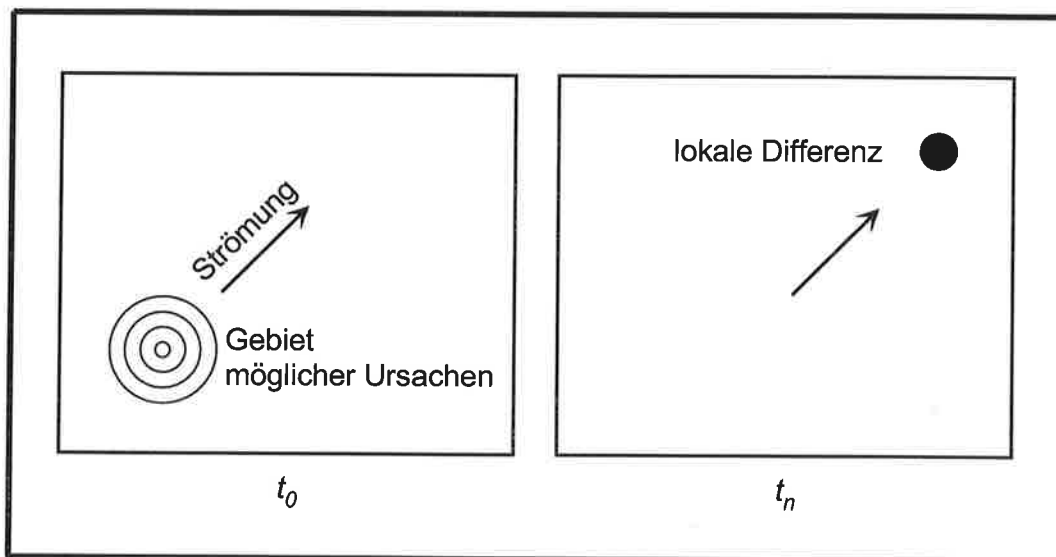




# Max-Planck-Institut für Meteorologie

## EXAMENSARBEIT Nr. 44



ERSTELLUNG EINES ADJUNGIERTEN MODELLS  
ZUR ASSIMILIERUNG VON DATEN IN EIN MODELL  
DER GLOBALEN OZEANISCHEN ZIRKULATION

VON  
Ralf Giering

HAMBURG, November 1996

Dissertation im Rahmen des Promotionsverfahrens

Autor:

Ralf Giering

Max-Planck-Institut  
für Meteorologie

MAX-PLANCK-INSTITUT  
FÜR METEOROLOGIE  
BUNDESSTRASSE 55  
D - 20146 HAMBURG  
F.R. GERMANY

Tel.: +49-(0)40-4 11 73-0  
Telefax: +49-(0)40-4 11 73-298  
E-Mail: <name> @ dkrz.de

ISSN 0938-5177

**Erstellung eines adjungierten Modells  
zur Assimilierung von Daten  
in ein Modell  
der globalen ozeanischen Zirkulation**

Dissertation  
zur Erlangung des Doktorgrades  
der Naturwissenschaften im Fachbereich  
Geowissenschaften  
der  
Universität Hamburg

vorgelegt  
von  
Ralf Giering

Hamburg  
1995

ISSN 0938-5177

Als Dissertation angenommen vom Fachbereich Geowissenschaften  
der Universität Hamburg

aufgrund der Gutachten von Herrn Prof. Dr. K. Hasselmann  
und Herrn Prof. Dr. W. Zahel

Hamburg, den 05.11.1995

Prof. Dr. M. Dunst  
Sprecher des Fachbereiches Geowissenschaften

## Zusammenfassung

Der Kombination von Informationen aus Daten und Modellen durch Datenassimilation kommt eine immer größere Bedeutung zu. Unterschiedlichste Eigenschaften des Ozeans werden gemessen, und die Anzahl an Messungen nimmt zu. Besonders die satellitengestützte Fernerkundung liefert seit einigen Jahren große Mengen an Messungen der Ozeanoberfläche. Auf der anderen Seite macht die Qualitätssteigerung der ozeanischen Zirkulationsmodelle inzwischen einen quantitativen Vergleich der Daten mit Modellsimulationen sinnvoll. Das Bestreben eine Modellintegration in größtmöglicher Nähe zu vorliegenden Daten zu halten wird gemeinhin als Datenassimilation bezeichnet

Unter den verschiedenen Methoden Daten in Modelle zu assimilieren, ist die adjungierte Methode ein sehr flexibles Werkzeug. Sowohl der Anfangszustand des Modells als auch der äußere Antrieb können angepaßt werden. Desweiteren lassen sich alle Arten von Messungen verwenden, sofern sich die gemessene Größe aus dem Modell ableiten läßt. Zur Erstellung des hierzu benötigten adjungierten Modells existieren verschiedene Verfahren. Die vorliegende Arbeit beschreibt einen systematischen Weg, wie auch komplexe Zirkulationsmodelle adjungiert werden können. Das Verfahren beruht auf dem Differenzieren von Algorithmen im Rückwärtsmodus. Ausgehend von dem kodierten Programm des Modells wird das adjungierte Modell Anweisung für Anweisung anhand von einfachen Regeln konstruiert. Die grundlegenden Konzepte werden vorgestellt und die Regeln erläutert. Zu einem globalen Modell der ozeanischen Zirkulation wurde auf diese Weise das adjungierte Modell erstellt.

In Zwillingsexperimenten können anhand von simulierten Messungen der Auslenkung und Temperatur der Ozeanoberfläche die Amplituden von Windschubmustern rekonstruiert werden. Dies gelingt auch, wenn die Daten mit einem weißen Rauschen, das die erwarteten statistischen Fehler dieser Messungen simuliert, versehen werden. Zur Unterdrückung eines  $2\Delta t$ -Signals in den Amplituden wird eine Glattheitsbedingung eingeführt. Eine Analyse der Fehlerkovarianzmatrix der Ergebnisse zeigt, daß dadurch das inverse Problem wesentlich besser bestimmt ist.

In weiteren Experimenten werden Oberflächendatensätze, die auf Analysen von Messungen beruhen, durch Variation des Windschubes assimiliert. Die Ergebnisse zeigen, daß eine dynamische Interpolation zwischen verschiedenen Arten von Daten mithilfe eines Zirkulationsmodells möglich ist. Kleine Änderungen von analysierten Windschubdaten genügen, um eine SST-Anomalie des Modells zu erzeugen, die sehr viel besser mit den Beobachtungen übereinstimmt.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Adjungierte Modelle</b>	<b>7</b>
2.1	Einführung am Beispiel der Datenassimilation . . . . .	7
2.2	Anwendungen . . . . .	9
2.3	Erstellung von adjungierten Modellen . . . . .	13
<b>3</b>	<b>Entwicklung des adjungierten Modells</b>	<b>15</b>
3.1	Differenzieren von Algorithmen . . . . .	15
3.2	Ableitung der Kostenfunktion . . . . .	16
3.3	Grundlegende Konzepte . . . . .	18
3.4	Adjungieren von Anweisungen . . . . .	20
3.5	Adjungierte Unterprogramme . . . . .	30
3.6	Zur Speicherung von Bedarfsvariablen . . . . .	31
3.7	Problematische Programmstrukturen . . . . .	32
3.8	Adjungieren von implizit erklärten Funktionen . . . . .	32
<b>4</b>	<b>Modellbeschreibung</b>	<b>37</b>
4.1	Modellgleichungen . . . . .	37
4.2	Diskretisierung . . . . .	38
4.3	Randbedingungen . . . . .	39
4.4	Implementierung . . . . .	40
<b>5</b>	<b>Experimente</b>	<b>41</b>
5.1	Zwillingsexperimente . . . . .	41
5.2	Assimilierung im Pazifik im Zeitraum 1982-83 . . . . .	51
<b>6</b>	<b>Diskussion und Ausblick</b>	<b>65</b>
	<b>Literatur</b>	<b>69</b>



# 1 Einleitung

Die Verknüpfung der Informationen aus Modelldynamik und Daten, Datenassimilation genannt, ist eine hilfreiche Technik zum besseren Verständnis des betrachteten Systems. In der Meteorologie wird es schon lange sehr intensiv zur Verbesserung der Wettervorhersage eingesetzt (Bengtsson, Ghil und Kallen 1981). Die Ozeanographie setzt die Datenassimilation bisher hauptsächlich zur Diagnose ein, denn hier gibt es sehr viel weniger Messungen im Verhältnis zu den raumzeitlichen Skalen der wesentlichen Prozesse. Satellitengestützte Messungen haben die Situation zwar für die Ozeanoberfläche verbessert (GEOSAT, Topex-Poseidon, ERS-1), doch auch in absehbarer Zukunft werden Ozeanographen ohne flächendeckende Messungen des inneren Ozeans auskommen müssen. Es besteht daher ein großes Interesse an der Rekonstruktion der inneren Schichtung des Ozeans aus Oberflächenmessungen.

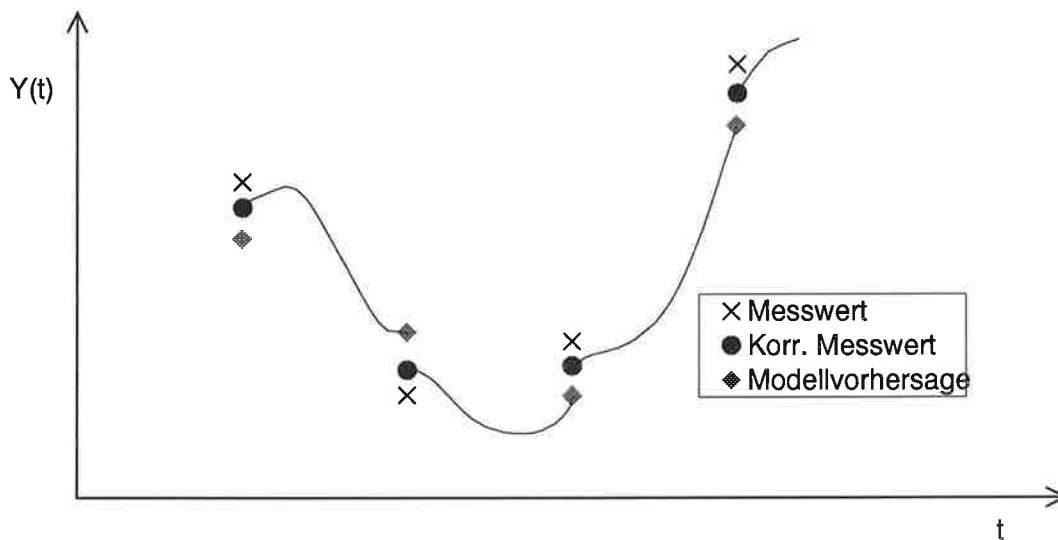
Verglichen mit der Atmosphäre ist der Ozean auf großen Skalen sehr viel stärker deterministisch. Bei genauer Kenntnis des Zustandes und der Randbedingungen ließe sich sein Verhalten sehr gut (auf Zeitskalen bis zu mehreren Jahren) diagnostizieren. Doch die äußeren Antriebe, wie Impuls, Wärme- und Frischwasserfluß, sind bis heute nur ungefähr bekannt. Da es in großen Teilen des Ozeans nur sehr wenige Messungen gibt, sind sowohl die Analysen der Windmessungen von Schiffen aus als auch die als Teil der operationellen Wettervorhersage gelieferten Windfelder in diesen Gebieten ungenau. Diese Situation wird sich bessern, da nun auch Radarrückstreuungsmessungen von Satelliten die Windstärke und das Seegangsspektrum flächendeckend liefern. Aus letzteren läßt sich durch Datenassimilation in ein Seegangmodell Information über den Wind gewinnen (Heras, Burgers und Janssen 1995). Im Gegensatz dazu sind Nettofrischwasser- und Nettowärme fluß prinzipiell schwer direkt zu bestimmen. Die Sicherheit über den Wärmefluß ist in vielen Gebieten des Ozeans größer als die Schätzung selber (Large und Pond 1982, Oberhuber 1988, Schiller 1993). Der äußere Antrieb muß daher hauptsächlich indirekt ermittelt werden. Dies kann mittels Datenassimilation entweder in Atmosphären- oder in Ozeanmodelle, am besten durch beides geschehen.

Die Kombination von Simulationsmodellen und Messungen durch Datenassimilation ist also für die Ozeanographie essentiell, um die großen Datenlücken zu schließen. Es wurden verschiedene internationale Programme durchgeführt und begonnen (Tropical Ocean Global Atmosphere, TOGA und World Ocean Circulation Experiment, WOCE) um neue Erkenntnisse über den Ozean zu gewinnen und weitere Forschungen anzustoßen. Die Zielsetzung von WOCE ist die Ermittlung der momentanen

großskaligen Zirkulation des Ozeans mittels einer Meßkampagne, der verbesserten Simulation und der Datenassimilation (Woods 1985). Dabei ist insbesondere die Datenassimilation gefordert, um die enormen Mengen an unterschiedlichsten Messungen optimal zu nutzen.

Die Methoden zur Datenassimilation können in zwei Gruppen eingeteilt werden. Einen detaillierten Einblick über die verschiedenen Methoden gibt das Buch von Anderson und Willebrand (1989).

*Sequentielle Methoden* integrieren das Modell einmal über das vorgegebene Zeitintervall und korrigieren den Systemzustand dabei aufgrund aktueller Messungen (Abb. 1). Die Schwierigkeit besteht darin, daß das Modell dadurch regelmäßig in



**Abbildung 1:** Skizzenhafte Darstellung der sequentiellen Methoden. Der Systemzustand wird hier nur durch einen Wert auf der Y-Achse repräsentiert.

einen neuen Zustand gebracht wird, der prinzipiell nicht verträglich mit seiner Dynamik ist. Inkonsistente Korrekturen injizieren eine unrealistisch hohe Energie in kleinen Wellenzahlen. Erst die Integration des Modells dämpft die ausgelösten Wellenprozesse (spin-up problem). Die verschiedenen sequentiellen Methoden unterscheiden sich im Grad der Konsistenz der Korrektur mit der Modelldynamik.

Ein sehr einfacher Ansatz besteht in dem *direkten Ersetzen* (direct insertion) von Modellvariablen durch Meßwerte (Hurlburt 1986). Dies ist nur sinnvoll, wenn sehr gute Daten flächendeckend vorhanden sind, da sonst durch große lokale Korrekturen leicht numerische Instabilitäten bei der weiteren Integration auftreten können.

Die *Relaxation* des Modells zu den Meßwerten (nudging), ist ein sehr robustes

Verfahren. Es berücksichtigt ebenso keine Modelldynamik und kann daher nur die gemessenen Modellgrößen gut anpassen. Dies Verfahren wird hauptsächlich zur Korrektur von groben Modellfehlern eingesetzt (Anthes 1974, Holland und Malanotte-Rizzoli 1989).

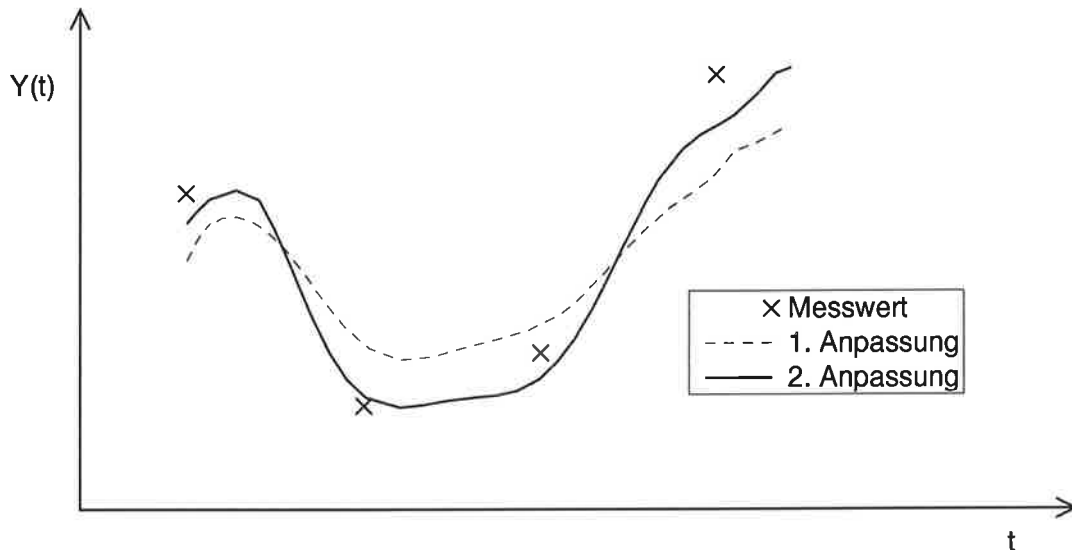
Das Verfahren der *sukzessiven Korrektur* (successive correction) verteilt die Information einer Messung auf ein Modellgebiet (Daley 1991, Fischer und Latif 1995). Dabei ist die Gewichtung abhängig vom räumlichen und zeitlichen Abstand zwischen Messung und Modellgitterpunkt. Die Gewichtungsfunktion wird empirisch bestimmt. Meist wird räumlich ein exponentieller Abfall mit zunehmender Entfernung und zeitlich ein linearer Abfall mit zunehmender Zeitdifferenz angenommen. Durch zusätzliche Unterscheidung zwischen zonalem und meridionalem Abstand ist dieser Ansatz in großen Gebieten des Ozeans brauchbar, er versagt aber in Bereichen starker Strömungen.

Die Methode der *optimalen Interpolation* basiert auf einem statistischen Ansatz (Lorenz 1981). Modellsimulation und Messungen stellen zwei unabhängige multivariate Schätzungen des Systemzustandes dar. Die optimale Schätzung ist die lineare Kombination mit der geringsten Fehlerkovarianz. Dies Verfahren setzt die Kenntnis der Kovarianzmatrizen der Meßfehler und der Modellsimulation voraus. Die Modellfehler sind stark korreliert, weil ein Zustand Informationen aus früheren Zuständen enthält und diese durch die Dynamik verteilt werden. Eine einzelne Messung wird also ein gewisses Modellgebiet beeinflussen. Im Gegensatz zu dem empirischen Ansatz der sukzessiven Korrektur wird hier die Modellfehlerkovarianz berechnet.

Wird die Fehlerstatistik des Modells mittels des Modells jeweils neu bestimmt, so gelangt man zur Methode des *Kalman-Filters* (Kalman 1960). Es wird der optimale Zustand unter Berücksichtigung aller früheren Messungen und ihrer Fehler bestimmt. Das Kalman-Filter ist für lineare dynamische Systeme wohldefiniert und es existieren Erweiterungen der Methode für nichtlineare Systeme ('extended Kalman filter', Ghil, Kohn, Tavantis, Bube und Isaacson (1981), und 'Kalman smoother', Bennet (1989)). Zusätzlich zur eigentlichen Assimilierung der Daten muß hier die Fehlerkovarianzmatrix des Systemzustandes mit jedem Zeitschritt der Modellintegration erneuert werden. Dies stellt einen erheblichen Aufwand dar. Deshalb wird diese Matrix meist nur approximiert, um Speicheraufwand und vor allem Rechenzeit auf ein handhabbares Maß zu reduzieren.

Die *Variationsmethoden* beschäftigen sich allgemein mit der Aufgabe die Extremwerte von Funktionalen zu finden. Bei den Variationsmethoden der Datenassimila-

tion definiert das Funktional den Abstand zwischen Modell und Daten entlang der Modelltrajektorie. Die Modellgleichungen sind dabei jederzeit exakt erfüllt (Abb. 2).



**Abbildung 2:** Skizzenhafte Darstellung der Variationsmethoden. Es sind mehrere Trajektorien gezeigt, sie unterscheiden sich im Wert der Kostenfunktion.

Die Variationsmethode der Datenassimilation, bereits 1970 von Sasaki vorgestellt, bekam erst durch die Arbeiten von Lewis und Derber (1985), Le Dimet und Talagrand (1986), Talagrand und Courtier (1987), Courtier und Talagrand (1987) und Thacker (1987) einen entscheidenden Anstoß. Im Gegensatz zu den sequentiellen Methoden wird nicht ein guter Zustand sondern eine optimale Trajektorie des Systems gesucht. Die Diskrepanz zwischen Modelltrajektorie und Daten wird durch Definition einer *Kostenfunktion* quantifiziert. Werden die Modellgleichungen als perfekt angenommen, dann hängt die Trajektorie nur von den Anfangs- und Randbedingungen ab. Ausgewählte Kontrollvariablen steuern diese Bedingungen, so daß insgesamt die Kostenfunktion in Abhängigkeit von den Kontrollvariablen formuliert ist. Da die Modellgleichungen immer erfüllt sind, spricht man auch von einer *dynamischen Interpolation* der Daten durch das Modell. Sämtliche statistischen Informationen über Daten und Modell gehen als Gewichte in diese Definition ein, so daß das Minimum der Kostenfunktion für den wahrscheinlichsten Satz von *Kontrollvariablen* erreicht wird (maximum likelihood). Die Datenassimilation besteht somit in der Minimierung der Kostenfunktion durch Variation der Kontrollvariablen.

Effektive Algorithmen zur Optimierung einer differenzierbaren Funktion verwenden deren Gradient um eine neue Suchrichtung zu bestimmen. Für sehr viele Kon-

trollvariablen ( $\approx 2 \cdot 10^5$ ) ist dessen Bestimmung durch Differenzenquotienten äußerst rechenzeitintensiv, da für jede Kontrollvariable eine Modellintegration nötig wäre. Die *adjungierte Methode* der Datenassimilation nutzt das *adjungierte Modell*. Es ist ein sehr effektiver Algorithmus um den Gradienten der Kostenfunktion bezüglich vieler Kontrollvariablen mittels einer einzigen Integration zu bestimmen.

Die Definition einer Kostenfunktion ermöglicht die einfache Einbindung zusätzlicher Information über das System durch Addition von entsprechenden 'Straf'termen. Solch 'a priori'-Information kann auch eine Glattheitsbedingung sein.

Das Kalman-Filter und die adjungierte Methode liefern für ein lineares Modell den gleichen Systemzustand für das Ende des Zeitintervalls in dem Daten assimiliert werden. Ihre Trajektorien unterscheiden sich, da das Kalman-Filter für eine Korrektur nur jene Daten berücksichtigt, welche älter als der aktuelle Zustand sind. Ein weiterer Unterschied besteht in der Handhabung der Modellfehlerkovarianz. Sie wird für das Kalman-Filter explizit errechnet, während sie bei der adjungierten Methode implizit im adjungierten Modell enthalten ist. Der große Vorteil der adjungierten Methode, daß diese Matrix nicht berechnet werden muß, ist gleichzeitig ein Nachteil, denn die Information über Modellfehler muß, falls erforderlich, extra ermittelt werden.

Die adjungierte Methode ist gleichermaßen für lineare und nichtlineare Modelle einsetzbar. Eine starke Nichtlinearität äußert sich in einer langsameren Konvergenz der Optimierung. Die Kostenfunktion hat dann u.U. mehrere Minima, und es hängt vom gewählten Startwert ab, welches lokale Minimum gefunden wird.

Ein Nachteil der adjungierten Methode ist ihr großer Bedarf an Rechnerressourcen. Das Modell und das zugehörige adjungierte Modell werden im Rahmen der Optimierung nicht nur einmal sondern mehrfach über den Assimilationszeitraum integriert. Dabei ist der Rechenaufwand zur Integration des adjungierten Modells bis zu einem Faktor 4 größer als derjenige für das Modell. Zur Integration des adjungierten Modells bedarf es im nichtlinearen Fall der Speicherung der gesamten Modelltrajektorie während der vorausgehenden Modellintegration. Dieser u.U. sehr große Speicherbedarf kann durch eine *Stützpunkttechnik* (checkpointing) auf Kosten einer zusätzlichen Modellintegration wesentlich reduziert werden.

Die vorliegende Arbeit soll die praktische Anwendbarkeit der adjungierten Methode für ein realistisches 3-dimensionales Ozeanmodell demonstrieren. Zunächst wird die adjungierte Methode in Kap. 2 im Detail erläutert. Die Anwendung dieser Methode setzt die Existenz des adjungierten Modells voraus. Die verschiedenen Ansätze

zur Konstruktion des adjungierten Modells werden vorgestellt. Der hier verwendete Ansatz beruht auf dem Differenzieren von Algorithmen. Nach kurzer Einführung in dieses Gebiet gibt Kap. 3 eine allgemeine Anleitung zur Konstruktion adjungierter Modelle. Nach den dargestellten Regeln wurde das adjungierte Modell des hier verwendeten Ozeanmodells erstellt. Nach der Beschreibung des Modells in Kap. 4 werden die Experimente vorgestellt. Zwillingsexperimente dienen der Überprüfung der Brauchbarkeit der Methode, d.h. der Untersuchung inwieweit die Methode in der Lage ist, aus der Reaktion des Modells auf vorgegebene Antriebsfelder ebendiese Antriebsfelder zu rekonstruieren. Experimente zur Assimilierung von analysierten Oberflächenmessungen schließen die Arbeit ab.

