{\nu} t^{\nu}}{\nu!} \mathbf{x}(0)$$
(C.4)

Die direkte numerische Auswertung dieser Lösung ist wegen der erforderlichen Multiplikation von Matrizen extrem aufwendig. Das ist ein Grund dafür, daß man die Lösung dieser DGL durch Eigenvektoren  $\mathbf{b}^{(i)}$  konstruiert:

$$\mathbf{A}\mathbf{b} = \lambda \mathbf{b} \tag{C.5}$$

oder ausgeschrieben

$$\sum_{k=1}^{N} A_{ik} b_k = \lambda b_i \tag{C.6}$$

Diese Gleichung läßt sich auch schreiben als

$$[\mathcal{A} - \lambda \mathcal{E}]\mathbf{b} = 0, \ \mathcal{E} \text{ ist Einheitsmatrix} \tag{C.7}$$

Gl.(C.7) hat genau dann eine nichttriviale Lösung, wenn det $[\mathcal{A} - \lambda \mathcal{E}] = 0$  ist. det $[\mathcal{A} - \lambda \mathcal{E}]$ ist ein Polynom N. Grades, das sogenannte charakteristische Polynom, es gibt N Lösungen für  $\lambda$ ,  $\lambda^{(1)}...\lambda^{(N)}$ . Ich nehme im Folgenden an, daß diese alle verschieden sind<sup>2</sup>. Dann ist klar, daß es N Eigenvektoren  $\mathbf{b}^{(i)}$  gibt. Diese Eigenvektoren sind linear unabhängig. Denn

<sup>&</sup>lt;sup>1</sup>autonom DGL heißt, daß die DGL keine explizite Zeitabhängigkeit aufweist.

<sup>&</sup>lt;sup>2</sup>Wenn das nicht der Fall ist, dann kann man durch beliebig kleine Änderung von A erreichen, daß kein  $\lambda^{(i)}$  mit einem anderen übereinstimmt.

angenommen, M < N von ihnen wären linear abhängig, aber M - 1 nicht. Dann gäbe es eine Linearkombination mit

$$\mathbf{u} = \sum_{k=1}^{M} \alpha_k \mathbf{b}^{(k)} = 0 \tag{C.8}$$

Außerdem folgte daraus

$$\mathcal{A}\mathbf{u} = 0 = \sum_{k=1}^{M} \lambda^{(k)} \alpha_k \mathbf{b}^{(k)}$$
(C.9)

Da alle  $\lambda^{(k)}$  verschieden sind, ergibt der Vergleich zwischen Gl.(C.8) und Gl.(C.9), daß schon M - 1 Eigenvektoren linear abhängig sein müßten. Aus diesem Widerspruch folgt die lineare Unabhängigkeit. Die  $\mathbf{b}^{(i)}$  bilden demnach eine *Basis*.

Eine spezielle Lösung von Gl.(C.1) ist

$$\mathbf{x} = e^{\lambda^{(k)} t} \mathbf{b}^{(k)} \text{ für alle } k \tag{C.10}$$

Die allgemeine Lösung ist eine Linearkombination dieser Lösungen

$$\mathbf{x} = \sum_{k=1}^{N} a_k e^{\lambda^{(k)} t} \mathbf{b}^{(k)} \tag{C.11}$$

mit irgendwelchen Koeffizienten  $a_k$ .

Die  $\lambda^{(k)}$  und damit die  $\mathbf{b}^{(k)}$  können durchaus auch dann *komplex* sein, wenn die Matrix  $\mathcal{A}$  reell ist. Beispiel:

$$\mathcal{A} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \tag{C.12}$$

hat die Eigenwerte

$$\lambda^{(1,2)} = 1 \pm i \tag{C.13}$$

und die Eigenvektoren

$$\mathbf{b}^{(1,2)} = \begin{pmatrix} 1\\ \pm i \end{pmatrix} \tag{C.14}$$

Dies bedeutet, daß man von vornherein mit komplexen Größen zu rechnen hat z.B. mit dem Skalarprodukt komplexer Vektoren.