## 2 Adjungierte Modelle

### 2.1 Einführung am Beispiel der Datenassimilation

Es liegen ein Satz von beobachteten Größen  $d \in \mathbb{R}^m$  ( $m \in \mathbb{N}$ ) und ein Simulationsmodell vor, das die entsprechenden Modellgrößen  $y \in \mathbb{R}^m$  berechnen soll. Gesucht ist nach der bestmöglichen Übereinstimmung zwischen Beobachtungen und Modellwerten. Die Güte eines Satzes von Modellwerten wird quantifiziert durch Definition einer Kostenfunktion  $J$ , z.B.:

$$J = \frac{1}{2} \langle y - d, y - d \rangle \quad , \quad (1)$$

wobei  $\langle \cdot, \cdot \rangle$  ein Skalarprodukt in  $\mathbb{R}^m$  bezeichnet. Je kleiner der Wert von  $J$ , desto besser stimmen Modell und Beobachtungen überein.

Bestimmte Parameter  $x \in \mathbb{R}^n$  ( $n \in \mathbb{N}$ ) zur Steuerung des Modellaufs, die Kontrollvariablen, werden ausgewählt. Die funktionale Abhängigkeit der Modellwerte  $y$  von den Kontrollvariablen  $x$  sei durch die Abbildung

$$\begin{aligned} \mathcal{H} : \mathbb{R}^n &\rightarrow \mathbb{R}^m \\ x &\mapsto y := \mathcal{H}(x) \end{aligned} \quad (2)$$

gegeben. Die Kostenfunktion  $J$  ist dann eine Funktion dieser Kontrollvariablen:

$$\begin{aligned} J : \mathbb{R}^n &\rightarrow \mathbb{R} \\ x &\mapsto J(x) := \frac{1}{2} \langle \mathcal{H}(x) - d, \mathcal{H}(x) - d \rangle \end{aligned} \quad (3)$$

Das Problem besteht nun darin, die Werte der Kontrollvariablen zu finden, für die  $J$  minimal wird. Effektive Algorithmen zur Minimierung einer differenzierbaren Funktion verwenden den Gradienten  $\nabla_x J(x_0)$  der Kostenfunktion  $J$  bezüglich der Kontrollvariablen  $x$ . In erster Ordnung ist die Taylor-Entwicklung von  $J$  um einen Satz von Kontrollvariablen  $x_0$ :

$$J(x) = J(x_0) + \langle \nabla_x J(x_0), x_0 - x \rangle + o(|x_0 - x|^2) \quad (4)$$

oder einfach:

$$\delta J = \langle \nabla_x J(x_0), \delta x \rangle \quad (5)$$

Im folgendem wird diese Notation aus der Variationsrechnung für lineare Approximationen benutzt.

Wird  $\mathcal{H}$  als stetig differenzierbar vorausgesetzt, so kann für jede Variation der Kontrollvariablen  $\delta x$  eine Variation  $\delta y$  approximiert werden durch:

$$\delta y = A(x_0) \delta x \quad , \quad (6)$$

wobei  $A(x_0)$  die Funktionalmatrix<sup>1</sup> von  $\mathcal{H}$  an der Stelle  $x_0$  darstellt. Die Jacobimatrix stellt die Wirkung des tangentialen Modells dar. Eine Variation der Kostenfunktion läßt sich nun durch die Variation der Kontrollvariablen ausdrücken. Dabei wird die Symmetrie des Skalarproduktes ausgenutzt und die Produktregel angewendet.

$$\begin{aligned} \delta J &= \frac{1}{2} \langle \mathcal{H}(x_0) - d, A(x_0) \delta x \rangle + \frac{1}{2} \langle A(x_0) \delta x, \mathcal{H}(x_0) - d \rangle \\ &= \langle \mathcal{H}(x_0) - d, A(x_0) \delta x \rangle \quad . \end{aligned} \quad (7)$$

Durch Anwendung der Definition eines *adjungierten Operators*  $A^*$  bezüglich eines Skalarproduktes  $\langle \cdot, \cdot \rangle$ :

$$\langle A^* v, w \rangle := \langle v, A w \rangle \quad (8)$$

wird (7) umgeformt zu

$$\delta J = \langle A^*(x_0) (\mathcal{H}(x_0) - d), \delta x \rangle \quad . \quad (9)$$

Da  $\delta x$  beliebig ist, folgt mit (5):

$$\nabla_x J(x_0) = A^*(x_0) (\mathcal{H}(x_0) - d) \quad . \quad (10)$$

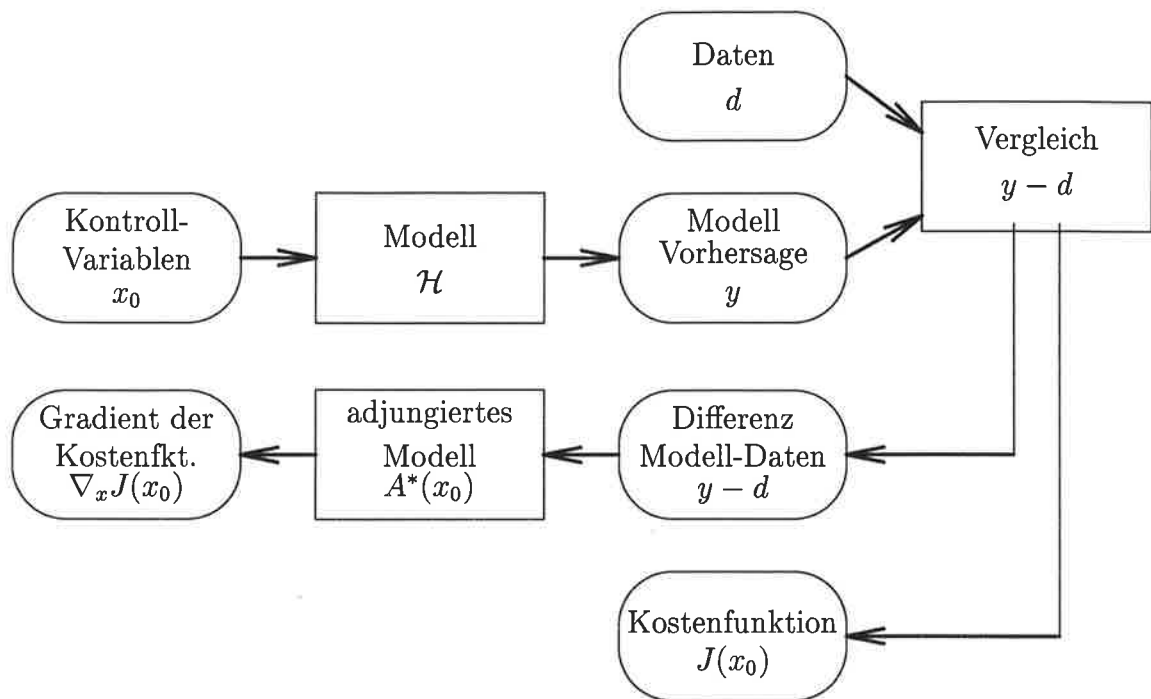
Der Operator  $A^*(x_0)$  beschreibt das adjungierte Modell. Es ist linear und adjungiert zum tangentialen Modell  $A(x_0)$ . Beide Operatoren hängen vom Punkt  $x_0$ , um den linearisiert wurde, ab. Die Differenz  $\mathcal{H}(x_0) - d$  heißt *Fehlanpassung* (innovation vector) und wirkt nach (10) als Antrieb des adjungierten Modells.

Die Berechnung der Kostenfunktion und des Gradientenvektors für einen vorgegebenen Satz von Kontrollvariablen ist in Abb. 3 schematisch dargestellt. Die Integration des adjungierten Modells ist vom Rechenaufwand vergleichbar mit der des tangentialen Modells. Die Anzahl der Operationen des tangentialen Modells wiederum ist abhängig von der Komplexität der Funktion  $\mathcal{H}$ . Das Verhältnis der Operationen ist jedoch unabhängig von der Anzahl der Kontrollvariablen. Daher wird auch die Rechenzeit zur Berechnung von Kostenfunktion und Gradientenvektor nur um einen konstanten Faktor größer sein als die zur Berechnung der Kostenfunktion allein. Eine detaillierte Analyse der notwendigen Operationen ergibt einen Faktor

---

<sup>1</sup>auch Jacobimatrix genannt





**Abbildung3:** Schematische Darstellung der Berechnung von Kostenfunktion und Gradientenvektor

zwischen 2 und maximal 5 (Baur und Strassen 1983, Griewank 1989). Die Approximation des Gradientenvektors  $\nabla_x J(x_0)$  durch Differenzenquotienten benötigt dagegen mindestens  $n + 1$  Modellläufe. Das adjungierte Modell hat deshalb gegenüber Differenzenquotienten zwei wesentliche Vorteile, der Gradient ist keine Approximation sondern exakt, und da alle Komponenten simultan berechnet werden, ist bei vielen Kontrollvariablen sehr viel weniger Rechenzeit nötig.

## 2.2 Anwendungen

### 2.2.1 Datenassimilation

Die adjungierte Methode der Datenassimilation optimiert den Anfangszustand des Modells oder den äußeren Antrieb durch Minimierung einer Kostenfunktion. Dabei können die Werte an allen Gitterpunkten variiert werden oder nur (zeitlich veränderliche) Amplituden von Strukturfunktionen. Die Kostenfunktion kann aus mehreren Summanden bestehen, welche unterschiedliche Informationen über das System beschreiben.

Der wichtigste Term beschreibt den Abstand zwischen Modell und Daten. Um

einen Meßwert mit dem Modell vergleichen zu können muß der zu dem Ort der Messung korrespondierende Modellwert berechnet werden. Solch eine Abbildung  $f$  von den Modellgitterwerten  $m$  zu der Modellvorhersage  $f(m)$  für den Meßwert  $d$  enthält zum einen die Diagnose der gemessenen Größe aus den simulierten prognostischen Modellgrößen und zum anderen eine Interpolation der Gitterwerte auf den Meßort. Gehorchen die Fehler einer Normalverteilung, dann hat der Term die Gestalt:

$$J_D = \frac{1}{2} (f(m) - d)^t (F + E)^{-1} (f(m) - d) , \quad (11)$$

wobei  $E$  die Fehlerkovarianzmatrix der Messungen bezeichnet und  $F$  diejenige der Abbildung  $f$ . In einem numerischen Modell repräsentiert ein Gitterwert ein endliches Gebiet im Orts- oder Frequenzraum, Prozesse auf Skalen unterhalb dieser Auflösung werden nicht simuliert sondern parametrisiert. Repräsentieren Messungen eine kleinere Skala, so bewirken diese Prozesse eine höhere Variabilität der Meßwerte an einem bestimmten Ort als die der Modellwerte. Dieser Abbildungsfehler sollte berücksichtigt werden, da er sehr viel größer als der Meßfehler sein kann.

Wird der Anfangszustand des Modells  $m_0$  variiert und existiert bereits eine Schätzung  $m_b$  (background) für diesen (z.B. durch Datenassimilation über ein vorangehendes Zeitintervall), so quantifiziert ein zusätzlicher Term  $J_B$  in der Kostenfunktion die Abweichungen von der Schätzung:

$$J_B = \frac{1}{2} (m_0 - m_b)^t B^{-1} (m_0 - m_b) , \quad (12)$$

dabei ist  $B$  die Fehlerkovarianzmatrix für die Schätzung  $m_b$ .

Alle anderen Informationen über das System werden ebenso als Terme in der Kostenfunktion berücksichtigt. Eine Glattheitsbedingung für eine Größe  $x$  kann z.B. als Term:

$$J_G = \frac{1}{2} w \nabla^2 x \quad (13)$$

oder

$$J_G = \frac{1}{2} w \nabla^6 x , \quad (14)$$

je nach erwartetem Frequenzspektrum der Größe, realisiert werden. Die Gewichtung  $w$  ist nicht festgelegt und kann je nach gewünschter Glattheit gewählt werden.

Es sollte darauf geachtet werden, daß genügend Einschränkungen für die zu variierenden Parameter in der Kostenfunktion vorhanden sind. Andernfalls kann das Problem unterbestimmt sein, d.h die Lösung ist nicht eindeutig. Die durch Optimierung gefundene Lösung hängt dann von den gewählten Startwerten der Kontrollvariablen ab.

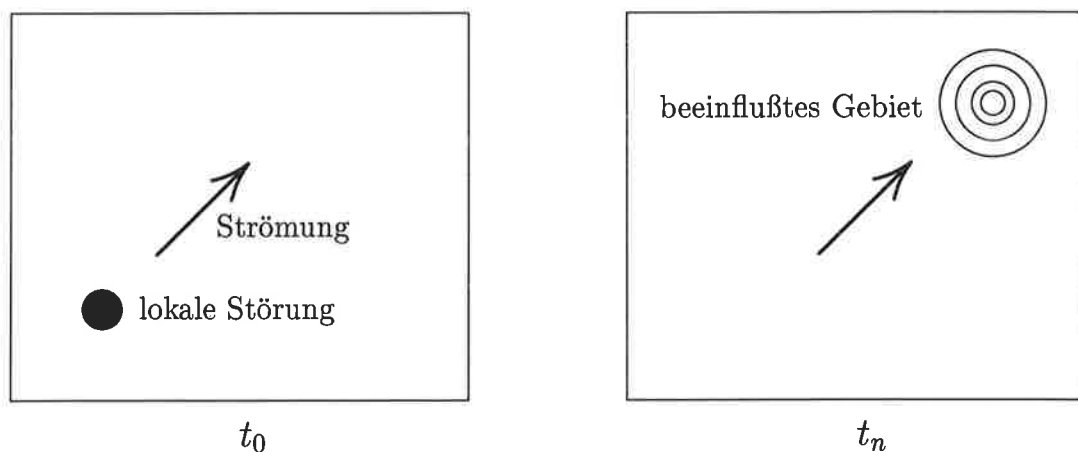
Die Konvergenz der Optimierung wird wesentlich von der Linearität des Modells bestimmt. Desweiteren beeinflusst die Kondition<sup>2</sup> der Hesseschen Matrix die Konvergenz. Je kleiner die Konditionszahl ist, desto besser ist die Konvergenz. Die Konditionszahl wird durch Addition einer diagonaldominanten Matrix verkleinert und somit die Konvergenz verbessert. Die Terme von 'a priori'-Informationen haben diese Eigenschaft meist.

### 2.2.2 Modell-Tuning

Zur Entwicklung eines numerischen Simulationsmodells gehört die Einstellung verschiedener Parameter in den zugrundeliegenden Gleichungen (z.B. Kopplungsparameter, Diffusionskonstanten). Dies kann systematisch durch Minimierung einer Kostenfunktion bei Variation der Parameter geschehen. So werden anhand von Beobachtungsdaten die Modellgleichungen selbst optimiert (Schröter 1992, Barzel 1994).

### 2.2.3 Sensitivitätsanalyse

Das tangentielle Modell dient zur Untersuchung der Auswirkungen von kleinen Störungen. Wird z.B. in einem horizontalen Strömungsmodell an einem Ort die Temperatur verändert, so wird diese Anomalie stromabwärts getragen und durch die Diffusion verbreitert (Abb. 4).



**Abbildung 4:** Vorwärts-Modell : Advektion und Diffusion von Störungen (Integration von  $t_0$  zu  $t_n$ )

<sup>2</sup>Verhältnis zwischen größtem und kleinstem Eigenwert



**Abbildung 5:** Adjungiertes Modell : Advektion und Diffusion von Einfluß (Integration von  $t_n$  zu  $t_0$ )

Das adjungierte Modell kann dagegen zur Analyse der Ursachen bestimmter Anomalien herangezogen werden (Abb. 5). Das Gebiet möglicher Ursachen für eine spätere Anomalie liegt stromaufwärts. Die Diffusion bewirkt hier eine Verbreiterung des Gebietes möglicher Ursachen.

#### 2.2.4 Führende instabile Moden (most unstable modes)

Für die Vorhersage von Systemen ist es wichtig zu wissen, welche anfänglichen Störungen am stärksten wachsen und damit die Prognose verschlechtern. Betrachten wir daher ein Modell  $\mathcal{H}$  (Def. 2), welches aus einem Zustand  $x$  den Zustand  $y$  zu einem späteren Zeitpunkt berechnet. Es wird nun die Variation  $\delta x$  um einen Zustand  $x$  gesucht, welche eine Variation  $\delta y$  mit einer möglichst großen Norm verursacht.

$$\begin{aligned} \|\delta y\|^2 &= \langle \delta y, \delta y \rangle \\ &= \langle A\delta x, A\delta x \rangle \\ &= \langle \delta x, A^*A\delta x \rangle \end{aligned} \quad (15)$$

Eine Variation  $\delta x$  der Kontrollvariablen mit gegebener Norm bewirkt daher die größtmögliche Anomalie, falls sie in Richtung des Eigenvektors  $v$  zum größten Eigenwert  $\lambda$  des Operators  $A^*A$  verläuft:

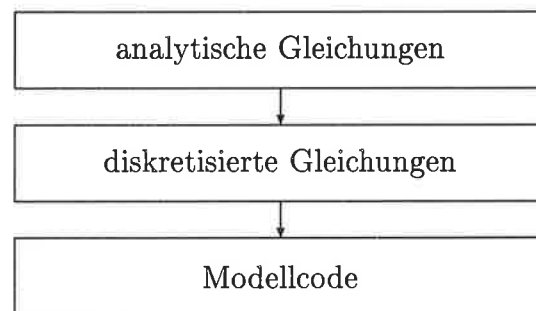
$$\lambda_m = \max, \quad A^*A v_i = \lambda_i v_i. \quad (16)$$

Zur Bestimmung extremer Eigenwerte und zugehöriger Eigenvektoren einer großen Matrix ist der Lanczos-Algorithmus (Lanczos 1957), eine Erweiterung des Ritz-

Verfahrens, gut geeignet, denn er benötigt nicht die einzelnen Matrixelemente, sondern nur das Ergebnis der Anwendung des Operators auf einen Vektor. So kann durch mehrmaligen Aufruf des tangentialen und des adjungierten Modells eine Approximation der größten Eigenvektoren gefunden werden (siehe z.B. Buizza, Tribba, Molteni und Palmer (1993)).

## 2.3 Erstellung von adjungierten Modellen

Bei der Entwicklung eines Simulationmodells werden üblicherweise, die in der Abb. 6 dargestellten Stufen durchlaufen. Nach Formulierung der analytischen Differentialgleichungen wird ein Diskretisierungsschema gewählt und die diskreten Gleichungen aufgestellt. Der letzte Schritt besteht dann darin, einen Algorithmus zur Lösung dieser Gleichungen in einer Programmiersprache zu formulieren. Die Entwicklung des adjungierten Modells kann an jedem dieser Schritte ansetzen.



**Abbildung6:** Schematische Darstellung der Entwicklung eines numerischen Modells

### 2.3.1 Adjungieren der analytischen Gleichungen

Die analytischen Gleichungen werden anhand der Regeln für adjungierte analytische Operatoren adjungiert. Die erhaltenen Gleichungen werden dann diskretisiert und ein Algorithmus zur Lösung dieser Gleichungen entwickelt.

Anhand der adjungierten analytischen Gleichungen kann die Informationsdynamik studiert werden, d.h. der durch die Gleichungen vermittelte Einfluß von adjungierten Variablen auf andere adjungierte Variablen. Bei der Diskretisierung ist darauf zu achten, daß das Resultat auch adjungiert zu den diskreten Modellgleichungen ist, denn Diskretisierung und Differentiation vertauschen als Operatoren nicht. Diese Methode wird im wesentlichen auf einfache Box-Modelle angewendet, (z.B. Schröter (1989)).

### 2.3.2 Adjungieren der diskreten Gleichungen

Aus Kostenfunktion und diskreten Gleichungen wird die Lagrange-Funktion aufgestellt. Die Ableitung dieser Funktion nach den Modellvariablen liefert die adjungierten diskreten Gleichungen. Als letzter Schritt wird wieder der adjungierte Code entwickelt.

Für diese Vorgehensweise ist keinerlei Kenntnis über adjungierte Operatoren nötig. Doch die Aufstellung und Ableitung der Lagrange-Funktion verlangt viel Schreibarbeit und große Sorgfalt insbesondere im Umgang mit mehrfachen Indizes. Die Randbedingungen müssen meist extra behandelt werden.

Thacker (1988) hat diese Methode vorgestellt und zunächst auf einfache Modelle angewendet. Der adjungierte Code eines Modells basierend auf den primitiven Gleichungen (GFDL-Ozeanmodell) wurde auf diese Weise entwickelt (Long und Thacker 1989, Tziperman und Thacker 1989, Tziperman, Thacker, Long und Hwang 1992, Tziperman, Thacker, Long, Hwang und Rintoul 1992).

### 2.3.3 Adjungieren des Modellcodes

Der Code eines numerischen Modells besteht aus vielen einzelnen Anweisungen. Der Algorithmus kann daher als eine Verkettung von Funktionen angesehen werden, wobei jede Funktion eine Anweisung darstellt. So wird das ganze Problem auf das Adjungieren von elementaren Anweisungen reduziert, die unter Anwendung der Kettenregel verknüpft werden. Dafür können einfache Regeln aufgestellt werden (Talagrand 1991, Thacker 1991, Giering und Kaminski 1996a).

Mit dieser Methode ist es möglich auch sehr umfangreiche Modelle mit komplizierten Randbedingungen zu adjungieren. Die Fehlersuche im adjungierten Code vereinfacht sich wesentlich, denn die zu einer Anweisung adjungierten Anweisungen sind an einer ganz bestimmten Stelle im adjungierten Code lokalisierbar. Ebenso ist es unproblematisch, nach Änderungen des Modells, das adjungierte Modell entsprechend anzupassen. Diese Methode beruht auf der Differentiation von Algorithmen, sie wird im nächsten Kapitel detailliert vorgestellt.

### 3 Entwicklung des adjungierten Modells

#### 3.1 Differenzieren von Algorithmen

Es sei

$$\begin{aligned} \mathcal{H} : \mathbb{R}^n &\rightarrow \mathbb{R}^m \\ x &\mapsto y \end{aligned} \quad (17)$$

eine Abbildung, gegeben durch einen numerischen Algorithmus. Die explizite Darstellung von  $\mathcal{H}$  kann sehr kompliziert sein, da der Algorithmus möglicherweise lang und komplex ist. Es ist jedoch häufig möglich, den Algorithmus in einzelne Schritte zu zerlegen, die ihrerseits eine einfache analytische Darstellung haben:

$$\begin{aligned} \mathcal{H}^l : \mathbb{R}^{n_{l-1}} &\rightarrow \mathbb{R}^{n_l} & (l = 1, \dots, k) \\ z^{l-1} &\mapsto z^l \end{aligned} \quad (18)$$

Dann ist  $\mathcal{H}$  eine Verkettung dieser Funktionen  $\mathcal{H}^l$

$$\mathcal{H} = \bigcirc_{l=1}^k \mathcal{H}^l, \quad (19)$$

mit  $n_1 = n$ ,  $z^0 = x$ ,  $n_k = m$  und  $z^k = y$ . Für eine differenzierbare Funktion  $\mathcal{H}$  ist die Funktionalmatrix  $A$  an der Stelle  $x_0$  definiert durch:

$$A_{ij}(x_0) := \left. \frac{\partial \mathcal{H}_i(x)}{\partial x_j} \right|_{x=x_0} \quad (i = 1, \dots, m; j = 1, \dots, n). \quad (20)$$

Nach Anwendung der Kettenregel auf (19) folgt:

$$A(x_0) = \prod_{l=1}^k \left. \frac{\partial \mathcal{H}^l}{\partial z^{l-1}} \right|_{z^{l-1} = z_0^{l-1}}, \quad (21)$$

wobei

$$z_0^{l-1} = \bigcirc_{j=1}^{l-1} \mathcal{H}^j(x_0) \quad (l = 2, \dots, k) \quad (22)$$

$$z_0^0 = x_0 \quad (23)$$

ist. Die Auswertung des Matrixproduktes auf der rechten Seite von (21) erfordert  $k - 1$  Matrixmultiplikationen. Da diese Multiplikation eine assoziative Verknüpfung ist, gibt es  $(k - 1)!$  Möglichkeiten die Reihenfolge der Multiplikationen festzulegen. Die Rechenzeit für das Produkt hängt von der Reihenfolge der Multiplikationen ab, weil die Matrizen i.a. unterschiedlich groß und dicht besetzt sind. Die Suche nach der

Permutation mit der geringsten Rechenzeit ist ein NP-wertiges Problem<sup>3</sup> (Griewank und Reese 1991).

Für zwei extreme Strategien zur Auswertung können die Eigenschaften geschlossen diskutiert werden. Im *Vorwärtsmodus* (forward mode) wird das Produkt in der gleichen Reihenfolge berechnet wie die Verkettung (19), d.h. es wird zunächst  $\frac{\partial \mathcal{H}^2}{\partial z^1} \cdot \frac{\partial \mathcal{H}^1}{\partial z^0}$  berechnet und dann das Ergebnis von links mit  $\frac{\partial \mathcal{H}^3}{\partial z^2}$  multipliziert usw.. Bei dem anderen Extrem, dem *Rückwärtsmodus* (reverse mode), wird umgekehrt verfahren, die Rechnung beginnt mit dem Produkt  $\frac{\partial \mathcal{H}^k}{\partial z^{k-1}} \cdot \frac{\partial \mathcal{H}^{k-1}}{\partial z^{k-2}}$ . Beide Methoden unterscheiden sich in der Größe der Matrizen, die als Zwischenergebnisse auftreten. Im ersten Fall haben diese immer  $n$  Spalten, im zweiten immer  $m$  Zeilen. Daher benötigt für  $m > n$  der Vorwärtsmodus weniger Operationen als der Rückwärtsmodus und für  $m < n$  ist es umgekehrt<sup>4</sup>. Ein wichtiger Unterschied zwischen beiden Methoden besteht in der Erfordernis von Zwischenergebnissen  $z^l$ . Im Vorwärtsmodus werden sie in der gleichen Reihenfolge benötigt, wie sie zur Berechnung der Kostenfunktion  $\mathcal{H}$  anfallen. Es ist daher leicht möglich, Gradienten und Kostenfunktion nebeneinander zu berechnen. Im Rückwärtsmodus dagegen werden sie in der umgekehrten Reihenfolge gebraucht. Ihre Bereitstellung erfordert zusätzliche Anweisungen im adjungierten Modell.

### 3.2 Ableitung der Kostenfunktion

Die Kostenfunktion des Kap. 2 ist eine skalarwertige Funktion,  $m = 1 \ll n$ . Daher wird der Rückwärtsmodus meist mit weniger Operationen auskommen. Für  $m = 1$  nennt man den Rückwärtsmodus auch adjungierten Modus. Bei der Auswertung des Produkts (21) fallen hier für  $m = 1$  immer nur Vektoren als Ergebnisse an.

Es sei die Kostenfunktion  $J$  durch eine Verkettung von Funktionen  $\mathcal{H}^l$  (18) gegeben:

$$J = \bigodot_{l=1}^k \mathcal{H}^l, \quad (24)$$

wobei  $n_k = m = 1$  ist. Für ein Zwischenergebnis (22) hängt die Variation von der Variation der Kontrollvariablen ab:

$$\delta z^l = \frac{\partial}{\partial x} \left( \bigodot_{i=1}^l \mathcal{H}^i \right) (x) \Big|_{x=x_0} \delta x. \quad (25)$$

<sup>3</sup>Die Rechenzeit zur exakten Lösung eines NP-wertigen Problems steigt exponentiell mit seiner Dimension.

<sup>4</sup>Wenn die Matrizen schwach besetzt sind und diese Eigenschaft zur Reduktion der Operationen ausgenutzt wird, kommen noch andere Kriterien zum tragen (Bischof und Hovland 1991).



Die Variationen aufeinanderfolgender Zwischenergebnisse sind durch

$$\delta z^l = \left. \frac{\partial \mathcal{H}^l(z^{l-1})}{\partial z^{l-1}} \right|_{z^{l-1} = z_0^{l-1}} \delta z^{l-1} \quad (26)$$

miteinander verbunden, wobei  $\delta z^0 := \delta x$  ist. Die adjungierte Variation  $\delta^* z^l$  eines Zwischenergebnisses  $z^l$  ist nun definiert als die Ableitung der Kostenfunktion bezüglich des Zwischenergebnisses:

$$\delta^* z^l := \nabla_{z^l} J(z^l) \Big|_{z^l = z_0^l} . \quad (27)$$

Mit der Definition des Gradienten (5) folgt:

$$\delta J = \langle \delta^* z^l, \delta z^l \rangle . \quad (28)$$

Die adjungierten Werte beschreiben also den Einfluß des zugehörigen Zwischenergebnisses auf die Kostenfunktion. Da (28) für alle  $l$  gilt, folgt aus (26):

$$\begin{aligned} \langle \delta^* z^{l-1}, \delta z^{l-1} \rangle &= \langle \delta^* z^l, \delta z^l \rangle \\ &= \left\langle \delta^* z^l, \left( \frac{\partial \mathcal{H}^l(z^{l-1})}{\partial z^{l-1}} \right) \Big|_{z^{l-1} = z_0^{l-1}} \delta z^{l-1} \right\rangle \\ &= \left\langle \left( \frac{\partial \mathcal{H}^l(z^{l-1})}{\partial z^{l-1}} \right)^* \Big|_{z^{l-1} = z_0^{l-1}} \delta^* z^l, \delta z^{l-1} \right\rangle . \end{aligned}$$

Dies gilt für alle  $\delta z^{l-1}$  und damit:

$$\delta^* z^{l-1} = \left( \frac{\partial \mathcal{H}^l(z^{l-1})}{\partial z^{l-1}} \right)^* \Big|_{z^{l-1} = z_0^{l-1}} \delta^* z^l , \quad (29)$$

oder komponentenweise:

$$\delta^* z_i^{l-1} = \sum_{j=1}^{n_l} \frac{\partial \mathcal{H}_j^l(z^{l-1})}{\partial z_i^{l-1}} \Big|_{z^{l-1} = z_0^{l-1}} \delta^* z_j^l . \quad (30)$$

Der letzte Schritt berechnet den Gradienten der Kostenfunktion bzgl. der Kontrollvariablen:

$$\delta^* z^0 = \delta^* x = \nabla_x J . \quad (31)$$

Gleichung (30) ist die grundlegende Beziehung für einen Schritt im Rückwärtsmodus.

### 3.3 Grundlegende Konzepte

Die elementaren Schritte in einem Programm sind die Anweisungen. Ein numerisches Modell ist im wesentlichen nur aus wenigen Typen von Anweisungen aufgebaut:

- Zuweisungen
- bedingte Anweisungen
- Wiederholungsanweisungen
- Prozeduraufrufe
- Anweisungsfolgen

Für diese Anweisungstypen sollen im folgenden Regeln zur Konstruktion des zugehörigen adjungierten Codes aufgestellt werden. Sofern für Beispiele notwendig, wird dabei auf die im wissenschaftlichen Bereich weit verbreitete Programmiersprache Fortran zurückgegriffen. Doch zunächst sollen einige grundlegende Konzepte vorgestellt werden, sie haben sich als hilfreich für die Entwicklung und Pflege von adjungierten Modellen herausgestellt.

Im folgenden wird das Programm zur Berechnung der Kostenfunktion als Code bezeichnet und dasjenige zur Berechnung des Gradienten im Rückwärtsmodus adjungierter Code.

#### 3.3.1 Adjungierte Variablen

In einem Programm halten Variablen die Zwischenergebnisse des vorangehenden Abschnitts. Diese Variablen können jedoch beliebig oft mit weiteren Ergebnissen überschrieben werden. Es ist daher möglich, daß zwei Zwischenergebnisse  $z_i^l$  und  $z_j^p$  ( $l \neq p$ ) in der gleichen Variablen aber an verschiedenen Abschnitten der Programmabfolge gespeichert werden.

Der adjungierte Code berechnet die zugehörigen adjungierten Werte  $\delta^* z_i^l$  der Zwischenergebnisse  $z_i^l$ . Es ist ausreichend, zu jeder Variablen im Code genau eine adjungierte Variable im adjungierten Code zu definieren. Sie hält die jeweiligen adjungierten Werte der Zwischenergebnisse der Variablen.

### 3.3.2 Aktive Variablen

Variablen, die von den Kontrollvariablen abhängen und Einfluß auf die Kostenfunktion haben, werden aktive Variablen genannt. Alle anderen Variablen sind passive Variablen, sie sind unabhängig von den Kontrollvariablen (z.B. vorgegebene Konstanten) oder haben keinen Einfluß auf die Kostenfunktion. Nur Anweisungen, die aktive Variablen berechnen, müssen adjungiert werden.

Zur Bestimmung der aktiven Variablen wird eine interprozedurale Datenabhängigkeitsanalyse vorgenommen. Ausgehend von den Kontrollvariablen werden nacheinander alle Variablen bestimmt, deren Werte von denen der Kontrollvariablen abhängen. Desweiteren werden ausgehend von der Kostenfunktion alle Variablen bestimmt, deren Werte den Wert der Kostenfunktion beeinflussen. Die Schnittmenge beider Variablenmengen enthält die aktiven Variablen.

### 3.3.3 Lokalität

Die Reihenfolge der Anweisungen des Codes und die systematische Anwendung der Kettenregel im Rückwärtsmodus legen die Position von adjungierten Anweisungen innerhalb des adjungierten Codes fest. Auch wenn dies Rechenzeit sparen könnte, werden Anweisungen nicht zusammengefaßt. Es besteht somit eine direkte Korrespondenz zwischen Anweisungen und den zugehörigen adjungierten Anweisungen. Als direkte Konsequenz wird jedem Unterprogramm, welches aktive Variablen berechnet, genau ein adjungiertes Unterprogramm zugeordnet. Dadurch kann nach Änderungen im Code der zugehörige adjungierte Code schnell lokalisiert und entsprechend angepaßt werden. Ebenso wird dadurch die Fehlersuche im adjungierten Code wesentlich vereinfacht.

### 3.3.4 Modularität

Jeder Teil des Codes soll weitgehend unabhängig von anderen Teilen adjungiert werden. Nun sind aber viele adjungierte Anweisungen von Zwischenergebnissen, d.h. von Werten der Variablen des ursprünglichen Codes abhängig. Die Variablen mögen in Indexausdrücken, Schleifengrenzen oder Bedingungen auftreten. Es können aber auch aktive Variablen in nichtlinearen Ausdrücken auf der rechten Seite von Zuweisungen sein. Ihre Werte werden für die Berechnung adjungierter Werte benötigt und müssen mit denjenigen im ursprünglichen Code übereinstimmen. Solche Variablen

werden im folgenden *Bedarfsvariablen* genannt.

Es wird nun gefordert, daß alle Bedarfsvariablen einer adjungierten Anweisung zu Beginn genau den Wert haben, den sie auch im Code vor der entsprechenden Anweisung hatten. Unter dieser Voraussetzung kann der adjungierte Code einer Anweisung unabhängig von den vorhergehenden und nachfolgenden Anweisungen konstruiert werden. Ebenso ermöglicht es die Erstellung von einfachen Regeln. Um diese Forderung zu erfüllen müssen eventuell zusätzliche Anweisungen im adjungierten Code eingefügt werden.

### 3.3.5 Nomenklatur

Es ist zur besseren Lesbarkeit des adjungierten Codes empfehlenswert eine einprägsame Konvention zur adjungierten Namensgebung einzuhalten. Alle adjungierten Namen können sich z.B. aus Kennbuchstaben AD und dem Namen aus dem Code zusammensetzen. In den folgenden Beispielen wird nach dieser Konvention verfahren. Es muß selbstverständlich darauf geachtet werden, daß die neuen adjungierten Namen sich von bereits vergebenen Namen unterscheiden. Gegebenenfalls ist ein alternatives Verfahren zur Namensgebung anzuwenden.

Alle Bedarfsvariablen bekommen im adjungierten Code den Namen den sie auch im Code haben. Anweisungen zur erneuten Berechnung dieser Variablen im adjungierten Code können somit direkt aus dem Code übernommen werden. Auch dies erhöht die Lesbarkeit des adjungierten Codes.

## 3.4 Adjungieren von Anweisungen

### 3.4.1 Zuweisung

Eine Zuweisung wird nur adjungiert, wenn sie eine aktive Variable berechnet. Die zu berechnende aktive Variable und alle aktiven Variablen des Ausdrucks auf der rechten Seite, ausgenommen Variablen zur Berechnung von Indizes, bilden den Ein- und Ausgabevektor. Dabei sind mehrdimensionale Variablen, mit unterschiedlichen Indizierungen, als verschiedene Variablen zu behandeln, falls sie auf verschiedene Adressen verweisen<sup>5</sup>. Die Zuweisung kann damit als ein Operator aufgefaßt werden, der auf den Eingangsvektor wirkt.

---

<sup>5</sup>Um dies festzustellen ist eine Analyse der Indexausdrücke durchzuführen.

Zur Berechnung der Funktionalmatrix dieser Zuweisung wird zunächst die tangentielle Zuweisung konstruiert. Die Faktoren vor den Variationen bilden die erste Zeile der Funktionalmatrix. Die anderen Zeilen besitzen in der Diagonalen eine Eins und ansonsten Nullen. Die adjungierte Matrix wird durch Transposition gebildet. Aus ihr lassen sich die adjungierten Zuweisungen ablesen. Die formale Einführung der Funktionalmatrix und deren Transponierter dient nur der Veranschaulichung; die adjungierten Zuweisungen können auch direkt aus (30) abgeleitet werden.

Die Prinzipien zur Erstellung des adjungierten Codes sollen an der folgenden Zuweisung illustriert werden:

$$Z = X * \text{SIN}(Y**2) \quad .$$

Der Ein- und Ausgabevektor dieser Zuweisung besteht aus den Variablen X, Y und Z. Alle drei Variablen seien aktiv. Die tangentielle Zuweisung ergibt sich dann zu

$$\delta Z = (\text{SIN}(Y ** 2)) * \delta X + (X * \text{COS}(Y ** 2) * 2 * Y) * \delta Y \quad .$$

Dies kann auch als Funktionalmatrix geschrieben werden:

$$\begin{pmatrix} \delta Z \\ \delta Y \\ \delta X \end{pmatrix}^l = \begin{pmatrix} 0 & X^{l-1} * \text{COS}(Y^{l-1} ** 2) * 2 * Y^{l-1} & \text{SIN}(Y^{l-1} ** 2) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta Z \\ \delta Y \\ \delta X \end{pmatrix}^{l-1} \quad .$$

Dabei kennzeichnet der Index  $l$  den Zustand nach und der Index  $l - 1$  den Zustand vor Ausführung der Zuweisung. Die adjungierte Matrix ist die transponierte Matrix, somit lautet die adjungierte Gleichung:

$$\begin{pmatrix} \delta^* Z \\ \delta^* Y \\ \delta^* X \end{pmatrix}^{l-1} = \begin{pmatrix} 0 & 0 & 0 \\ X^{l-1} * \text{COS}(Y^{l-1} ** 2) * 2 * Y^{l-1} & 1 & 0 \\ \text{SIN}(Y^{l-1} ** 2) & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta^* Z \\ \delta^* Y \\ \delta^* X \end{pmatrix}^l \quad .$$

Dies wird durch Anwendung der in Kap. 3.3.5 vereinbarten Nomenklatur in folgende Zuweisungen übersetzt:

$$\begin{aligned} \text{ADY} &= \text{ADY} + \text{ADZ} * X * \text{COS}(Y**2) * 2 * Y \\ \text{ADX} &= \text{ADX} + \text{ADZ} * \text{SIN}(Y**2) \\ \text{ADZ} &= 0.0 \quad . \end{aligned}$$

Die adjungierte Variable ADZ muß auf Null zurückgesetzt werden. Der Grund dafür ist das Überschreiben des alten Wertes der Variablen Z durch die Zuweisung im Code.

Dieser Wert hat dann keinen Einfluß auf die Kostenfunktion. Zu den beiden anderen adjungierten Variablen wird jeweils der zusätzliche Einfluß addiert. Die Werte der zugehörigen Variablen  $X$ ,  $Y$  gehen ja nicht verloren und können auch später, über andere Zuweisungen, einen zusätzlichen Einfluß auf die Kostenfunktion haben. Die Zuweisung an die adjungierte Variable auf der linken Seite muß immer als letztes erfolgen, denn ihr ursprünglicher Wert wird für die anderen adjungierten Zuweisungen benötigt.

Zur Ausführung dieser adjungierten Zuweisungen müssen die Variablen  $X$  und  $Y$  dieselben Werte besitzen, die sie auch in dem Moment vor Ausführung der ursprünglichen Zuweisung hatten; sie sind Bedarfsvariablen der adjungierten Zuweisungen.

### 3.4.2 Bedingte Anweisung

Eine bedingte Anweisung im Code (Bsp. 1 links) bewirkt, daß entweder die Anweisung  $A$ ,  $B$  oder  $C$  ausgeführt wird. Da im adjungierten Code die entsprechende

<u>Code</u>	<u>adjungierter Code</u>
IF (Bedingung A) THEN	IF (Bedingung A) THEN
Anweisung A	adjungierte Anweisung A
ELSE IF (Bedingung B)	ELSE IF (Bedingung B)
Anweisung B	adjungierte Anweisung B
ELSE	ELSE
Anweisung C	adjungierte Anweisung C
END IF	END IF

#### Beispiel 1: *adjungierte bedingte Anweisung*

adjungierte Anweisung ausgeführt werden soll, muß der boolesche Wert der Bedingungen bekannt sein, um zwischen den Möglichkeiten auszuwählen. Entweder werden die aktuellen Wahrheitswerte im Code für den adjungierten Code gespeichert oder es werden die gleichen Bedingungen erneut ausgewertet (Bsp. 1 rechts).

### 3.4.3 Wiederholungsanweisungen (Schleifen)

Schleifen erfüllen verschiedene Funktionen innerhalb eines Programms. In erster Linie berechnen sie Felder komponentenweise. Für die Entwicklung des adjungierten Codes ist es dabei wichtig zu wissen, ob dies unabhängig von anderen Komponenten

geschieht oder nicht und ob dabei Bedarfsvariablen überschrieben werden. Um dies festzustellen ist eine Schleifenanalyse durchzuführen, wie sie auch zur Transformation von Code für die Vektorisierung und Parallelisierung von Schleifen angewandt wird.

**Parallele Schleifen** Gibt es keinerlei gegenseitige Abhängigkeiten zwischen den einzelnen Schleifenläufen, so können sie parallel ausgeführt werden. In diesem Fall ist die adjungierte Schleife eine Schleife bestehend aus den adjungierten Anweisungen (Bsp. 2).

<u>Code</u>	<u>adjungierter Code</u>
DO I = low, up, step	DO I = low, up, step
Anweisung	adjungierte Anweisung
END DO	END DO

**Beispiel 2: adjungierte parallele DO-Schleife**

**Sequentielle Schleifen** Ist das Ergebnis eines Schleifendurchlaufs von einem vorhergehenden Durchlauf abhängig, dann darf die Reihenfolge nicht geändert werden. Die adjungierte Schleife muß dann die adjungierten Anweisungen in umgekehrter Reihenfolge ausführen. Dazu sind untere und obere Grenze zu vertauschen und die negative Schrittweite einzusetzen (Bsp. 3). Erreicht die Laufvariable nicht die obere

<u>Code</u>	<u>adjungierter Code</u>
DO I = low, up, step	DO I = up, low, -step
Anweisung	adjungierte Anweisung
END DO	END DO

**Beispiel 3: adjungierte DO-Schleife**

Grenze, so ist der erste Wert der Laufvariablen der adjungierten DO-Schleife (up) durch den Ausdruck

$$\text{low} + \text{step} * \text{INT}((\text{up} - \text{low})/\text{step})$$

zu ersetzen.

Besondere Beachtung verlangen Schleifen in denen Bedarfsvariablen überschrieben werden (Bsp. 4). Die Variable FAC hält das aktuelle Zwischenergebnis einer Pro-

Code

```

FAC = 1.0
DO I = 1, N
    FAC = FAC * X(I)
END DO

```

**Beispiel 4:** *Überschreiben einer Bedarfsvariablen innerhalb der Schleife (FAC und X seien aktiv)*

duktberechnung. Die adjungierte Zuweisung benötigt den Wert von FAC; FAC ist eine Bedarfsvariable. Sie bekommt aber in jedem Schleifendurchlauf einen neuen Wert, die alten Werte stehen nach Ausführung der Schleife nicht mehr zur Verfügung. Um den Wert von FAC zu erhalten, wäre es möglich, die Werte von FAC während der Ausführung des Codes wegzuschreiben und innerhalb der adjungierten Schleife entsprechend einzulesen. Dies erfordert aber sehr viele Ein- und Ausgabeanweisungen und sollte daher nur im Falle sehr großer Felder angewendet werden.

Eine andere Möglichkeit besteht darin, die Bedarfsvariable im Code um eine Dimension zu erweitern. Durch Indizierung der zusätzlichen Dimension mit der Schleifenvariablen bleiben alle Zwischenergebnisse der Bedarfsvariablen erhalten. Innerhalb der adjungierten Schleife sind dann alle benötigten Zwischenergebnisse verfügbar, sofern die neue Variable im Code herausgeschrieben oder vor der adjungierten Schleife wiederberechnet wurde (Bsp. 5).

Code

```

H(0) = 1.0
DO I = 1, N
    H(I) = H(I-1) * X(I)
END DO
FAC = H(N)

```

adjungierter Code

```

ADH(N) = ADFAC
ADFAC = 0.0
DO I = N,1,-1
    ADX(I) = ADX(I) + ADH(I)*H(I-1)
    ADH(I-1) = ADH(I-1) + ADH(I)*X(I)
    ADH(I) = 0.0
END DO
ADH(0) = 0.0

```

**Beispiel 5:** *Lösung des Konflikts durch Einführung eines Hilfsfeldes H(0:N)*

Die rechenaufwendigste Lösung ist, den aktuellen Wert der Bedarfsvariablen für



jeden adjungierten Durchlauf neu auszurechnen. Dabei bekommt die Laufvariable der inneren Schleife nur die Werte bis zum nächsten Wert der aktuellen adjungierten Laufvariablen (Bsp. 6). Diese Methode hat den Vorteil, daß sie keinen zusätzlichen Speicherplatz braucht.

Code

```
DO I = N, 1, -1
  FAC = 1.0
  DO L = 1, I-1
    FAC = FAC * X(I)
  END DO
  ADX(I) = ADX(I) + ADFAC * FAC
  ADFAC = ADFAC * X(I)
END DO
ADFAC = 0.0
```

**Beispiel 6:** Lösung eines Konflikts durch Neuberechnung des benötigten Wertes der Bedarfsvariablen FAC

### 3.4.4 Prozeduraufruf

Prozeduren, die aktive Variable berechnen, werden aktive Prozeduren genannt und müssen adjungiert werden. Die adjungierte Anweisung zu einem Aufruf einer aktiven Prozedur ist der Aufruf der adjungierten Prozedur, einschließlich etwaiger zusätzlicher Zuweisungen.

In der adjungierten Prozedur steht der entsprechende adjungierte Code. Erst nach der Entwicklung der adjungierten Prozedur steht fest, welche Variablen aus dem Code von dem aufrufenden Programm bereitgestellt werden müssen. Diese Bedarfsvariablen stehen entweder als Argument in der aktuellen Parameterliste der adjungierten Prozedur oder es sind globale Variablen in COMMON-Blöcken. Sie müssen die Werte besitzen, die sie direkt vor dem zugehörigen Aufruf im Code hatten.

In Abb. 7 ist dies anhand eines einfachen Beispiels dargestellt. Ein Strich über einer Variablen bedeutet: "dieses Argument wird innerhalb des Unterprogramms benutzt". Ein Strich unter einer Variablen signalisiert dagegen, daß dieses Argument innerhalb des Unterprogramms verändert wird, es ist ein Ergebnis des Unterprogramms.

<u>Code</u>	<u>Adjungierter Code</u>
CALL SUB ( $\bar{X}$ , $\bar{Y}$ , $\bar{A}$ )	CALL ADSUB ( $\bar{X}$ , $\bar{Y}$ , $\bar{A}$ , $\overline{ADX}$ , $\overline{ADY}$ )

**Beispiel 7:** *Prozeduraufruf und Aufruf der korrespondierenden adjungierten Prozedur (A sei eine passive Variable, X, Y seien aktive Variablen)*

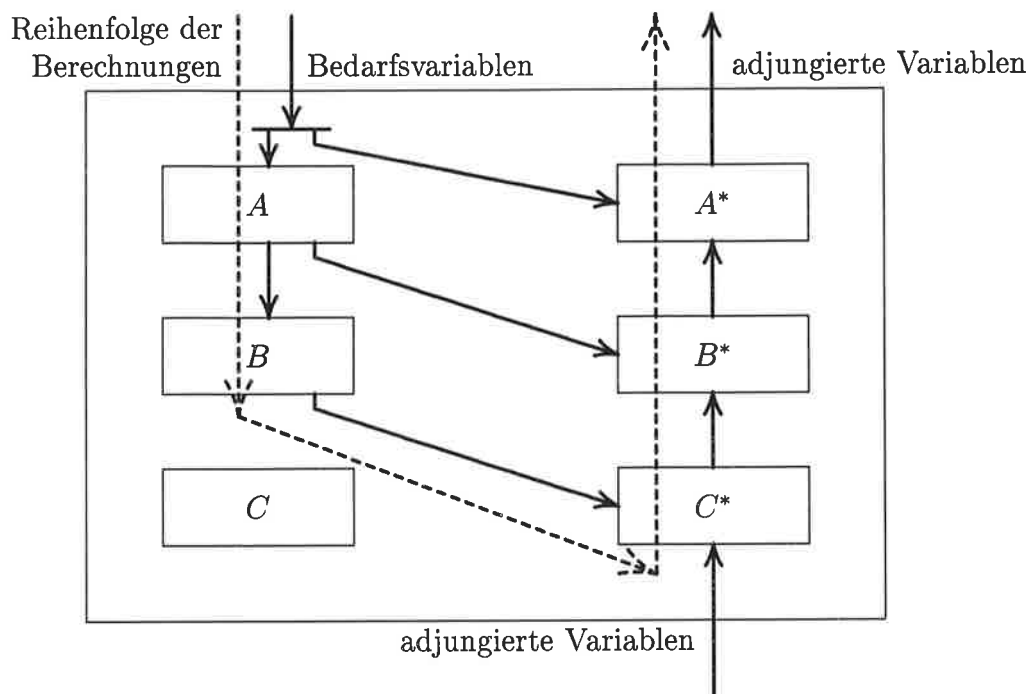
Von einer adjungierten Variablen innerhalb der Argumentenliste eines adjungierten Unterprogramms wird immer angenommen, daß sie Argument und Ergebnis des Unterprogramms ist. Selbstverständlich müssen die Typen der Argumente in der Liste des Aufrufs denjenigen des Unterprogramms entsprechen und die Anzahl an Argumenten übereinstimmen.