### Verhalten von $\mathbf{x}(t)$ :

a)  $t \to \infty$ :

$$\lim_{t \to \infty} | \mathbf{x}(t) | \propto e^{\lambda_{max}t}, \ \lambda_{max} = \max_{\{i\}} \{ \Re(\lambda^{(i)}) \}$$
(C.15)

Das asymptotische Verhalten, hängt nicht vom Anfangswert  $\mathbf{x}(0)$  ab<sup>3</sup>. Demzufolge ist

$$\lim_{t \to \infty} |\mathbf{x}(t)| = 0, \qquad \text{wenn } \Re(\lambda^{(i)}) < 0 \text{ für alle } i.$$

$$\min_{\{\lambda^{(i)}\}} \qquad \text{sind Eigenwerte von } \mathcal{A}$$
(C.16)

<sup>&</sup>lt;sup>3</sup>Die Wahrscheinlichket ist Null, daß eine Komponente  $a_k$  bei zufälliger Wahl von  $\mathbf{x}(0)$  verschwindet.

Gl.(C.16) ist das Stabilitätskriterium für Fixpunkte einer gewöhnlichen DGL, da diese um die Fixpunkte linear entwickelt werden kann.

b) endliche t:

Das asymptotische Verhalten sagt nichts darüber aus, was für endliche Zeiten t passiert. Tatsächlich kann dies zu einem Problem werden. Zwei Aussagen sind hilfreich:

Ist 
$$A_{ki} = A_{ik}^*$$
 (Hermitezität), (C.17)  
dann ist  $|\mathbf{x}(t)| < |\mathbf{x}(0)| \cdot e^{\lambda_{max}t}$ 

Diese Aussage ergibt sich daraus, daß bei Hermitezität alle Eigenvektoren zu verschiedenen  $\lambda^{(i)}$  orthogonal zueinander sind<sup>4</sup>.

Ist 
$$|\mathbf{x}(0)| = 1$$
, aber ist  $A_{ki} \neq A_{ik}^*$ , (C.18)

dann gibt es a priori keine obere Schranke für  $|\mathbf{x}(t)|$ .

Beispiel

$$\mathcal{A} = \begin{pmatrix} -3\epsilon/2 & 1/2\\ \epsilon^2/2 & -3\epsilon/2 \end{pmatrix}, \ \epsilon > 0$$
(C.19)

Eigenwerte:

$$\lambda^{(1)} = -\epsilon, \ \lambda^{(2)} = -2\epsilon \tag{C.20}$$

Eigenvektoren:

$$\mathbf{b}^{(1)} = \begin{pmatrix} 1 \\ \epsilon \end{pmatrix}, \ \mathbf{b}^{(2)} = \begin{pmatrix} 1 \\ -\epsilon \end{pmatrix}$$
(C.21)

Als Anfangswert setze ich

$$\mathbf{x}(0) = \begin{pmatrix} 0\\1 \end{pmatrix} \tag{C.22}$$

dann ist die Lösung der DGL mit  $\mathcal{A}$  aus Gl.(C.19)

$$\mathbf{x}(t) = \frac{e^{-\epsilon t}}{2\epsilon} \mathbf{b}^{(1)} - \frac{e^{-2\epsilon t}}{2\epsilon} \mathbf{b}^{(2)}$$
(C.23)

Wie man sieht, dreht **x** in  $\begin{pmatrix} 1 \\ \epsilon \end{pmatrix}$ -Richtung und

$$|\mathbf{x}(t)| \to \frac{e^{-t}}{2\epsilon}$$

aber

$$|\mathbf{x}(t=1/\epsilon)| \approx \frac{1}{2e\epsilon}$$
 (C.24)

 $\epsilon$  darf beliebig klein sein. Demnach läßt sich ohne genauere Kenntnis der jeweiligen Matrix  $\mathcal{A}$  keine obere Schranke für  $|\mathbf{x}(t)|$  angeben<sup>5</sup>.

<sup>&</sup>lt;sup>4</sup>Bei Hermitezität sind die Eigenwerte außerdem reell.

<sup>&</sup>lt;sup>5</sup>Dies kann zu unangenehmer Konsequenz bei Linearisierung einer DGL um den Fixpunkt führen, nämlich der, daß der Gültigkeitsbereich der Linearisierung sehr klein wird.