### 3.4.5 Adjungieren von Anweisungsfolgen

Die zu einer Anweisungsfolge adjungierte Anweisung ist die Folge der einzelnen adjungierten Anweisungen in umgekehrter Reihenfolge, entsprechend der Anwendung der Kettenregel im Rückwärtsmodus. Die adjungierten Anweisungen benötigen im allgemeinen Zwischenergebnisse. Bedarfsvariablen, die innerhalb der Sequenz berechnet werden, müssen, um das Prinzip der Modularität einzuhalten, innerhalb der adjungierten Sequenz neu berechnet werden. Dazu sind Anweisungen der ursprünglichen Sequenz den adjungierten Anweisungen voranzustellen.

Die Vorgehensweise ist anhand der symbolischen Darstellung einer Sequenz von drei Anweisungen in Abb. 7 erläutert: auf der linken Seite sind die drei Anweisungen  $A$ ,  $B$  und  $C$  gezeigt und auf der rechten Seite die entsprechenden adjungierten Anweisungen. Zunächst werden für jede Anweisung der Sequenz die eingehenden und berechneten Variablen bestimmt. Nach Aufstellung der adjungierten Anweisungen sind die jeweiligen Bedarfsvariablen bekannt (Pfeile zwischen linker und rechter Seite in Abb. 7). Zur Berechnung der Bedarfsvariablen müssen die Abschnitte der ursprünglichen Anweisungen eingefügt werden, die Bedarfsvariablen direkt berechnen oder andere Zwischenergebnisse, welche zur Berechnung von Bedarfsvariablen gebraucht werden.

Bedarfsvariablen, die außerhalb der Anweisungsfolge berechnet wurden, müssen der adjungierten Anweisungsfolge von außen bereitgestellt werden. Bedarfsvariablen, die innerhalb der Anweisungsfolge berechnet wurden, müssen entweder während der Ausführung des Codes abgespeichert und vor der adjungierten Anweisung eingelesen werden (siehe Abs. 3.6), oder aber vor der adjungierten Anweisungsfolge neu berech-



**Abbildung 7:** Schematische Darstellung einer Adjungierten Sequenz von Anweisungen ( $A, B$ , und  $C$  sind Anweisungen,  $A^*, B^*$ , und  $C^*$  sind die zugehörigen adjungierten Anweisungen). Die durchgezogenen Pfeile stellen ein- und ausgehende Variablen dar. Pfeile von links nach rechts entsprechen BedarfsvARIABLEN.

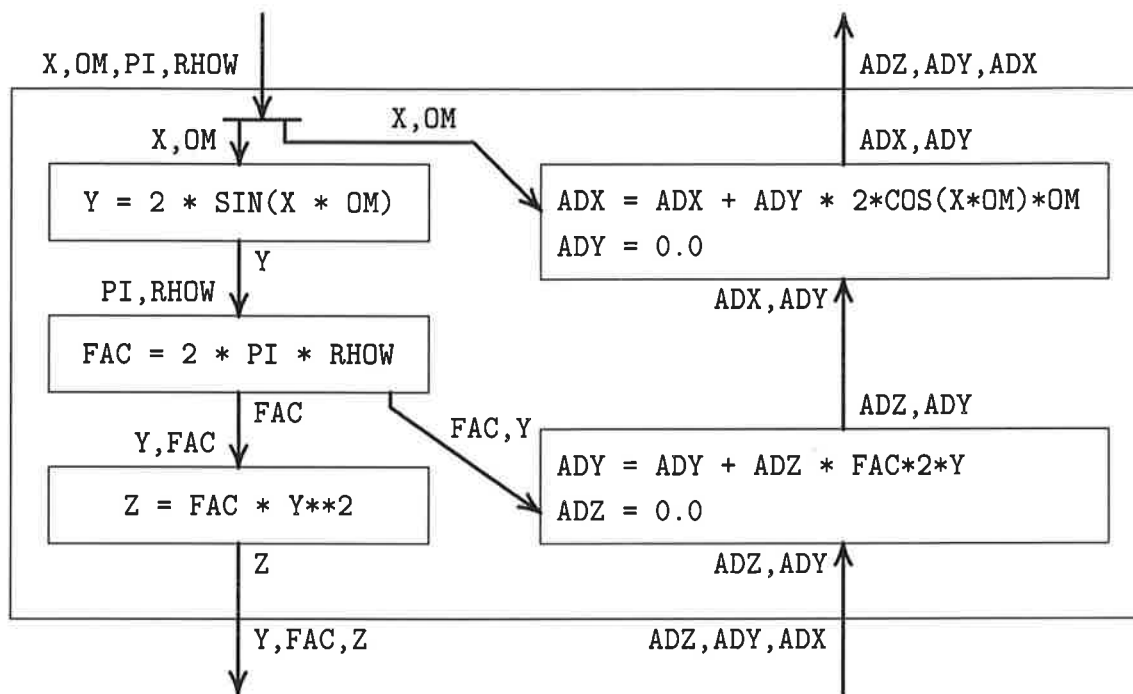
net werden (linke Seite in Abb. 7). Dabei ist zu beachten, daß die BedarfsvARIABLEN zu Beginn der adjungierten Anweisung den Wert haben, den sie im Code vor Ausführung der Anweisung hatten (Modularität, siehe Abs. 3.3.4). Die BedarfsvARIABLEN der gesamten adjungierten Anweisungsfolge sind sowohl die direkt zur Ausführung der adjungierten Anweisungen benötigten, wie auch jene, die in die Neuberechnung von Zwischenergebnissen eingehen.

Die vollständige adjungierte Anweisungsfolge (gestrichelter Pfeil in Abb. 7) enthält also eine Folge von Neuberechnungen aus dem originalen Code und eine Folge von den eigentlichen adjungierten Anweisungen. Das Schema zum Adjungieren einer Anweisungsfolge von drei Anweisungen ist dann

$$A' B' C^* B^* A^* , \quad (32)$$

wobei  $A'$  ( $B'$ ) nur aus den Anweisungen von  $A$  ( $B$ ) besteht, die zur Berechnung von BedarfsvARIABLEN erforderlich sind.

In Bsp. 8 wird die Regel auf eine einfache Folge von drei Anweisungen angewendet. Die Variablen links von einem Pfeil sind eingehende Variablen, die rechts da-



**Beispiel 8:** Adjungierte Sequenz von Anweisungen ( $X, Y$  und  $Z$  seien aktiv). Die adjungierten Anweisungen innerhalb eines Kästchens werden von oben nach unten ausgeführt.

von veränderte Variablen. Die dritte adjungierte Zuweisung (rechts unten in Bsp. 8) benötigt die Werte der Variablen  $FAC$  und  $Y$ . Sie werden von der ersten und zweiten Anweisung berechnet, daher werden diese Anweisungen der Sequenz der adjungierten Anweisungen vorangestellt (Bsp. 9). Die erste adjungierte Anweisung (links oben in

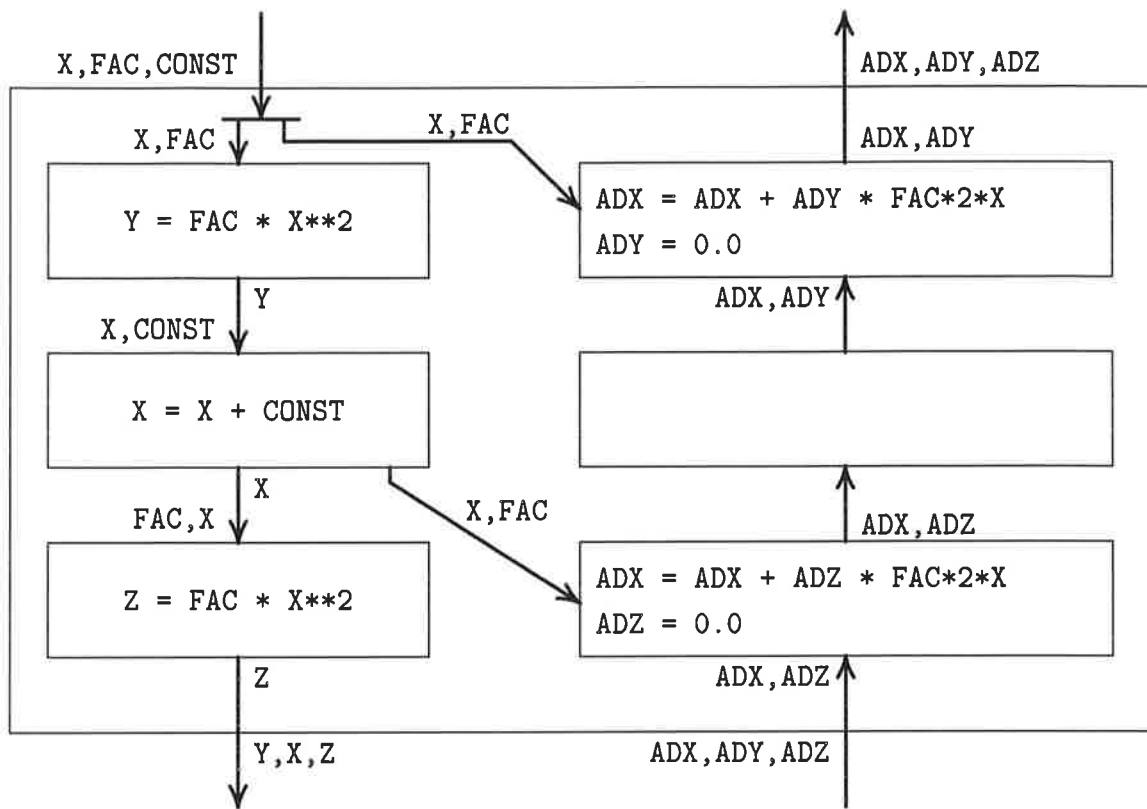
```

Y   = 2 * SIN(X * OM)
FAC = 2 * PI * RHOW
ADY = ADY + ADZ * FAC * 2 * Y
ADZ = 0.0
ADX = ADX + ADY * 2 * COS(X * OM) * OM
ADY = 0.0

```

**Beispiel 9:** Adjungierter Code zur Anweisungsfolge des Beispiels 8

Bsp. 8) benötigt die Werte der Variablen  $X$  und  $OM$ . Sie sind Eingabevariablen der ursprünglichen Sequenz und müssen daher auch der adjungierten Sequenz bereitgestellt werden. Insgesamt sind die Variablen  $X, OM, PI$  und  $RHOW$  Bedarfsvariablen der adjungierten Sequenz.



**Beispiel 10:** Adjungierte Sequenz von Anweisungen mit Konflikt durch Überschreiben ( $X, Y$  und  $Z$  seien aktiv). Die mittlere adjungierte Zuweisung bewirkt keine Veränderung der adjungierten Variablen und kann entfallen.

Die Bereitstellung von Variablen kann zu Konflikten führen. Diese treten immer dann auf, wenn eine Bedarfsvariable von zwei oder mehr adjungierten Anweisungen mit verschiedenen Werten benötigt wird. In Bsp. 10 steht auf der rechten Seite der ersten und dritten Zuweisung ein nichtlinear von  $B$  abhängender Term. Die entsprechenden adjungierten Anweisungen benötigen daher den Wert von  $B$ , dieser wird jedoch in der zweiten Zuweisung verändert.

Ähnlich den sequentiellen Schleifen gibt es auch hier mehrere Möglichkeiten sicherzustellen, daß eine Bedarfsvariable den richtigen Wert hat.

1. Der Wert wird in dem Code abgespeichert und an der entsprechenden Stelle des adjungierten Codes wieder eingelesen.
2. Es wird im adjungierten Code eine lokale Hilfsvariable eingeführt, der benötigte Wert berechnet und in der Hilfsvariablen für den späteren Gebrauch gehalten.
3. Die Variable wird immer dann neu berechnet, wenn ihr Wert gebraucht wird.

Dies setzt allerdings voraus, daß dazu alle nötigen Variablen bereitstehen. Der Code wird dann nach folgendem Schema adjungiert:

$$A'B' C^* A'' B^* A^* \quad (33)$$

Dabei müssen vor den Anweisungen  $A''$  und  $A^*$  die Anfangswerte aller veränderten und im weiteren Verlauf benötigten Variablen wiederhergestellt werden. Angewandt auf Bsp. 10 müßte die Variable B auf den Wert vor Beginn der Anweisungsfolge zurückgesetzt werden.

Die Entscheidung zwischen diesen Alternativen hängt von den Variablen und den gewünschten Eigenschaften des adjungierten Codes ab. Skalare sollten neu berechnet werden, da der zusätzliche Aufwand zum Lesen und Schreiben meist die Berechnungszeit übersteigt. Ist kurze Rechenzeit wichtiger als der Speicherbedarf des adjungierten Codes so sind Felder bevorzugt abzuspeichern. Bei dem hier verwendeten Zirkulationsmodell hat es sich als sinnvoll herausgestellt, die Speicherung von Variablen auf die Zeitintegrationsschleife zu beschränken und alle anderen Zwischenergebnisse neu zu berechnen.

### 3.5 Adjungierte Unterprogramme

Ein Unterprogramm hat Eingabe- und Ausgabe-Variablen. Diese werden entweder über eine Argumentenliste übergeben, oder es sind globale Variablen. Zunächst muß festgestellt werden, welche dieser Variablen aktiv sind. Alle lokalen Variablen, die von den aktiven Eingabevariablen abhängen und die zur Berechnung der aktiven Ausgabevariablen verwendet werden, sind aktive lokale Variablen. Sind alle aktiven Variablen innerhalb des Unterprogrammes bekannt, so kann der adjungierte Code als Sequenz von Anweisungen konstruiert werden. Die Bedarfsvariablen dieser Sequenz müssen dem adjungierten Unterprogramm von dem aufrufenden Programm bereitgestellt werden (siehe Kap. 3.4.4). Ist solch eine Variable ein Argument des Unterprogramms, so ist sie dem adjungierten Unterprogramm auch als Argument zu übergeben. Die Argumentenliste des adjungierten Unterprogrammes besteht damit aus den adjungierten aktiven Variablen sowie denjenigen Eingabevariablen der ursprünglichen Argumentenliste, die Bedarfsvariablen der adjungierten Sequenz sind.

Der Deklarationsteil des adjungierten Unterprogrammes setzt sich aus den Deklarationen der Bedarfsvariablen und der adjungierten Variablen zusammen. Dabei ist für COMMON-Blöcke, die aktive Variablen enthalten, ein adjungierter COMMON-Block zu bilden, der die zugehörigen adjungierten Variablen enthält.

Der eigentliche Code des adjungierten Programmes beginnt mit Anweisungen zur Initialisierung der lokalen adjungierten Variablen mit Null, denn zu ihnen werden immer Werte addiert (siehe Kap. 3.4.1). Darauf folgt die adjungierte Sequenz. Globale adjungierte Variablen (in COMMON-Blöcken) müssen vor Aufruf des adjungierten Modells initialisiert werden.

## 3.6 Zur Speicherung von Bedarfsvariablen

### 3.6.1 Zusätzliche Anweisungen

Soll der Wert einer Bedarfsvariablen im adjungierten Modell nicht neu berechnet werden, so muß ihr Wert während der Ausführung des Codes gespeichert werden. Dies kann durch eine zusätzliche Anweisung, etwa durch Aufruf eines entsprechenden Unterprogramms, geschehen. Im adjungierten Code wird dann, bevor der Wert benötigt wird, ein anderes Unterprogramm aufgerufen, welches diesen Wert bereitstellt. Die Werte können auf einer temporären Datei oder im Hauptspeicher gehalten werden. Die Wahl hängt vom verfügbaren Speicherplatz und der gewünschten Zugriffszeit ab. Da die Werte meist in umgekehrter Reihenfolge ihrer Berechnung angefordert werden, sollten sie in beliebiger Reihenfolge lesbar sein. Eine Datei mit direktem Zugriff auf ihre Datensätze erfüllt diese Forderung.

### 3.6.2 Stützpunkttechnik

Die Speicherung aller Bedarfsvariablen während einer Modellintegration kann einen erheblichen Platz erfordern. Es ist jedoch möglich, auf Kosten von Rechenzeit, diesen Speicherbedarf wesentlich zu reduzieren. Dazu werden nicht alle Bedarfsvariablen sondern nur die zu bestimmten Zeitpunkten des Programmablaufs benötigten abgespeichert. Alle anderen Bedarfsvariablen werden nun, von diesen Zwischenergebnissen ausgehend, neu berechnet.

In einem Programm der numerischen Simulation einer Systemtrajektorie sollten die Stützpunkte sinnvollerweise die Systemzustände in Abständen von einigen Zeitschritten sein. Aus diesen Zuständen lassen sich diejenigen zu den dazwischen liegenden Zeitpunkten durch erneute Integration einfach berechnen.

Die zusätzliche Rechenzeit für diese Stützpunkttechnik beträgt etwa soviel wie eine Berechnung der Kostenfunktion. Hat die Trajektorie  $N$  ( $N = N_1 * N_2$ ) Zeitschritte und werden  $N_1$  Stützpunkte äquidistant ausgewählt, so beträgt der Speicherbedarf

nur noch  $N_1 + N_2$  Einheiten anstatt  $N_1 * N_2$ .

Der Speicherbedarf kann auf Kosten der Rechenzeit noch weiter reduziert werden. Bei geeigneter Wahl der Stützpunkte beträgt der Speicherbedarf nur noch  $\log_2(N)$  Einheiten, der Rechenaufwand ist dann proportional zu  $\log_2(N)$  (Griewank 1992)

### 3.7 Problematische Programmstrukturen

Die Reihenfolge der Ausführung von Anweisungen beeinflusst wesentlich die Konstruktion des adjungierten Codes. Für Kontrollanweisungen die eine komplizierte Reihenfolge definieren, können daher keine einfachen Regeln aufgestellt werden. In der Programmiersprache Fortran betrifft dies hauptsächlich die Anweisungen GOTO, ENTRY und alternatives RETURN. In den meisten Fällen sind diese Anweisungen nicht essentiell und können leicht durch andere ersetzt werden, die mit den Prinzipien einer strukturierten Programmierung besser vereinbar sind.

Für die Datenabhängigkeitsanalyse ist die eindeutige Zuordnung von Werten zu Variablennamen wichtig. Diese kann in Fortran durch EQUIVALENCE-Anweisungen oder unterschiedliche Belegung von gleichnamigen COMMON-Blöcken erschwert werden.

### 3.8 Adjungieren von implizit erklärten Funktionen

Häufig wird innerhalb eines Programmes eine implizite Gleichung gelöst (z.B. bei implizitem oder semi-implizitem Zeitschrittverfahren). Dann wird für ein vorgegebenes  $p \in \mathbb{R}^m$  ein  $x \in \mathbb{R}^n$  gesucht, welches das Gleichungssystem

$$x - f(x, p) = 0 \quad (34)$$

löst. Falls  $(a, p)$  Gl. (34) genügt, die Funktion

$$\begin{aligned} f : \mathbb{R}^n \times \mathbb{R}^m &\rightarrow \mathbb{R}^n \\ (x, y) &\mapsto f(x, y) \end{aligned} \quad (35)$$

im Punkt  $(a, p)$  total differenzierbar ist und

$$1 - \frac{\partial f}{\partial x}(a, p) \quad (36)$$

invertierbar ist, kann der Satz über implizite Funktionen angewendet werden um Gl. (34) nach  $x$  aufzulösen. Dann existiert eine Umgebung  $U \subset \mathbb{R}^m$  von  $p$  und eine



im Punkte  $p$  differenzierbare Funktion  $g : U \rightarrow \mathbb{R}^n$ , so daß für alle  $\tilde{p} \in U$  auch das Paar  $(g(\tilde{p}), \tilde{p})$  Gl. (34) erfüllt. Weiterhin ist die Ableitung von  $g$  gegeben durch:

$$\frac{\partial g}{\partial y}(p) = (1 - A(g(p), p))^{-1} \cdot B(g(p), p) , \quad (37)$$

wobei  $A$  und  $B$

$$A(x, y) := \frac{\partial f}{\partial x}(x, y) \quad (38)$$

$$B(x, y) := \frac{\partial f}{\partial y}(x, y) \quad (39)$$

die Funktionalmatrizen von  $f$  bezüglich  $x$  bzw.  $y$  sind.

Für die numerische Lösung wird meist ein iteratives Verfahren mit einer die Konvergenz garantierenden Funktion  $f$  verwendet:

$$\begin{aligned} \mathbf{S1} : & \quad x_0 = \text{erste Schätzung} \\ \mathbf{S2} : & \quad x_n = f(x_{n-1}, p) \quad (n = 1, \dots, N) \\ \mathbf{S3} : & \quad x = x_N , \end{aligned} \quad (40)$$

wobei  $N$  die Anzahl der durchgeführten Iterationen  $\mathbf{S2}$  ist, um eine gegebene Genauigkeit zu erreichen. Der adjungierte Code ist wieder eine Iteration bestehend aus den entsprechenden adjungierten Anweisungen:

$$\begin{aligned} \mathbf{S3}^* : & \quad \delta^* x_N = \delta^* x \\ & \quad \delta^* x = 0 \\ & \quad \vdots \\ \mathbf{S2}^* : & \quad \delta^* x_{n-1} = A^*(x_{n-1}, p) \delta^* x_n \quad (n = 1, \dots, N) \\ & \quad \delta^* p = B^*(x_{n-1}, p) \delta^* x_n \\ & \quad \delta^* x_n = 0 \\ & \quad \vdots \\ \mathbf{S1}^* : & \quad \delta^* x_0 = 0 . \end{aligned} \quad (41)$$

Ist die Funktion  $f$  nichtlinear in  $x$ , so hängt  $A$  oder  $B$  von  $x_{n-1}$  ab, und es müssen für jede adjungierte Iteration die Zwischenergebnisse  $x_{n-1}$  der ursprünglichen Iteration bereitgestellt werden. Dazu werden die Werte der Variablen während der Iteration gespeichert und während der adjungierten Iteration gelesen. Dies kann je nach Anzahl der Iterationen einen erheblichen Speicheraufwand erfordern. Es ist daher sehr wünschenswert, die Benutzung von Zwischenwerten im adjungierten Code zu vermeiden.

<u>Code</u>	<u>Adjungierter Code</u>
X = erste Schätzung	
DO I = 1, N	DO I = N, 1, -1
speichere X	lese X
CALL FCT( X, P )	CALL ADFCT( X, P, ADX, ADP )
END DO	END DO
	ADX = 0.0

**Beispiel 11:** *Iteration zur Lösung eines impliziten Gleichungssystems und die entsprechende adjungierte Iteration (P und X sind aktive Variablen)*

Auf der linken Seite von Bsp. 11 ist eine einfache Iteration gezeigt. Die Prozedur FCT berechnet X in Abhängigkeit von P und dem alten Wert von X. Wenn etwa das entsprechende adjungierte Unterprogramm ADFCT nach den vorgestellten Regeln erstellt worden ist, dann stellt ADFCT eine Implementierung des Operators  $S2^*$  dar. In der adjungierten Iteration (rechte Seite von Bsp. 11) werden die Bedarfsvariablen vor der Ausführung des Schleifenkerns gelesen.

Unter der Voraussetzung, daß die Iteration die implizite Gleichung bis auf Maschinengenauigkeit gelöst hat, kann ein adjungierter Code konstruiert werden, der keine Zwischenergebnisse benötigt. Ausgehend von (37) verursacht eine beliebige Variation  $\delta p$  eine Variation  $\delta x$  der Lösung  $x := g(p)$ :

$$\delta x = (1 - A(x, p))^{-1} \cdot B(x, p) \delta p . \quad (42)$$

Nach Einführung von

$$\delta z := B(x, p) \delta p \quad (43)$$

und Anwendung der Gleichung (30) erhält man:

$$\delta^* z = [(1 - A(x, p))^{-1}]^* \delta^* x . \quad (44)$$

Die Gleichung legt für einen bekannten Einfluß  $\delta^* x$  den Einfluß  $\delta^* z$  fest. Sie kann auch implizit formuliert werden:

$$\delta^* z = A(x, p)^* \delta^* z + \delta^* x . \quad (45)$$

Auch diese implizite Gleichung kann durch Iteration gelöst werden:

$$\begin{aligned}\delta^* z_N &= \delta^* x \\ \delta^* z_{n-1} &= A^*(x, p) \delta^* z_n + \delta^* x \quad (n = 1, \dots, N) \\ \delta^* z_n &= 0 \\ \delta^* z &= \delta^* z_0 .\end{aligned}$$

Christianson (1993) hat gezeigt, daß diese 'adjungierte Iteration' dasselbe Konvergenzverhalten zeigt wie die ursprüngliche Iteration. Mittels der Lösung  $\delta^* z$  und der adjungierten Gleichung zu (43):

$$\delta^* p = B^*(x, p) \delta^* z \quad (46)$$

kann nun die adjungierte Variable  $\delta^* p$  berechnet werden. Die Operatoren  $A^*$  und  $B^*$  hängen nicht mehr von den Zwischenergebnissen  $x_n$  ab. Es müssen daher für den adjungierten Code nur die Lösung  $x$  der Iteration und  $p$  bereitgestellt werden.

Die Ausführung von (46) außerhalb der Iterationsschleife bewirkt gegenüber dem ersten adjungierten Code außerdem einen Rechenzeitvorteil, besonders bei aufwendiger Berechnung von  $B^* \delta^* z$ . Doch die Erstellung von adjungiertem Code nach diesem Verfahren ist etwas kompliziert, da der ursprüngliche adjungierte Code aufgeteilt werden muß. Der erste Teil ( $A^*$ ) steht weiterhin innerhalb der Iteration, der Zweite ( $B^*$ ) folgt der Iterationsschleife. Es soll gezeigt werden, wie durch minimale Änderungen des ursprünglichen Algorithmus, d.h. ohne Aufteilung des adjungierten Codes, adjungierter Code nach dem neuen Verfahren konstruiert wird.

Als erstes wird die Zuweisung an  $\delta^* z_{n-1}$  in zwei Zuweisungen aufgespalten und dazu eine Hilfsvariable  $\delta^* z'$  eingeführt. Nach Einbeziehung der Berechnung von  $B^* \delta^* z_n$  in die Schleife lautet der Algorithmus dann:

$$\begin{aligned}\delta^* z_N &= \delta^* x \\ \delta^* z'_{n-1} &= A^* \delta^* z_n \quad (n = 1, \dots, N) \\ \delta^* p &= B^* \delta^* z_n \\ \delta^* z_n &= 0 \\ \delta^* z_{n-1} &= \delta^* z'_{n-1} + \delta^* x \\ \delta^* z &= \delta^* z_0 .\end{aligned}$$

Nun wirkt hier der adjungierte Operator  $S2^*$  der Iteration (41), auf  $\delta^*z_n$  anstatt auf  $\delta^*x_n$ , daher kann auf die Implementation dieses Operators ADFCT zurückgegriffen werden. Durch Vertauschen der Variablen ADX und ADZ wird der adjungierte Code auf der rechten Seite in Bsp. 12 erhalten:

<u>Code</u>	<u>Adjungierter Code</u>
X = erste Schätzung	ADZ = ADX
DO I = 1, ITER	DO I = 1, ITER
	ADP = 0.0
CALL FCT( X, P )	CALL ADFCT( X, P, ADX, ADP )
	ADX = ADX + ADZ
END DO	END DO
	ADX = 0.0

**Beispiel 12:** *Iterative Lösung einer impliziten Gleichung und der neue dazu adjungierte Code (P und X seien aktive Variablen)*

Die adjungierte Hilfsvariable ADZ wird vor der Iteration mit dem Wert der adjungierten Variablen ADX belegt. Während der Iteration bleibt diese Hilfsvariable unverändert. Die adjungierte Variable ADP wird innerhalb der adjungierten Schleife mit Null initialisiert. Dadurch werden alle Veränderungen von ADP in den vorangehenden Schleifendurchläufen ignoriert und nur der letzte Schleifendurchlauf legt ADP fest. So wird derselbe Effekt erreicht, wie durch die Berechnung von  $B^*\delta^*z$  nach der adjungierten Schleife.

## 4 Modellbeschreibung

### 4.1 Modellgleichungen

Das Modell der großskaligen geostrophischen Zirkulation (Large Scale Geostrophic, LSG) beruht auf einem Konzept von Hasselmann (1982) und wurde von Maier-Reimer entwickelt (Maier-Reimer, Mikolajewicz und Hasselmann 1993).

Die charakteristischen räumlichen Muster der großskaligen Zirkulation des Ozeans sind sehr viel größer als der interne Rossbyradius. Die typische Zeitskala dieser Zirkulation ist groß im Vergleich zu der Laufzeit von Schwerewellen und barotropen (häufig extern genannten) Rossbywellen. Daher bestimmt im wesentlichen die Dichteschichtung die baroklinen Geschwindigkeiten unter der Annahme lokaler Geostrophie. Unter Ausnutzung dieser Eigenschaften kann mit einem sehr großen Zeitschritt (im Vergleich zu Modellen basierend auf den primitiven Gleichungen) integriert werden.

Das LSG-Modell basiert auf diesen Näherungen. Die schnellen Wellen werden durch ein implizites Schema für die zeitliche Integration gedämpft. Das Modell löst die primitiven Gleichungen einer viskosen inkompressiblen Flüssigkeit unter Anwendung der Boussinesq- und hydrostatischen Approximation und unter Vernachlässigung der Impulsadvektion.

Die Impulserhaltung lautet dann in sphärischen Koordinaten  $(\lambda, \varphi, z)$ :

$$\begin{aligned} \frac{\partial u}{\partial t} - 2v\Omega \sin \varphi + \frac{1}{\rho_0 R \cos \varphi} \frac{\partial p}{\partial \lambda} &= \frac{\mathcal{T}^\lambda}{\rho_0} + A_h \Delta_h u \\ \frac{\partial v}{\partial t} + 2u\Omega \sin \varphi + \frac{1}{\rho_0 R} \frac{\partial p}{\partial \varphi} &= \frac{\mathcal{T}^\varphi}{\rho_0} + A_h \Delta_h v, \end{aligned} \quad (47)$$

wobei  $(u, v, w)$  der Geschwindigkeitsvektor,  $\rho_0$  eine Referenzdichte,  $\Omega$  die Winkelgeschwindigkeit und  $R$  der Radius der Erde ist. Der Windschub der Atmosphäre  $\mathcal{T}$  ist die Impulsquelle an der Ozeanoberfläche. Die horizontale Eddy-Viskosität  $A_h$  beträgt  $3 \cdot 10^5 \frac{m^2}{s}$  und  $\Delta_h$  ist der horizontale Laplace-Operator. Die Dichte des Wassers  $\rho$  wird mithilfe der UNESCO-Formel (1983) aus Salzgehalt  $S$ , in-situ-Temperatur  $T$  und Druck  $p$  bestimmt:

$$\rho = \rho(S, T, p) \quad (48)$$

$$T = T(\vartheta, p), \quad (49)$$

dabei hängt die in-situ-Temperatur  $T$  von der potentiellen Temperatur  $\vartheta$  entsprechend einer Linearisierung von Bryden's Formel (Bryden 1973) in  $4km$  Tiefe und bei

2°C ab:

$$T = \vartheta + 0.12 \frac{K}{km} z . \quad (50)$$

Die Erhaltung von potentieller Temperatur und Salzgehalt lautet:

$$\begin{aligned} \frac{\partial \vartheta}{\partial t} + u \frac{1}{R \cos \varphi} \frac{\partial \vartheta}{\partial \lambda} + v \frac{1}{R} \frac{\partial \vartheta}{\partial \varphi} + w \frac{\partial \vartheta}{\partial z} &= q_{\vartheta} + K_h \Delta_h \vartheta \\ \frac{\partial S}{\partial t} + u \frac{1}{R \cos \varphi} \frac{\partial S}{\partial \lambda} + v \frac{1}{R} \frac{\partial S}{\partial \varphi} + w \frac{\partial S}{\partial z} &= q_S + K_h \Delta_h S . \end{aligned} \quad (51)$$

Die Quellen  $q_{\vartheta}$ ,  $q_S$  beschreiben wiederum den Austausch mit der Atmosphäre. Die horizontale Diffusion beträgt  $K_h = 200 \frac{m^2}{s}$ . Das Diskretisierungsschema verursacht eine größere Vertikaldiffusion als Beobachtungen ergeben. Daher wird auf eine explizite Formulierung von Diffusion verzichtet. Die numerische Implementation der Konvektion ist im nächsten Abschnitt beschrieben.

Die Massenerhaltung und die hydrostatische Beziehung schließen das Gleichungssystem:

$$\frac{1}{R \cos \varphi} \frac{\partial u}{\partial \lambda} + \frac{1}{R \cos \varphi} \frac{\partial (v \cos \varphi)}{\partial \varphi} + \frac{\partial w}{\partial z} = 0 \quad (52)$$

$$p = g \rho_0 \left( \zeta + \frac{1}{\rho_0} \int_z^0 \rho(z') dz' \right) , \quad (53)$$

dabei ist  $\zeta$  die Auslenkung der Oberfläche und  $g$  die Erdbeschleunigung.

Da der thermohaline Fluß an der Oberfläche durch Eis wesentlich beeinflusst wird, verlangt die realistische Simulation der großskaligen Zirkulation die Berücksichtigung der Eisbedeckung. In dem eingebundenen einfachen Eismodell ist der Wärmefluß proportional zur Temperaturdifferenz zwischen Atmosphäre und Ozean und umgekehrt proportional zur Eisdicke. Der Wärmefluß kann zur Produktion oder Abbau von Eis führen.

## 4.2 Diskretisierung

Das Modell ist horizontal auf einem Arakawa-E-Gitter (Arakawa 1966) mit einer effektiven Auflösung von  $3.5 * 3.5^\circ$  diskretisiert. Der Äquator liegt dabei zwischen zwei Gitterlinien. Das Modell hat 11 Schichten, deren Zentren in den Tiefen 25, 75, 150, 250, 450, 700, 1000, 2000, 3000, 4000 und 5000m liegen. Die vertikalen Geschwindigkeiten werden an den Schichtgrenzen berechnet.

Sobald lokal instabile Schichtung auftritt, findet ein vollständiger Austausch der Wassermassen statt. Sind die beteiligten Gitterzellen unterschiedlich groß, so wird der

übersteigende Beitrag zugemischt. Die Stabilität wird nacheinander für alle Schichten des Modells überprüft, so kann innerhalb eines Zeitschritts Oberflächenwasser bis zum Boden gelangen. Diese Formulierung der Konvektion ist nicht stetig differenzierbar. Die Linearisierung berücksichtigt daher nicht die Sensitivität gegenüber einer Veränderung des Konvektionsmusters.

Die implizite zeitliche Diskretisierung des Modells ermöglicht einen Zeitschritt  $\Delta t$  von einem Monat. Dadurch werden die schnellen Schwerewellen mit einem Faktor  $1 + (\Delta t \Omega \sin \varphi)^2$  gedämpft. Außerhalb eines schmalen äquatorialen Bandes ist die geostrophische Balance daher weitgehend erfüllt.

Die horizontalen Geschwindigkeiten werden für die implizite Formulierung in den barotropen und die baroklinen Anteile zerlegt. Aus der Impulserhaltung folgen dann elliptische Gleichungssysteme. Das barokline System wird iterativ, durch punktweise Unterrelaxation, gelöst. Wegen der dämpfenden Eigenschaften des Verfahrens ist nur eine geringe Genauigkeit der Lösung erforderlich. Der Abbruch nach bereits 20 Iterationen spart hierbei Rechenzeit. Die Integration des adjungierten Modells verlangt jedoch die exakte Lösung des baroklinen Gleichungssystems (siehe Abs. 3.8), erst etwa 100 Iterationen erfüllen diese Forderung. Das Barotrope System wird durch direkte Gaußsche Elimination gelöst. Weitere Details des Modells sind in (Maier-Reimer et al. 1993) beschrieben.

### 4.3 Randbedingungen

Zur Erzeugung eines saisonal stationären Zustandes wird das Modell, ausgehend von einem homogenen Ozean der Temperatur  $\vartheta = 2.5^\circ$  und dem Salzgehalt von  $S = 34.7psu$ , über einen Zeitraum von 10.000 Jahren integriert. Die thermohalinen Randbedingungen an der Meeresoberfläche werden durch eine Newton-Kopplung bestimmt. Zur Relaxation werden die Jahresmittelwerte des Oberflächen-Salzgehalts bzw. die monatlichen atmosphärischen Temperaturen des COADS-Datensatzes (Comprehensive Ocean Atmosphere Data Set, Woodruff, Sluts, Jenne und P.M.Steurer (1987)) herangezogen. Der klimatologische Windschub (Hellermann und Rosenstein 1983) trieb das Modell an der Oberfläche an. Für die seitlichen Ränder gilt eine Haftbedingung (no-slip). Aus den letzten 500 Jahren dieser Integration wurde der mittlere klimatologische Frischwasser- und Wärmefluß ( $F_S$  bzw.  $F_\vartheta$ ) berechnet.

Da im folgenden Abweichungen von der Klimatologie betrachtet werden sollen, wird eine Mischung aus Newton-Kopplung und Fluß-Randbedingung verwendet. Sie

ermöglicht stärkere Anomalien, da eine reine Newton-Kopplung Störungen stark dämpft. In den Tropen, d.h. zwischen  $30^\circ S$  und  $30^\circ N$ , wird der mittlere Fluß vorgeschrieben. Nördlich von  $40^\circ N$  und südlich von  $40^\circ S$  wirkt die Newton-Kopplung. In den beiden Bändern zwischen  $30^\circ$  und  $40^\circ$  findet ein linearer Übergang von der einen Randbedingung zur anderen statt. Formal kann dies mit einer entsprechenden Gewichtungsfunktion  $g(\varphi)$  beschrieben werden:

$$\vartheta_{neu} = \vartheta_{alt} + \Delta t \left( g(\varphi) \frac{1}{\tau_\vartheta} (\vartheta_{neu} - \vartheta_{COADS}) + (1 - g(\varphi)) F_\vartheta \right) \quad (54)$$

$$S_{neu} = S_{alt} + \Delta t \left( g(\varphi) \frac{1}{\tau_S} (S_{neu} - S_{COADS}) + (1 - g(\varphi)) F_S \right) . \quad (55)$$

Die Relaxationszeit der Newton-Kopplung beträgt für die Temperatur

$$\tau_\vartheta = 40 \frac{W}{m^2 K} \frac{1}{\rho c_p \Delta z} \quad (56)$$

und für die Salinität

$$\tau_S = 1.5 * 10^{-5} \frac{m}{s} \frac{1}{\Delta z} , \quad (57)$$

wobei  $c_p$  die Wärmekapazität des Wassers und  $\Delta z$  die Dicke der obersten Modellschicht ist.

#### 4.4 Implementierung

Das adjungierte Modell wurde nach dem im vorangehenden Kapitel beschriebenen Verfahren anhand des numerischen Programms des Modells erstellt.

Die prognostischen Variablen  $\vartheta, S, u, v, \zeta$  und einige diagnostische Zwischenergebnisse werden während der Integration des Modells in eine Datei geschrieben. Für die Integration des adjungierten Modells werden diese Variablen zu den entsprechenden Zeiten wieder eingelesen. Der Verzicht auf zeitraubende Neuberechnungen reduziert die Rechenzeit des adjungierten Modells. Tatsächlich wird ein Faktor von 1.4 gegenüber der Rechenzeit des Modells erreicht. Damit beträgt die Rechenzeit zur Auswertung von Kostenfunktion und Gradient etwa das 2.4-fache der für die Kostenfunktion erforderlichen.



## 5 Experimente

Mit dem Modell wurden zunächst Zwillingsexperimente durchgeführt. Hierbei werden die Daten von dem Modell selber erzeugt. Dies gewährleistet Konsistenz zwischen Daten und Modell. Die Daten können vom Modell rekonstruiert werden, da die sicherlich existierenden Modellfehler so auch in den Daten vorhanden sind. Wesentliche Probleme mit 'echten' Daten werden somit vermieden und es kann die Frage untersucht werden, inwieweit das gestellte Problem überhaupt lösbar ist. Ist die Antwort negativ, so erübrigen sich Anwendungen mit realistischen Daten.

Im zweiten Teil kommen reale Oberflächendaten zum Einsatz. Auch hier werden verschiedene Kombinationen von Datensätzen durch Variation des Windschubes assimiliert.

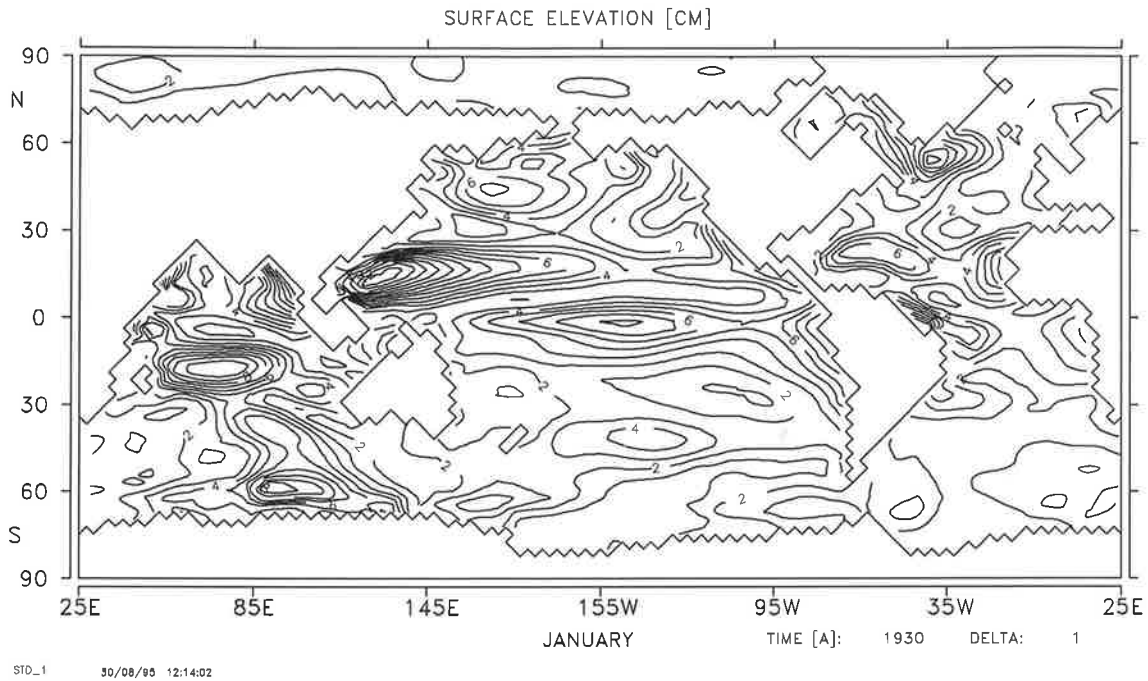
Alle Experimente optimieren Randbedingungen durch Assimilierung von Oberflächendaten. Der Anfangszustand des Modells bleibt unverändert und die Kostenfunktion enthält keinen Term (12) zur Bestrafung von Variationen des Anfangszustandes. Daher kann auf die Formulierung einer Modellfehlerkovarianz verzichtet werden.

### 5.1 Zwillingsexperimente

Aus den Anomalien der monatlichen Windschubdaten (Hellermann und Rosenstein 1983) wurden die führenden vier Empirischen Orthogonalen Funktionen (EOF) separat für die zonale und meridionale Komponente bestimmt. Eine Linearkombination aus diesen EOFs wird zu dem klimatologischen Windschub addiert. Die EOFs werden dabei mit zwei Faktoren gewichtet. Der erste Faktor ist das Betragsmaximum der jeweiligen EOF-Amplitudenzeitserien (principal components). Der zweite Faktor  $a_k$  ( $k = 1, \dots, 8$ ) wird durch  $k$  autoregressive Prozesse ausgewürfelt:

$$\begin{aligned} a_k(t_1) &= 5X \\ a_k(t_i) &= \frac{9}{10} a_k(t_{i-1}) + \frac{1}{2} X \quad (i = 2, \dots, s), \end{aligned} \quad (58)$$

dabei ist  $X$  eine gleichverteilte Zufallsvariable mit Werten zwischen  $-1$  und  $1$ , Mittelwert  $0$  und Varianz  $\frac{1}{3}$ . Dadurch ist dieser Faktor auf das Intervall  $[-5, 5]$  eingeschränkt und nur kleine Veränderungen ( $\pm \frac{1}{2}$ ) zwischen aufeinanderfolgenden Zeitpunkten sind zugelassen. Somit weichen die Amplituden nicht übermäßig weit von denjenigen der EOF-Amplitudenzeitserien ab und der Windschub sowie seine zeitliche Variation verbleiben in einem realistischen Bereich.

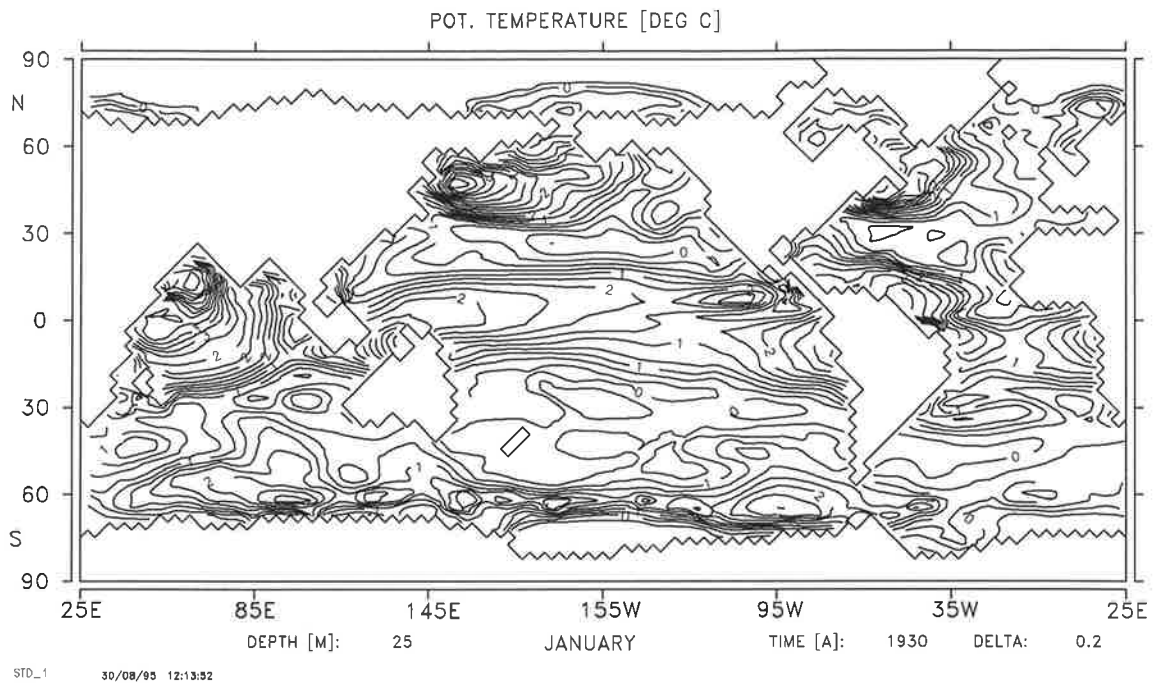


**Abbildung 8:** Varianz der Oberflächenauslenkung durch anomalen Windschub

Dieser Windschub treibt das Modell über  $s = 24$  Zeitschritte (2 Jahre) an. Die Modelltrajektorie hängt somit von  $n = 192$  Kontrollvariablen, den Amplituden der EOFs, ab. Das Modell reagiert auf die Windschubanomalien mit leichten Abweichungen von der Klimatologie. So beträgt die global gemittelte Varianz der Oberflächenauslenkung (sea surface height, SSH) etwa  $3\text{cm}$  mit Maximalwerten von  $14\text{cm}$  im westlichen Pazifik (Abb. 8). Die Varianz der Oberflächentemperatur (sea surface temperature, SST) hat eine andere Struktur. Global gemittelt beträgt die Varianz etwa  $1\text{K}$  (Abb. 9). Die Maxima liegen hauptsächlich an den nördlichen Rändern der westlichen Randströme und betragen bis über  $4\text{K}$ . Veränderungen im Windschub beeinflussen die Stärke der Oberflächenströmungen und somit hauptsächlich die SST in Gebieten, denen Wärme zugeführt oder entzogen wird. Die Zeitserien von SSH- und SST-Anomalien (SSHA, SSTA) sind die simulierten Daten für die im folgenden beschriebenen sechs Zwillingsexperimente.

Zur Rekonstruktion der Amplituden werden entweder die SSHA oder die SSTA oder beide Datensätze verwandt. Die Kostenfunktion enthält im letzten Fall folgende Terme:

$$J = \frac{1}{2} \sum_{\text{Gitter}} \left( \frac{\text{SSHA} - \text{SSHA}_{\text{Daten}}}{\sigma_{\text{SSHA}}} \right)^2 + \frac{1}{2} \sum_{\text{Gitter}} \left( \frac{\text{SSTA} - \text{SSTA}_{\text{Daten}}}{\sigma_{\text{SSTA}}} \right)^2$$



**Abbildung9:** Varianz der Oberflächentemperatur durch anomalen Windschub

$$+ \frac{1}{2} \sum_{k=1}^8 \sum_{i=2}^{n-1} \left( \frac{a_k(t_{i+1}) - 2a_k(t_i) + a_k(t_{i-1}))}{\sigma_{glatt}} \right)^2. \quad (59)$$

Der dritte Term in der Kostenfunktion bestraft hochfrequente Anteile in den Amplitudenzeitserien. Veränderungen der Kontrollvariablen in dieser Richtung sind nicht gut bestimmt, die Wirkung eines zu starken Windschubs kann durch einen schwächeren Windschub im nächsten Zeitschritt für alle folgenden Zustände weitgehend kompensiert werden, so daß der Einfluß einer  $2\Delta t$  Welle auf die Kostenfunktion gering ist. Für diesen Glattheitsterm wurde eine Gewichtung von  $\sigma_{glatt} = 10^{-1}$  gewählt.

Dieser zusätzliche Term sei für eine Zeitserie von Amplituden als Funktion betrachtet:

$$J_k := J_k(a_k(t_1), \dots, a_k(t_n)) \quad (60)$$

Die Hessesche Matrix des Terms hat nur in der Diagonalen und den ersten beiden

unteren und oberen Nebendiagonalen von Null verschiedene Einträge:

$$\left( \frac{\partial J_k}{\partial a_k(t_i) a_k(t_j)} \right) = \begin{pmatrix} 1 & -2 & 1 & & & & & & & \\ -2 & 5 & -4 & 1 & & & & & & 0 \\ 1 & -4 & 6 & -4 & 1 & & & & & \\ & 1 & -4 & 6 & -4 & 1 & & & & \\ & & & & \ddots & & & & & \\ & & & & & 1 & -4 & 6 & -4 & 1 \\ & & & & & & 1 & -4 & 6 & -4 & 1 \\ 0 & & & & & & & 1 & -4 & 5 & -2 \\ & & & & & & & & 1 & -2 & 1 \end{pmatrix}. \quad (61)$$

Da sich die Wirkung einer Windschubvariation nur langsam mit ihrer Wirkungszeit ändert, sind die Werte der Hessesche Matrix des gesamten Systems in den beiden Nebendiagonalen meist positiv und in der gleichen Größenordnung wie die in der Diagonalen. Der Glattheitsterm bewirkt eine Verstärkung der Diagonaldominanz, weil die Werte in der ersten Nebendiagonalen seiner Hesseschen Matrix negativ sind. Dies wiederum verbessert die Kondition der Hesseschen Matrix des Gesamtsystems und erhöht damit die Konvergenz der Minimierung. Für eine detaillierte Diskussion sei auf Thacker (1989) verwiesen.

In den ersten drei Experimenten (Exp. 1-3 in Tab. 1) werden die exakten simulierten Daten assimiliert. Um die Terme in der Kostenfunktion gegeneinander zu gewichten wird  $\sigma_{SSHA} = 5cm$  bzw.  $\sigma_{SSTA} = 1K$  gesetzt. Die gewichtete totale RMS- (root mean square) Differenz wurde aus den einzelnen RMS-Differenzen durch Normierung mit diesen Fehlern und der Anzahl an Meßwerten erhalten (siehe Tab. 1). Ein Wert von 1 zeigt an, daß im Mittel die Datenabweichung dem erwarteten Fehler entspricht. Die jeweilige Datenabweichung für die rekonstruierten Amplituden liegt weit unterhalb der angenommenen Fehler. Sie verschwindet nicht, weil der Glattheitsterm eine vollständige Rekonstruktion verhindert. Da die Datenabweichung klein gegenüber dem Glattheitsterm ist, hat letzterer eine starke Wirkung. Dies ist deutlich in den rekonstruierten Amplituden zu sehen (Abb. 10). Für alle Komponenten der EOFs wird der niederfrequente Anteil sehr gut wiedergegeben und der hochfrequente Anteil unterdrückt. Das gefundene Minimum nach Assimilierung der SSHA-Daten hat eine kleinere verbleibende Diskrepanz als jenes nach Assimilierung der SSTA-Daten. Dies deutet darauf hin, daß die Amplituden durch die SSHA-Verteilungen stärker bestimmt sind als durch die SST-Anomalien. Dies ist zu erwarten, da die Auslenkung der Oberfläche die Dichteschichtung des Ozeans widerspiegelt. Die SST-Anomalie wird nur indirekt, hauptsächlich durch die Divergenz des Ekmantransportes, von

Experiment	0	1	2	3	4	5	6
Ass. von SSHA		+	-	+	-	-	-
Ass. von SSTA		-	+	+	-	-	-
Ass. von fehlerbehafteter SSHA		-	-	-	+	-	+
Ass. von fehlerbehafteter SSTA		-	-	-	-	+	+
RMS SSHA [cm]	1.5	0.17	0.47	0.16	5.0	5.0	5.0
RMS SSTA [celsius]	0.25	0.14	0.25	0.13	0.57	0.56	0.56
gewichteter totaler RMS		0.10	0.20	0.09	1.07	1.06	1.06
Beitrag des Glattheitsterms zur Kostenfunktion [Prozent]		25	3	7	0.1	0.7	0.5
Iterationen		21	20	20	9	21	15
Funktionsberechnungen		22	37	33	20	25	24

**Tabelle1:** Ergebnisse der Zwillingsexperimente. Experiment 0 beschreibt die Resultate ohne Assimilierung.

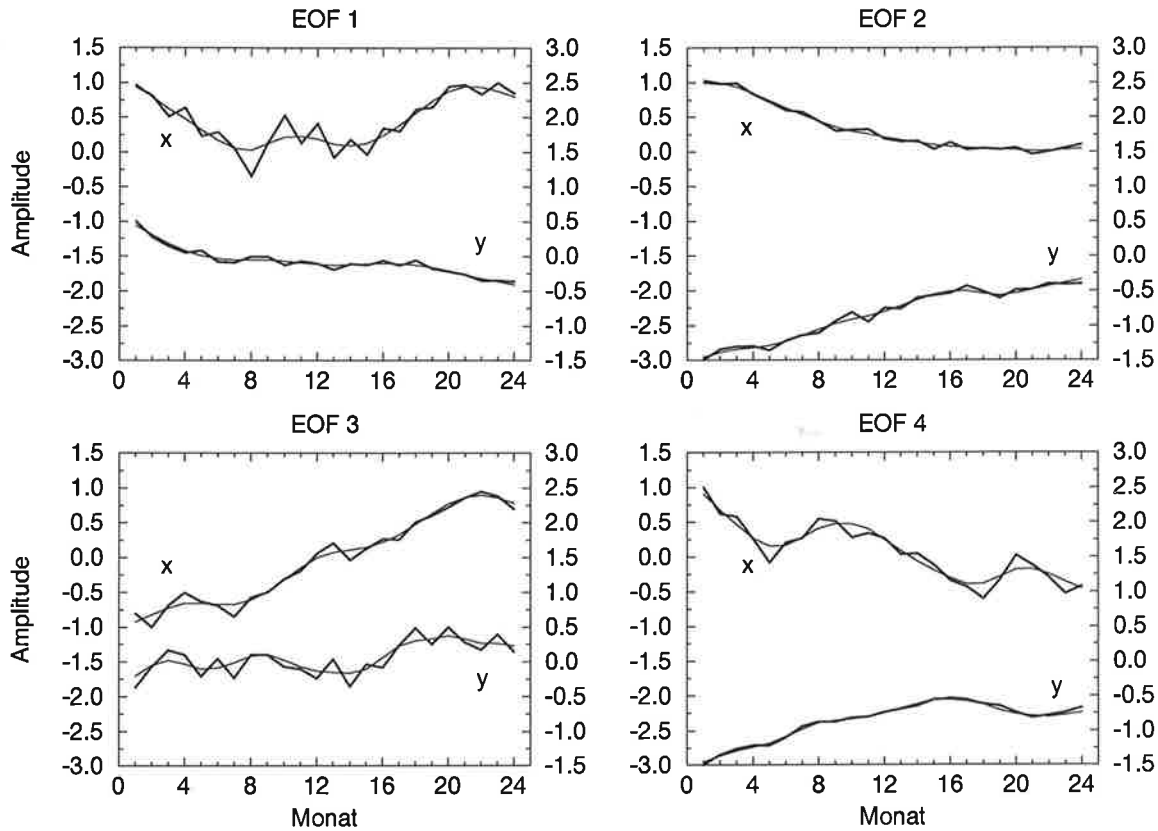
dem Windschub beeinflusst. Dennoch besitzen die SSTA-Daten eine unabhängige Information, die eine verbesserte Rekonstruktion der Amplituden erlaubt.

Die drei Experimente werden mit künstlich verrauschten Daten wiederholt (Exp. 3-6). Das weiße Rauschen ist räumlich und zeitlich unkorreliert und hat eine Amplitude von  $\sigma_{SSHA} = 5cm$  bzw.  $\sigma_{SSTA} = 0.5K$ , was in etwa dem erwarteten Fehler von Satellitenmessungen (Altimeter- und Infrarot-Messungen AVHRR) entspricht.

Für alle drei Experimente erreicht die verbleibende Datenabweichung den simulierten Fehler der Daten (Tab. 1). Da er größer ist als in den ersten Experimenten ist das relative Gewicht des Glattheitsterms in der Kostenfunktion kleiner. Die großskaligen Strukturen der ursprünglichen Amplituden werden im wesentlichen rekonstruiert (Abb. 11). Die größten Differenzen bestehen im letzten Drittel des Zeitintervalls und in den Amplituden der EOFs 3 und 4.

Eine wichtige Frage ist, inwieweit die einzelnen Kontrollvariablen durch die Daten bestimmt sind. Die Formulierung der Kostenfunktion wurde entsprechend einer Normalverteilung der Meßwerte vorgenommen. Da das Modell nur schwach nichtlinear ist, ist anzunehmen, daß auch die Verteilung der Kontrollvariablen nur wenig von einer Normalverteilung abweicht. Dann ist die Fehlerkovarianz der Kontrollvariablen die Inverse der Hesseschen Matrix am Minimum.

Die Hessesche Matrix der Kostenfunktion am Minimum wurde durch Differen-



**Abbildung 10:** Rekonstruierte Amplituden (dünne Linien) des Experiments 3 im Vergleich zu den Amplituden mit denen die Daten erzeugt wurden (dicke Linien). Die vier Bilder enthalten die Amplituden jeweils einer zonalen und meridionalen EOF. Als Datensätze für die Assimilierung waren die SSH- und SST-Anomalien gegeben. Von den meridionalen Amplituden ( $y$ , rechte Skala) wurde konstant 2.0 abgezogen um sie unter den zonalen Amplituden ( $x$ , linke Skala) im gleichen Bild darzustellen.

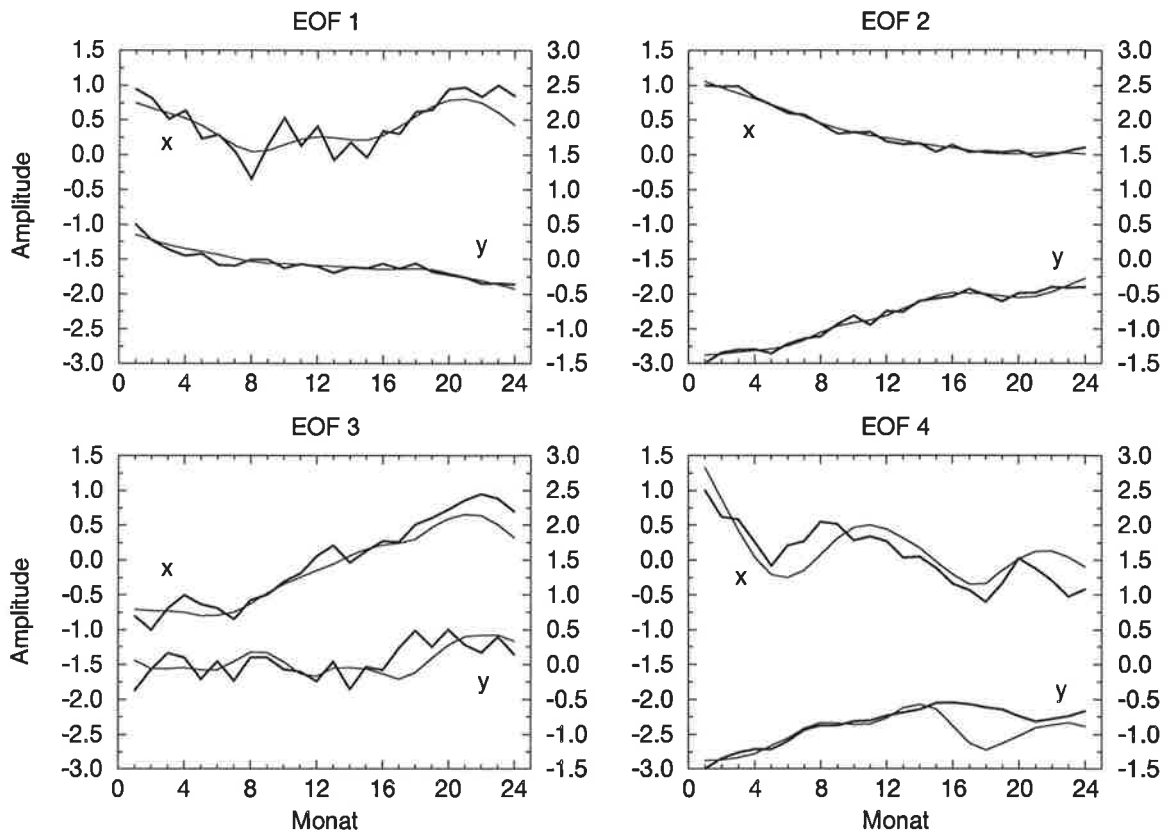
zenquotienten der Gradienten bestimmt:

$$\left(\nabla^2 J\right)_i = \frac{\nabla_{x+\epsilon x_i} J - \nabla_x J}{\epsilon} \quad (i = 1, \dots, n), \quad (62)$$

Da dies  $n + 1$  Berechnungen der Komponenten des Gradienten erfordert, ist der Rechenaufwand sehr groß. Daher wurde die symmetrische Hessesche Matrix nur für Experiment 6 (mit und ohne Glattheitsbedingung) berechnet und eine Hauptachsen-  
transformation durchgeführt:

$$\nabla^2 J = V D V^t. \quad (63)$$

$V$  ist eine  $n \times n$  Matrix, deren Spalten aus den Eigenvektoren  $v_i$  besteht. Die Einträge der Diagonalmatrix  $D$  sind die Eigenwerte  $\lambda_i$ .



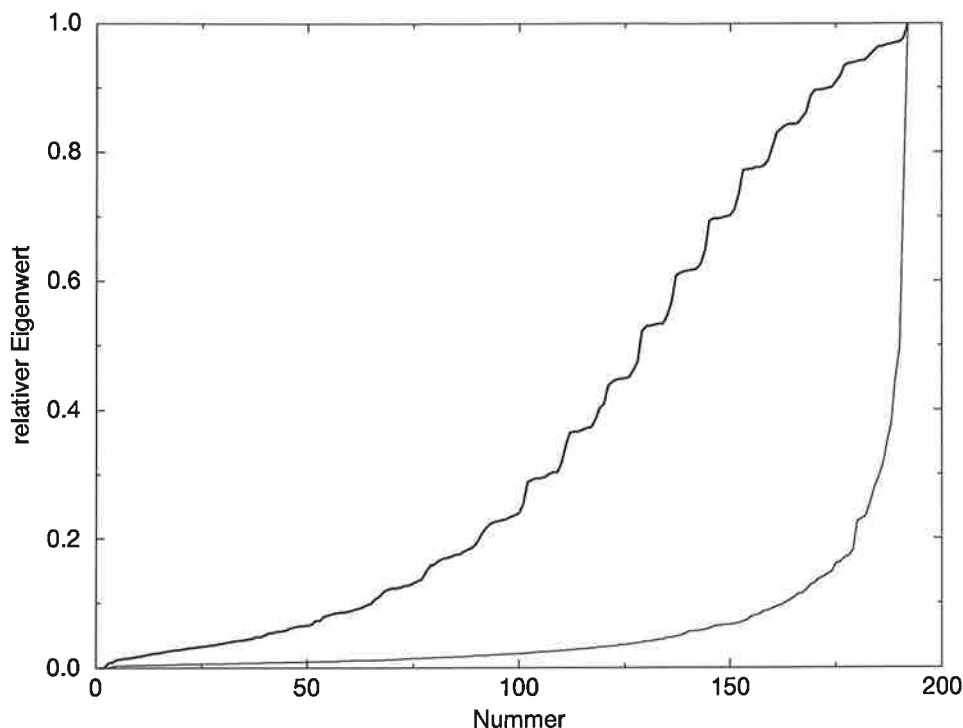
**Abbildung 11:** Rekonstruierte Amplituden wie in Abb. 10 für Experiment 6. Als Datensätze für die Assimilierung waren die mit einem Rauschen versehenden SSH- und SST-Anomalien gegeben.

Das Eigenwertspektrum zeigt Häufungen von jeweils 8 vergleichbaren Eigenwerten (Abb. 12). Sie werden durch die Glattheitsbedingung verursacht, denn sie ist unabhängig für die 8 Amplituden formuliert. Fehlt diese Bedingung, so verschwinden die Clusters und es gibt viel mehr relativ kleine Eigenwerte.

Ein großer Eigenwert der Hesseschen Matrix bedeutet: die zugehörige Richtung von Veränderungen der Kontrollvariablen ist gut bestimmt. Kleine Eigenwerte gehören zu Richtungen in denen Veränderungen kaum Auswirkungen auf die Kostenfunktion haben. Die Eigenwerte unterhalb einer zu bestimmenden Grenze sind daher für die Frage der Bestimmtheit des Ergebnisses nicht relevant. Die Grenze ergibt sich aus der erwarteten Genauigkeit  $\Delta J$  der Kostenfunktion  $J$ . Nur die Eigenvektoren von Eigenwerten  $\lambda_k$ , für die

$$\frac{\lambda_k}{\lambda_{max}} \geq \frac{\Delta J}{J} \quad (64)$$

gilt, gehören zu Richtungen, in denen die Kontrollvariablen durch die Daten und



**Abbildung 12:** Eigenwertspektrum der Hesseschen Matrix für Experiment 6 mit Glattheitsbedingung (dicke Linie) und ohne (dünne Linie)

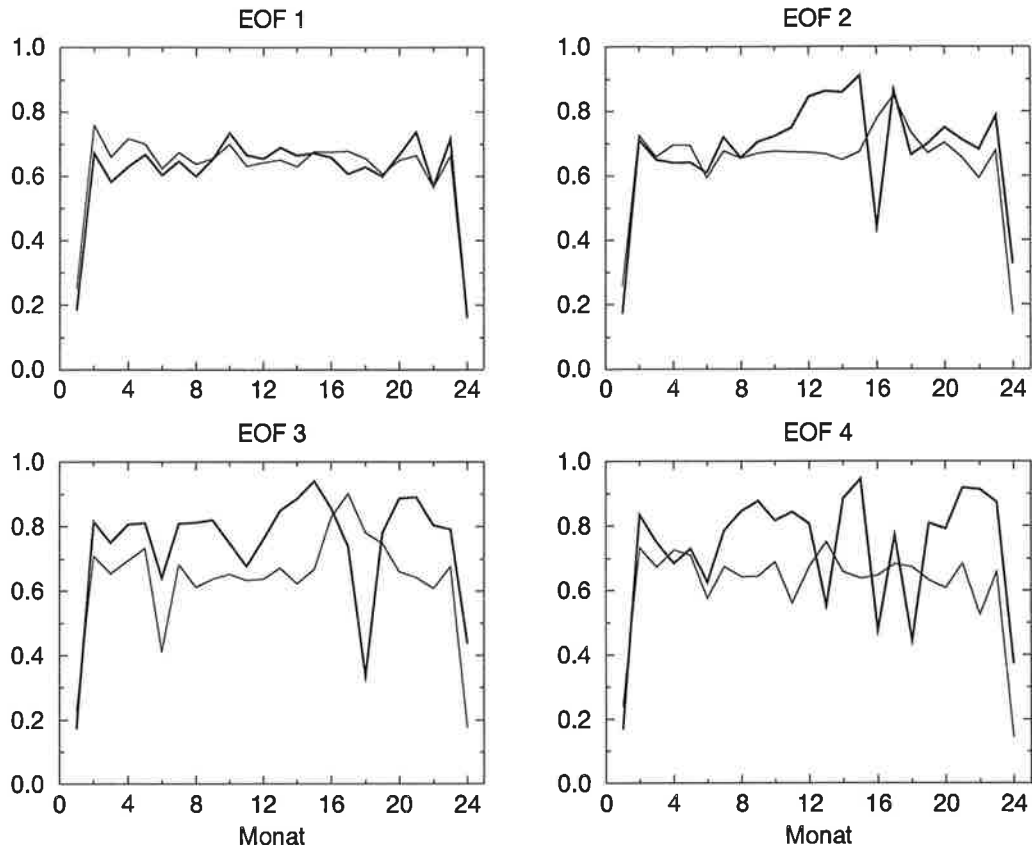
die Glattheitsbedingung festgelegt sind. Alle Eigenvektoren, deren Eigenwerte (64) erfüllen, spannen einen Unterraum auf. Der relative Anteil der Kontrollvariablen, welcher in dem Unterraum liegt, ist durch die sogenannte *Auflösungsmatrix* (resolution matrix) gegeben:

$$R_k = V_k V_k^t, \quad (65)$$

wobei die Matrix  $V_k$  die  $k$  Eigenvektoren als Spaltenvektoren enthält, deren Eigenwerte (64) erfüllen. Sind alle Richtungen gut bestimmt, so gehören alle Eigenvektoren zu  $V_k$  und  $R_k$  ist die Einheitsmatrix. Anderenfalls ist  $R_k$  eine Matrix, deren Diagonalelemente den Anteil der zugehörigen Kontrollvariablen angibt, welcher durch die Kostenfunktion am Minimum festgelegt ist.

Für die Hessesche Matrix des Experimentes 6 wird ein Fehler der Kostenfunktion von 6% angenommen, da der verbleibende gewichtete totale RMS Fehler für dieses Experiment 1.06 beträgt (Tab. 1). Damit sind 145 der 192 Eigenvektoren relevant. Die Diagonale der zugehörigen Auflösungsmatrix ist in Abb. 13 gezeigt. Amplituden zu Beginn und Ende des Zeitintervalls sind nicht gut bestimmt. Dies ist eine Folge der Glattheitsbedingung; Amplituden am Rand des Zeitintervalls erscheinen seltener

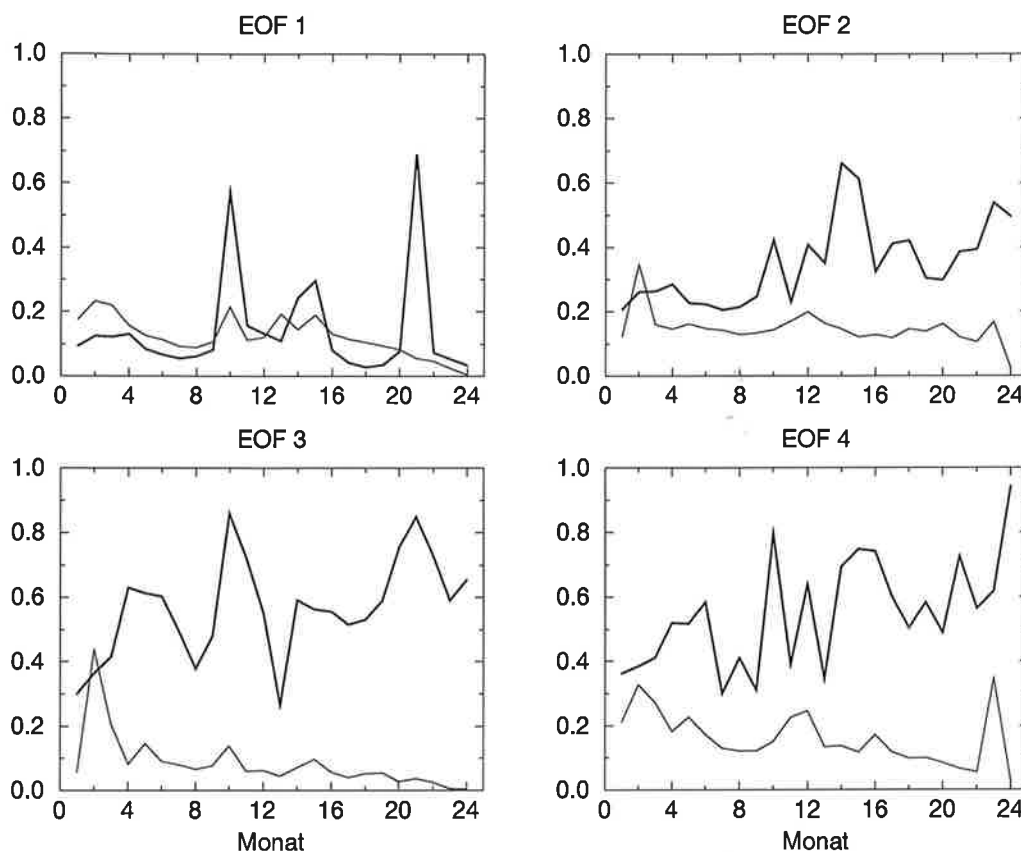




**Abbildung 13:** Diagonalelemente der Auflösungsmatrix. Die dünnen Linien entsprechen den Amplituden der zonalen EOFs und die dicken Linien denen der meridionalen EOFs.

in ihren Termen, daher wirkt sie stärker für 'innere' Amplituden. Die Amplituden der meridionalen EOF 3 und 4 sind besser bestimmt als die zugehörigen zonalen Amplituden. Für fast alle Zeitserien ist für die inneren Amplituden ein leichter Abfall mit der Zeit zu erkennen, d.h frühe Amplituden sind besser bestimmt als spätere. Allgemein wirken Randbedingungen, immer nur auf den nachfolgenden Anteil der Trajektorie. So hat der Windschub zu Beginn des Zeitintervalls Einfluß auf die gesamte Trajektorie, während der Windschub zum Ende des Intervalls nur noch den letzten Zustand beeinflussen kann. Vorausgesetzt wird dabei, daß die Wirkung der Randbedingungen eine bestimmte Zeit über dem Rauschen des Modells bleibt. Insgesamt liegen die Diagonalelemente der Auflösungsmatrix zwischen 0.5 und fast 0.9, die Amplituden werden also gut rekonstruiert und sind auch gut bestimmt.

Wird im Experiment auf die Glattheitsbedingung verzichtet, so sind fast alle Amplituden schlechter bestimmt (Abb. 14). Die Amplituden der meridionalen EOFs 2,3 und 4 sind dann enger festgelegt als die entsprechenden Amplituden der zonalen



**Abbildung 14:** Diagonalelemente der Auflösungsmatrix für Experiment 6 aber ohne Glattheitsbedingung. Die dünnen Linien entsprechen den Amplituden der zonalen EOFs und die dicken Linien denen der meridionalen EOFs.

EOFs. Deutlich ist hier der Abfall mit der Zeit für die Amplituden der zonalen EOFs zu erkennen. Doch für die Amplituden der meridionalen EOFs ist eher ein Anstieg mit der Zeit vorhanden. Offensichtlich ist das Gedächtnis des Modells für Variationen des meridionalen Windschubes sehr klein. Der beobachtete Anstieg resultiert dann daraus, daß die Wirkung von frühen Variationen durch die folgenden überdeckt wird und so deren Rekonstruktion erschwert.

## 5.2 Assimilierung im Pazifik im Zeitraum 1982-83

### 5.2.1 Datensätze und Kostenfunktion

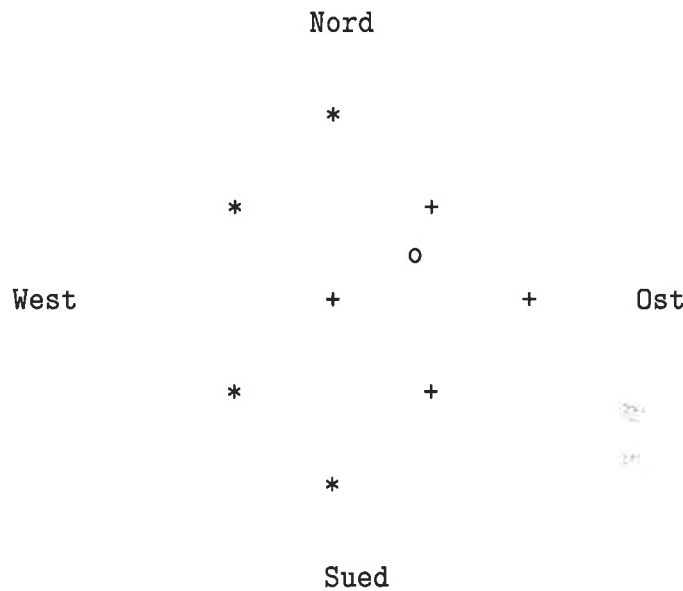
In den Experimenten werden drei verschiedene Datensätze (ECMWF-Windschub, FSU-Windschub und Reynolds-SST) zur Assimilierung verwendet. Es folgt zunächst eine kurze Beschreibung der Datensätze und die Definition der Kostenfunktion.

Von allen Datensätzen wird zunächst eine Klimatologie bestimmt. Verglichen werden dann immer die Modell- und Datenanomalien bezogen auf ihre jeweilige Klimatologie. Diese Vorgehensweise erwies sich als notwendig, da die Modellklimatologie noch erheblich von den Messungen abweicht. Eine Assimilierung absoluter Größen würde hauptsächlich die großen Fehler in der Modellklimatologie reduzieren und die Variationen um die Klimatologie bleiben fast ohne Einfluß. Eine Verbesserung des zyklischen stationären Modellzustandes durch Assimilierung von hydrographischen Daten wird zur Zeit in einem Experiment von Wenzel und Schröter (1995) erreicht.

Die Simulationen des Modells sind auf den Rändern der Topographie, vor allem im Bereich der westlichen Randströme, ungenau. Deshalb wird eine Messung nur dann berücksichtigt, wenn alle vier ihn umgebenden Gitterpunkte und vier weitere westlich davon gelegene zum Modellozean gehören (Abb. 15). Eine lineare Interpolation der Modellwerte an den vier umgebenden Gitterpunkten liefert die Modellvorhersage für die Messung. Die Interpolationsfaktoren werden vorab ermittelt, so daß während der Optimierung, zur Bestimmung der korrespondierenden Modellwerte, nur noch Linearkombinationen berechnet werden müssen.

**ECMWF-Windschub** Die monatlichen Windschubmittelwerte des Europäischen Zentrums für mittelfristige Wettervorhersagen (ECMWF) liegen global auf einem  $2.5^\circ * 2.5^\circ$  Gitter vor (Shaw 1987). Sie sind das Ergebnis einer Modellanalyse von halbtäglichen Messungen. Die Fehler dieses Datensatzes sind daher sicherlich stark korreliert. Doch hierüber und über die Größe der Fehler liegen keine Angaben vor. Es wird ein unkorrelierter Fehler von  $\sigma_E = 12mPa$  angenommen. Der Term in der Kostenfunktion ist die Summe der quadratischen Abweichungen zwischen dem Windschub des Modells  $\mathcal{T}$  und den Windschubdaten  $\mathcal{T}_E$  gewichtet mit dem Inversen des erwarteten Fehlers:

$$J_E = \frac{1}{2} \sum_{\text{Daten}} \left( \frac{\mathcal{T} - \mathcal{T}_E}{\sigma_E} \right)^2 . \quad (66)$$



**Abbildung15:** Horizontale Interpolation von Modellgitter auf einen Datenpunkt. Es bedeuten: + : Gitterpunkte, deren Modellwerte interpoliert werden. \* : Gitterpunkte, die zusätzlich zum Modellozean gehören müssen. o : Ort des Datums

**FSU-Windschub** An der Florida State University (FSU) wurde der monatliche Windschub des tropischen Pazifik berechnet (Legler und O'Brien 1984). Die Analyse basiert auf Schiffsmessungen des NODC (National Oceanographic Data Center). Die Daten sind auf einem  $2^\circ * 2^\circ$  Gitter zwischen 30S und 30N, bzw. 123O bis 69W gegeben. Auch für diesen Datensatz ist keine Fehlerabschätzung vorhanden. Aufgrund des engeren Gitters wird ein unkorrelierter Fehler von  $\sigma_F = 15mPa$  angenommen. Der Term in der Kostenfunktion für diesen Datensatz lautet:

$$J_F = \frac{1}{2} \sum_{\text{Daten}} \left( \frac{T - T_F}{\sigma_F} \right)^2 . \quad (67)$$

**Reynolds-SST** Die Temperaturen der Ozeanoberfläche (SST) werden auf einem globalen  $2^\circ * 2^\circ$  Gitter bereitgestellt (Reynolds 1988). Sie basieren auf in-situ- und Satellitenmessungen (AVHRR). Reynolds klassifiziert die einzelnen Werte nach ihren wesentlichen Quellen:

**Klasse 1:** mindestens 5 in-situ-Messungen

**Klasse 2:** Werte an den Polen :  $-1.8^\circ C$  gesetzt

**Klasse 3:** mit Eis bedeckte Fläche :  $-1.8^{\circ}C$  gesetzt

**Klasse 4:** Interpolation von in-situ- und Satellitenmessungen (mindestens 10 Satellitenmessungen)

**Klasse 5:** Interpolation von in-situ- und Satellitenmessungen (weniger als 10 Satellitenmessungen)

Im weiteren werden nur die Daten der Klassen 1,5 und 6 mit einem geschätzten Fehler von  $\sigma_R = 0.5^{\circ}C$  verwendet.

Ein Vergleich der Klimatologie von oberster Modellschichttemperatur  $T_1$  und der Reynolds-SST zeigt einen sehr starken linearen Zusammenhang oberhalb  $8.8^{\circ}C$ :

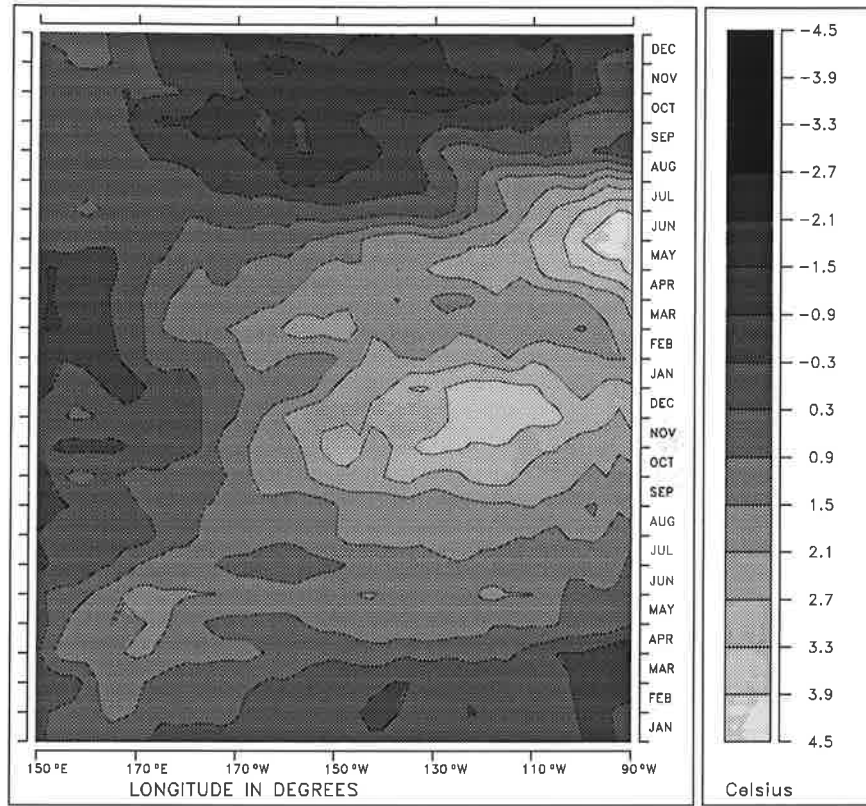
$$SST \approx -1.67 + 1.19 T_1 \quad (68)$$

Die Diskrepanz beruht im wesentlichen auf der fehlenden Modellierung der Solareinstrahlung. Die Strahlungsdichte ist Breitengrad- und jahreszeitabhängig. Ihre Eindringtiefe ist abhängig von der Wellenlänge, im Mittel beträgt sie etwa 20m. Durch die Absorption der Strahlung erwärmt sich das Wasser. Die Temperatur der obersten Schicht des Modells ist deshalb kälter als erwartet, wobei die Differenz mit der kurzwelligeren Strahlungsdichte und daher auch mit der Wassertemperatur zunimmt.

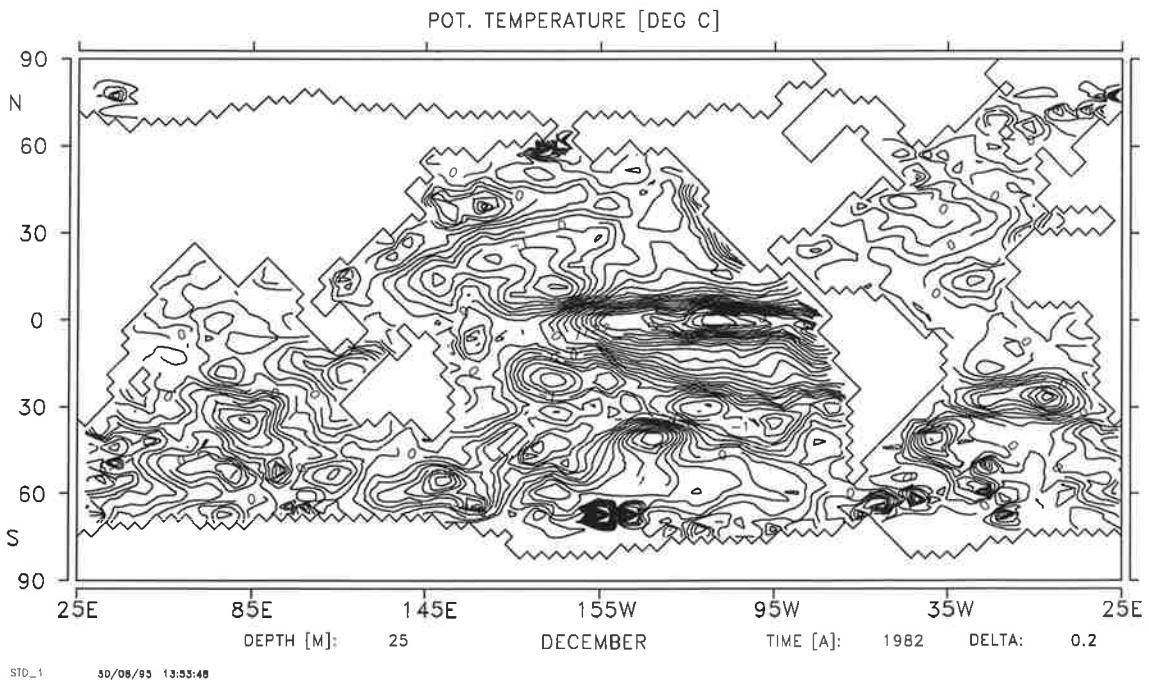
Die Oberflächentemperatur (SST) des Modells wird mithilfe dieser linearen Abbildung berechnet. Somit beträgt der Term in der Kostenfunktion für diesen Datensatz:

$$J_R = \frac{1}{2} \sum_{\text{Daten}} \left( \frac{SST - SST_R}{\sigma_R} \right)^2 \quad (69)$$

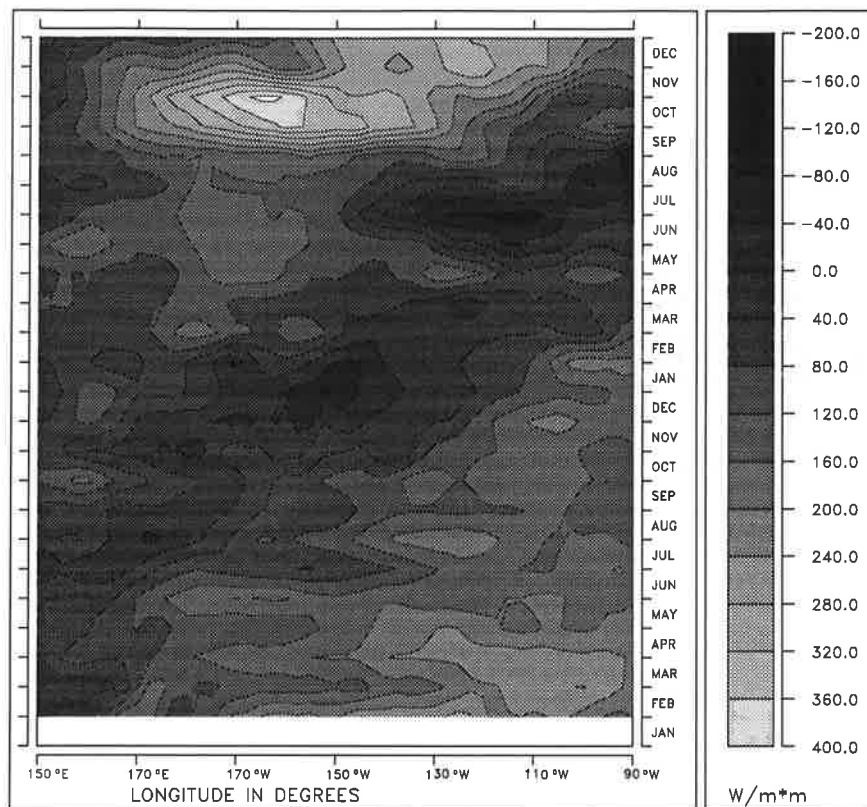
In Abb. 16 sind die Anomalien der Reynolds-SST im Pazifik in einem Hovmöller-Diagramm dargestellt. Der Schnitt verläuft auf dem Äquator; die Zeitachse geht von Januar 1982 bis Dezember 1983. Es sind zwei Maxima von etwa  $4^{\circ}K$  zu erkennen, im Dezember 1982 bei etwa  $120^{\circ}$  West und zwischen Mai und Juni 1983 bei etwa  $90^{\circ}$  West. Sie gehören beide zu der starken El-Niño Anomalie, die sich zu Beginn 1982 entwickelt und vom Westpazifik in den Ostpazifik wandert. Im Dezember 1982 ist die Anomalie am stärksten ausgeprägt (Abb. 17). Ihr Maximum konzentriert sich auf den Äquator bei etwa  $115^{\circ}W$ . Die Anomalie hat eine zonale Ausdehnung von mehr als  $50^{\circ}$ . Im Norden ist sie bei etwa  $10^{\circ}N$  scharf begrenzt, im Süden reicht sie bis etwa  $30^{\circ}S$ .



**Abbildung16:** Hovmöllerdiagramm der Reynolds-SST-Anomalien auf dem Äquator. Die Zeitachse beginnt unten mit dem Januar 1982.



**Abbildung17:** Reynolds-SST-Anomalie im Dezember 1982.



**Abbildung18:** Hovmöllerdiagramm der Wärme flu ßanomalien auf dem Äquator nach Assimilierung aller Datensätze und Variation von Windschub und Wärme flu ß

### 5.2.2 Ergebnisse

In den durchgeführten drei Experimenten werden entweder nur die Winddaten, oder nur die SST-Daten oder beide zusammen assimiliert. Die Kontrollvariablen sind immer die Windschubanomalien an jedem Gitterpunkt und für jeden Zeitschritt. Damit sind etwa  $2.6 \cdot 10^5$  Variablen zu optimieren.

Bei zusätzlicher Variation des Wärme flusses ist offensichtlich eine perfekte Anpassung der SST-Anomalien an die Daten für jeden Wind möglich. Der erforderliche Flu ß ist aber sehr unrealistisch, denn die Anomalien betragen mehr als  $400 \frac{W}{m^2}$  (Abb. 18); realistisch erscheinen Anomalien bis etwa  $40 \frac{W}{m^2}$  (Barnett, Latif, Kirk und Roekner 1991).

Die Resultate der drei Experimente sollen anhand des El-Niño-Ereignisses 1982/83 im Pazifik verglichen werden (Abb. 16 und 17). Dabei ist die Stärke der Anomalie, ihre Struktur und ihr zeitlicher Verlauf zu untersuchen.

Im ersten Experiment (Exp. 1) wird das Modell mit addierten Windschubanomalien, erhalten aus einer Interpolation der beiden Windschub Datensätze, angetrieben. Dies ist das klassische Sensitivitätsexperiment. Es kann als eine Optimierung der Kostenfunktion

$$J = J_E + J_F \quad (70)$$

betrachtet werden. Da nur Randbedingungen des Modells in die Kostenfunktion eingehen, bedarf es zur Minimierung keiner Modellintegration.

Das Modell reagiert auf diesen anomalen Windschub mit einer sehr starken SST-Anomalie (Abb. 19). Gegenüber den SST-Daten ist das erste Maximum etwa  $1^\circ K$  zu stark und es liegt um etwa  $30^\circ$  zu weit westlich. Beide Maxima erscheinen etwa einen Monat zu spät. Im Dezember 1982 gibt es eine starke negative SST-Anomalie im Indischen Ozean (Abb. 20), sie ist in den Reynolds-Daten (Abb. 17) nicht vorhanden. Verantwortlich dafür sind seewärts gerichtete anomale Winde, die ein Aufsteigen von kaltem Tiefenwasser bewirken.



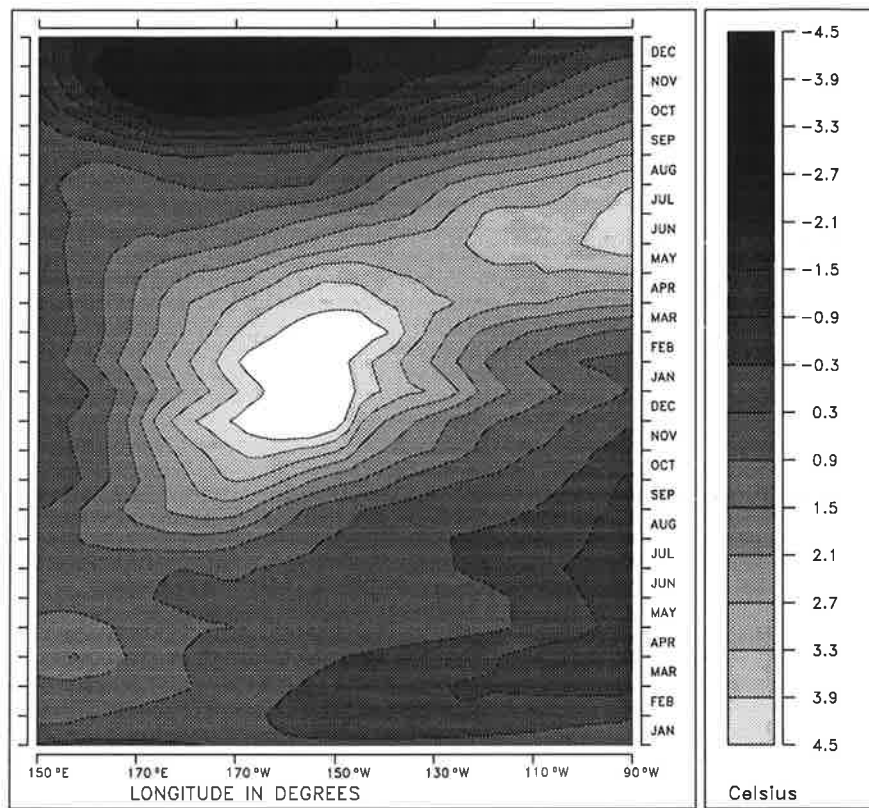


Abbildung19: Hovmöllerdiagramm der SST-Anomalien auf dem Äquator als Modellreaktion auf den anomalen Windschub der FSU- und ECMWF-Daten (Exp. 1)

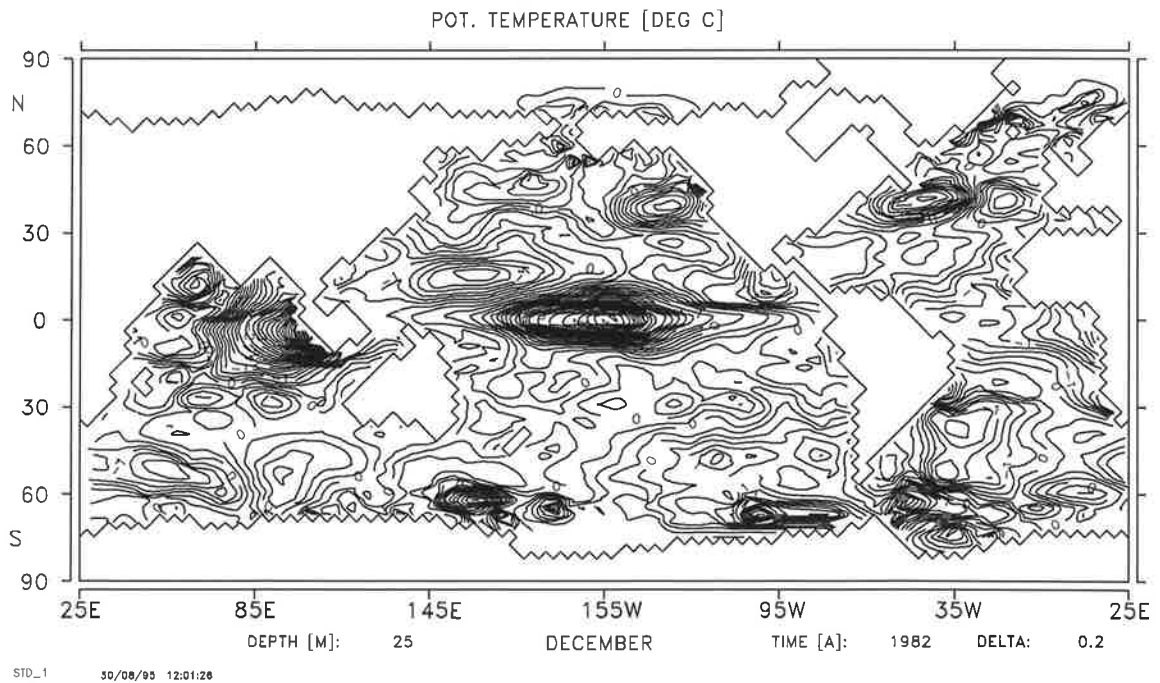


Abbildung20: SST-Anomalien im Dezember 1982 (Exp. 1)

Im zweiten Experiment (Exp. 2) werden die Reynolds SST-Daten durch Variation der Windschubanomalien assimiliert. Hierzu wird die Kostenfunktion

$$J = J_R \quad (71)$$

minimiert. Ohne die Formulierung einer Glattheitsbedingung für den Windschub ist dieses Problem sicherlich unterbestimmt. Da die Anomalien zum Start der Optimierung identisch Null sind und eine Gradientenmethode zur Suche des Minimums eingesetzt wird, wird die Lösung aus dem Lösungsraum mit den kleinsten Anomalien gefunden.

Dieses Experiment gibt die SST-Daten auf dem Äquator sehr gut wieder (Abb. 21). Position und Stärke der Anomalie stimmen gut mit den Daten überein. Auch ihre südöstliche Ausdehnung bis  $30^\circ S$  ist hier vorhanden (Abb. 22). Allein die kleinskalige Struktur fehlt in dieser Anpassung.

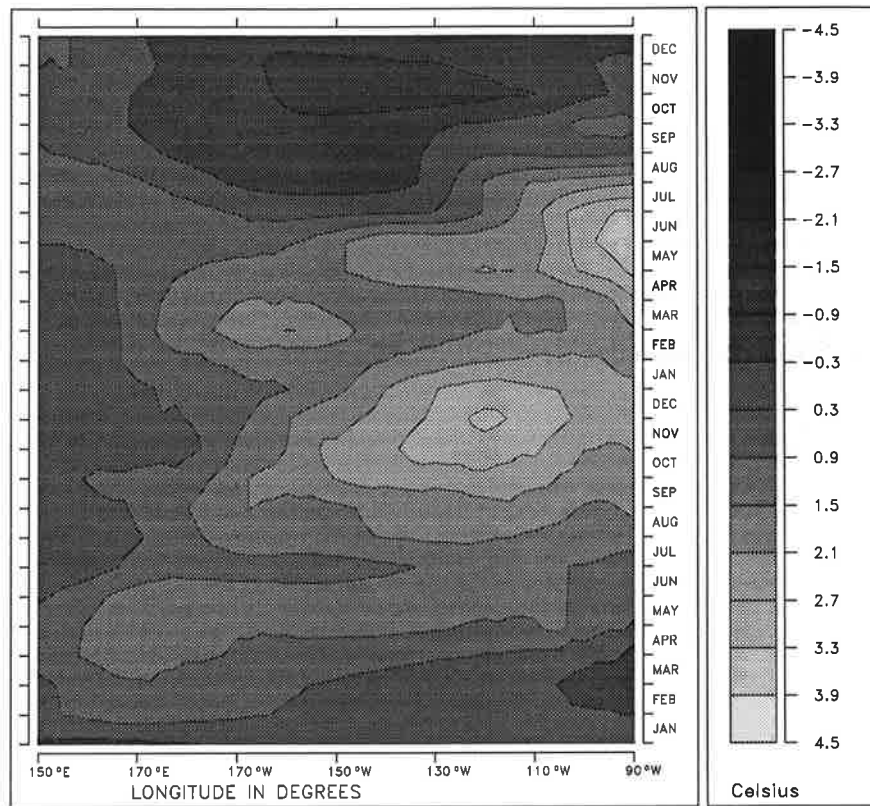


Abbildung21: Hovmöllerdiagramm der SST-Anomalien auf dem Äquator nach Assimilierung der Reynolds SST-Daten (Exp. 2)

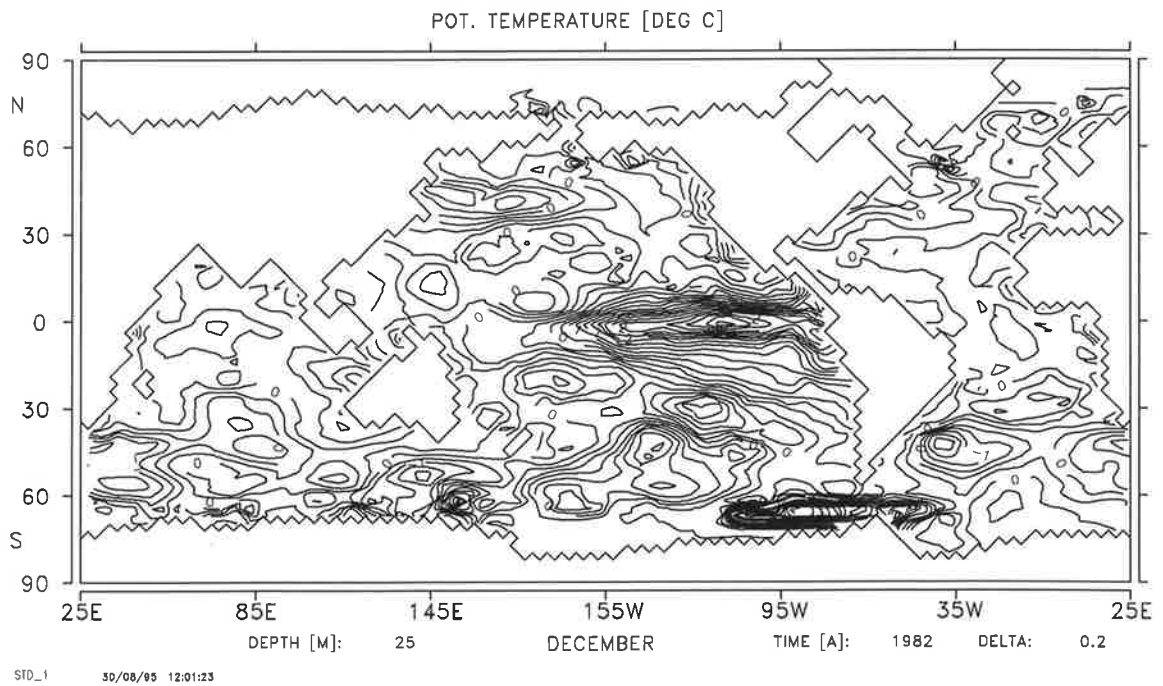


Abbildung22: SST-Anomalien im Dezember 1982 (Exp. 2)

Eine dynamische Interpolation zwischen den Windschubdatensätzen und den SST-Daten wird im dritten Experiment (Exp. 3) durchgeführt. Die Kostenfunktion enthält alle drei Terme:

$$J = J_E + J_F + J_R \quad (72)$$

Der Minimierungsalgorithmus variiert auch hier die Windschubanomalien, sie sind nun aber durch die Winddaten eingeschränkt.

Auch die SST-Anomalie des Exp. 3 liegt sehr nahe an den SST-Daten (Abb. 23). Das erste Maximum liegt ebenso wie in Exp. 1 zu weit westlich. Die Zeitpunkte der Maxima und die Stärke der Anomalien stimmen jetzt gut mit den Messungen überein. Die südöstliche Ausdehnung der Anomalie im Dezember 1982 ist vorhanden, aber weit weniger ausgeprägt als in den SST-Daten oder im Exp. 2 (Abb. 24). Die Anomalie im NW-Pazifik fehlt fast vollständig.

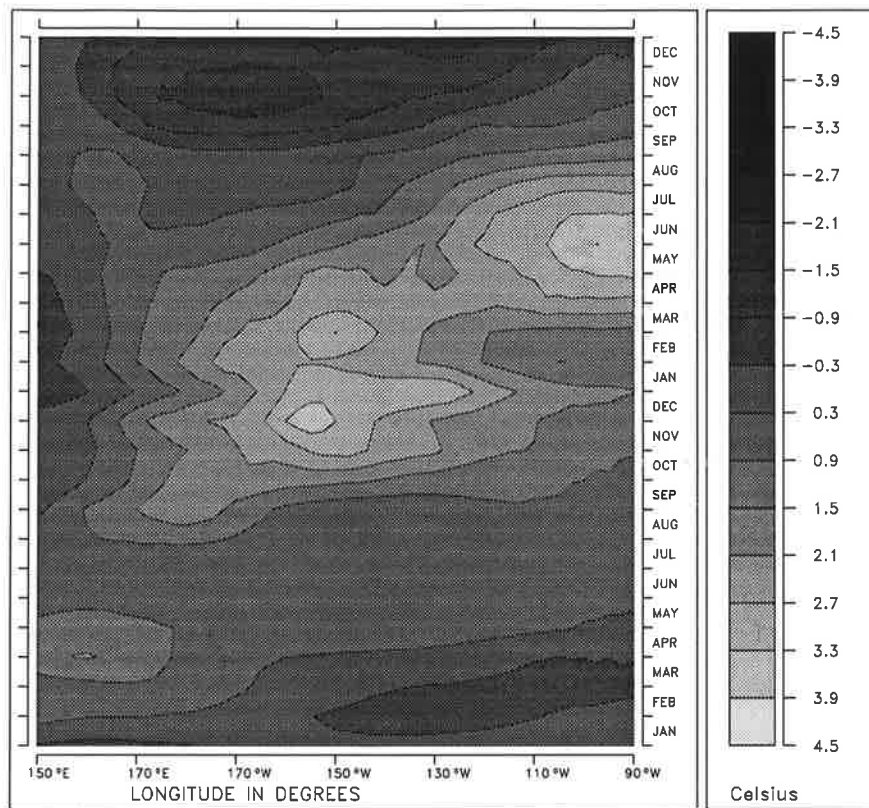


Abbildung23: Hovmöllerdiagramm der SST-Anomalien auf dem Äquator nach Assimilierung aller Datensätze (Exp. 3)

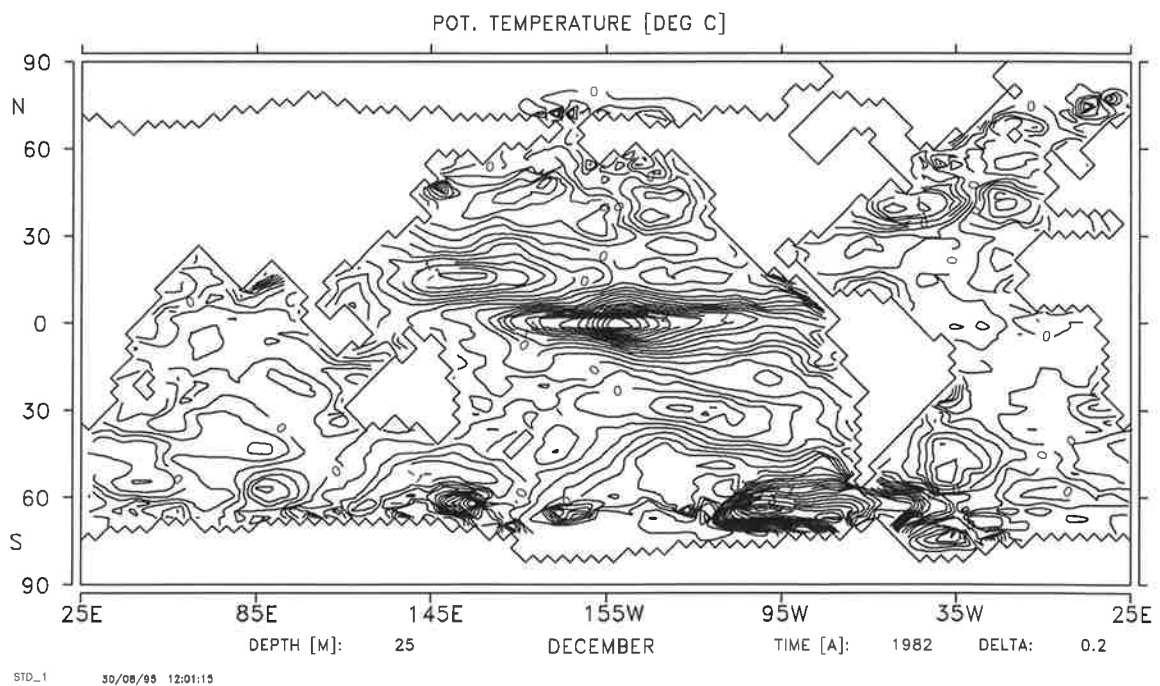


Abbildung24: SST-Anomalien im Dezember 1982 (Exp. 3)

Für den Antrieb des Modells mit einer Interpolation beider Windschubanomalien (Exp. 1) liegt die globale RMS-Differenz von Modell- und Reynolds-SST zwischen  $0.75$  und  $1.7^\circ K$  und ist damit relativ groß (Abb. 25). Die Differenz zwischen den Winddaten und ihrer Interpolation ist erwartungsgemäß klein (FSU:  $6mPa$ , ECMWF:  $11mPa$ ). Nach Assimilierung der Reynolds-SST (Exp. 2) kann der RMS-Fehler zwischen Modell und Reynolds-SST bis auf ungefähr  $0.4^\circ K$  reduziert werden. Die dazu nötigen Windschubanomalien sind aber von den Anomaliedaten weit entfernt (FSU RMS:  $30mPa$ , ECMWF RMS:  $35mPa$ ). Die Assimilierung aller Datensätze (Exp. 3) liefert dagegen Windschubanomalien mit einer RMS-Differenz zu den Anomaliedaten, die vergleichbar mit derjenigen der Interpolation der Windschubdaten sind. Dieser Windschub erzeugt eine SSTA-Zeitreihe mit einer RMS-Differenz zu der Reynolds-SST von etwa  $0.7^\circ K$ . Obwohl die Windschubanomalien, die aus den Datensätzen gewonnen wurden, eine schlechte SST-Anomalie erzeugen, kann durch kleine Änderungen, die im Bereich der Datenfehler bleiben, eine sehr viel bessere SST-Anomalie erreicht werden.

Das räumliche Muster der RMS-Differenzen zwischen Modell- und Reynolds-SST zeigt die größten Differenzen im Bereich der Eiskante (Abb. 26). Das Eismodell erlaubt nur vollständige Überdeckung mit Eis oder Eisfreiheit einer Gitterzelle. Da jedoch der Wärmefluß zwischen Ozean und Atmosphäre durch Eis wesentlich verringert wird, ändert der Wechsel von geringer Eisbedeckung zu Eisfreiheit die SST sehr stark. Ein kleiner Phasenfehler der Eisanomalie im Modell genügt, um die meist nur einen Gitterpunkt breiten großen Differenzen zu erzeugen. Für ständig eisbedeckte Zonen gibt es keine Differenzen, da diese Daten nicht berücksichtigt werden.

Ein weiteres Maximum liegt im südöstlichen Pazifik bei etwa  $50^\circ S$ . Es ist anzunehmen, daß dieses aus der geringen Datendichte und damit der Qualität der analysierten Daten in diesem Gebiet resultiert.

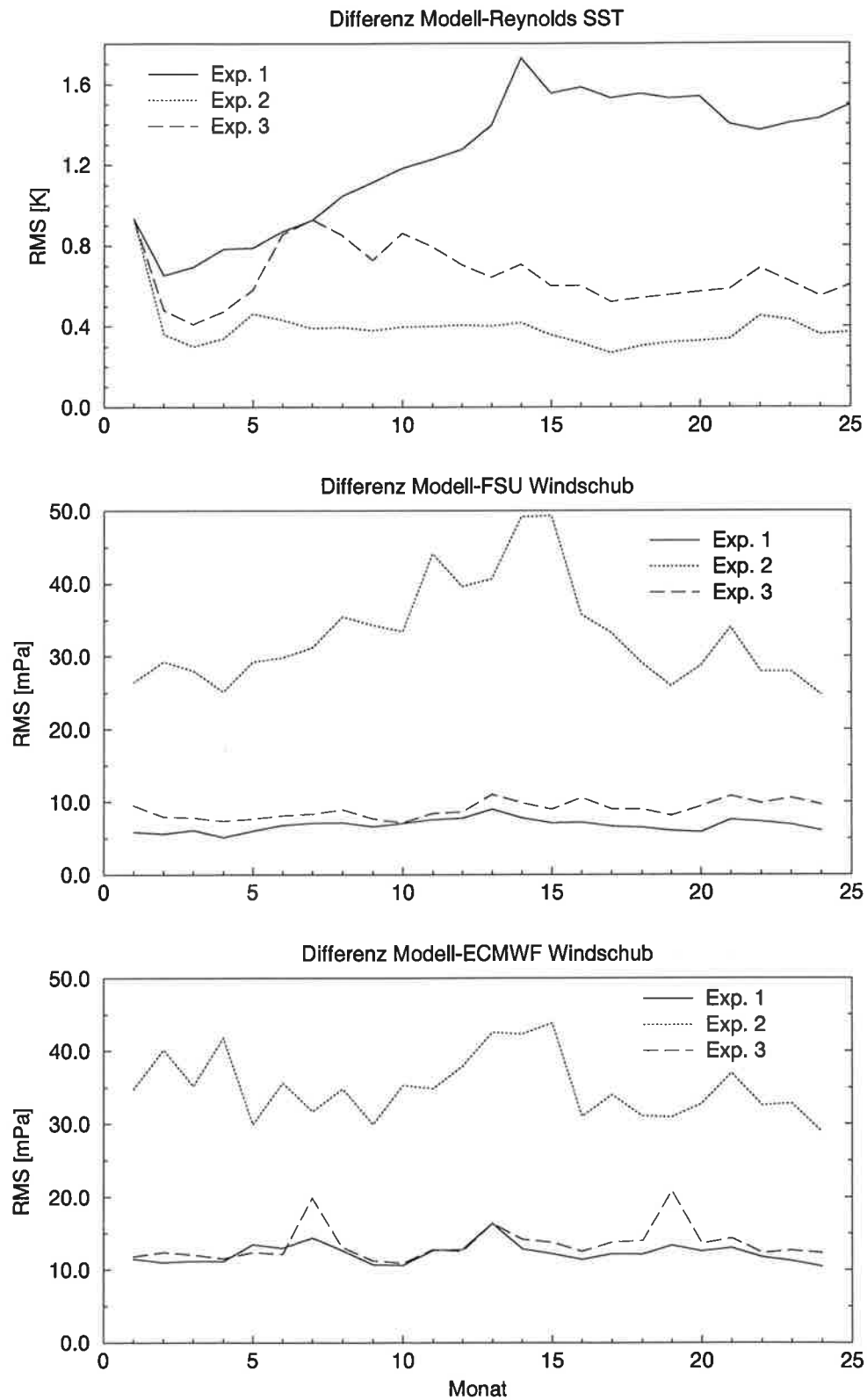
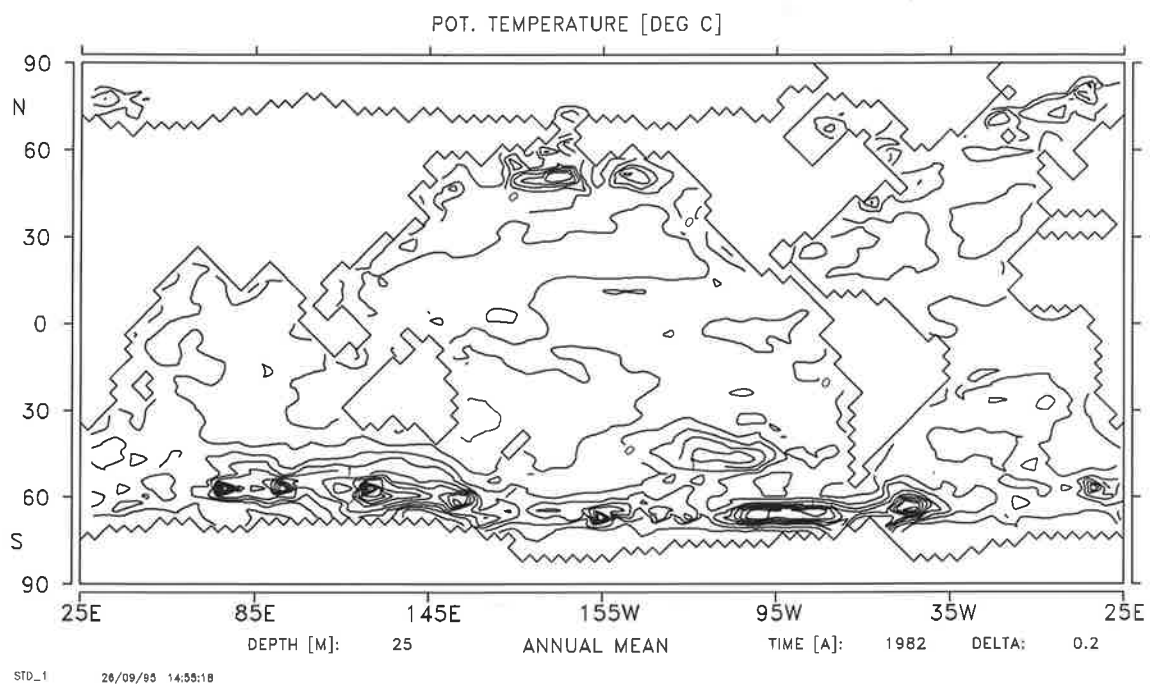


Abbildung 25: Global gemittelte RMS-Differenz zwischen Modell und Daten



**Abbildung 26:** RMS-Differenz zwischen Modell und Reynolds-SST-Daten für Exp. 3



## 6 Diskussion und Ausblick

Thema der vorliegenden Arbeit ist die Erstellung eines adjungierten Modells zur Assimilierung von Daten in ein Modell der globalen ozeanischen Zirkulation. Die adjungierte Methode besteht in der Bestimmung einer optimalen Trajektorie im Sinne eines minimalen Abstandes zwischen Modell und Daten mithilfe einer Gradientenmethode. Der Abstand wird dabei durch Definition einer Kostenfunktion quantifiziert. Diese Definition enthält die statistischen Informationen über Modell- und Datenfehler. Das adjungierte Modell liefert hierbei einen sehr effektiven Algorithmus zur Bestimmung des Gradienten der Kostenfunktion bezüglich der Kontrollvariablen.

Die verschiedenen Verfahren zur Erstellung von adjungierten Modellen wurden vorgestellt und deren Vor- und Nachteile diskutiert. Das in der vorliegenden Arbeit verwendete Verfahren beruht auf der Differentiation von Algorithmen im Rückwärtsmodus. Ausgehend von dem numerischen Programm des Modells wird das adjungierte Modell Anweisung für Anweisung anhand von einfachen Regeln konstruiert. Es werden grundlegende Konzepte aufgestellt, die einerseits die Ableitung von Regeln zur Konstruktion von adjungierten Anweisungen ermöglichen und andererseits die Pflege sowie die Erweiterbarkeit von adjungierten Code vereinfachen. Durch systematische Anwendung dieser Regeln kann der adjungierte Code auch für sehr umfangreiche und komplexe Modelle konstruiert werden. Ein zentrales Problem bei dem adjungierten Code ist die Bereitstellung von Bedarfsvariablen. Die Einbindung von Anweisungen zur Wiederberechnung von Bedarfsvariablen wurde dargestellt. Die dabei möglicherweise auftretenden Konflikte sind detailliert beschrieben worden und Lösungsmöglichkeiten wurden vorgeschlagen. Für die iterative Lösung nichtlinearer impliziter Gleichungen wurde alternativ die Konstruktion adjungierten Codes, der das Speichern von Bedarfsvariablen erspart, beschrieben.

Die Zwillingsexperimente zeigten zunächst die prinzipielle Brauchbarkeit der adjungierten Methode. Anhand von simulierten Oberflächen'messungen' konnten die Amplituden von Windschubmustern rekonstruiert werden, auch wenn die simulierten Messungen mit einem künstlichen aber realistischen Fehler versehen waren. Die Resultate zeigten, daß die Auslenkung der Ozeanoberfläche mehr Information über den Windschub enthält als ihre Temperatur. Dennoch konnte die zusätzliche Information der Temperatur zur besseren Rekonstruktion des Windschubes genutzt werden. Es zeigte sich auch eine Unbestimmtheit der Kontrollvariablen bezüglich  $2\Delta t$ -Wellen. Da diese vom Modell ohnehin nicht aufgelöst werden können, wurde ein zusätzlicher Term zur Bestrafung dieser Wellen in die Kostenfunktion eingeführt. Die

Analyse der Hesseschen Matrix der Kostenfunktion am Minimum zeigt, daß dadurch das Problem sehr viel besser bestimmt ist.

In einer realistischeren Anwendung wurde gezeigt, wie das Modell für die Jahre 1982/83 zu einer dynamisch konsistenten Interpolation zwischen verschiedenen Datensätzen verwendet werden kann. Die Variation des Windschubes bei gleichzeitiger Einschränkung durch Windschubdaten führte zu geringfügig von den Daten abweichenden Windschüben, die aber eine sehr viel bessere Modelltrajektorie erzeugten. Für zukünftige Datenassimilationen sollten neben weiteren Oberflächendaten (z.B. Auslenkung) auch direkte Messungen des inneren Aufbaus herangezogen werden. Erste Experimente mit XBT-Daten (expandable bathythermograph data) zeigen, daß sie zur verbesserten Bestimmung der großskaligen Zirkulation beitragen können. Zum intensiven Einsatz als Assimilationsmodell sollte das LSG-Modell in einigen Punkten verändert werden. Die Formulierung der Konvektion und des Eismodells sollten stetig differenzierbar sein, um den Einfluß durch Einsetzen von Konvektion oder Eisbedeckung auf die adjungierten Variablen zu berücksichtigen. Die Simulation der solaren Einstrahlung und eine höhere vertikale Auflösung des oberen Ozeans würden die Ergebnisse sicherlich verbessern.

Bei der hier angewandten adjungierten Methode der Datenassimilation bleiben Modellfehler weitgehend unberücksichtigt. Ihre Einbeziehung ist auf verschiedene Weise möglich. Die Differenzen zwischen Modell und Daten könnten zusätzlich in Abhängigkeit von der raumzeitlichen Struktur der Modellfehler in der Kostenfunktion gewichtet werden. Im Prinzip wurde dies durch die Einschränkung auf diejenigen Daten, die von acht Ozeangitterpunkten umgeben sind, erreicht. Die explizite Berücksichtigung der Modellfehler wird durch Definition der Modellgleichung als schwache Nebenbedingung in der Kostenfunktion, gewichtet mit ihren Fehlern, realisiert. Dann werden die Zustandsvariablen aller Zeitschritte zu Kontrollvariablen, und zusätzliche Bedingungen (Daten, 'a priori'-Informationen und -Annahmen) sind nötig, damit das Problem nicht stark unterbestimmt ist. Für systematische Modellfehler bieten sich zwei Methoden an. Zum einen können sie durch eine entsprechende Korrektur bei der Abbildung von Modellgitter auf den Meßort eliminiert werden. In den vorgestellten Experimenten wurde die SST aus der Temperatur der obersten Modellschicht mithilfe solch einer Korrektur berechnet. Zum anderen kann eine gut gewählte Verteilung von zusätzlichen Quellen und Senken die systematischen Fehler der Modelltrajektorie verringern. Dies erfordert jedoch u.U. ein erneutes kalibrieren des Modells.

Ein großer Nachteil der adjungierten Methode ist der rechenintensive Aufwand zur Fehlerabschätzung der Ergebnisse. Sie ist aber für die weitere Verwendung der

Ergebnisse wichtig. Für normalverteilte Datenfehler und nur schwach lineare Modelle kann die Fehlerkovarianzmatrix durch die inverse Hessesche Matrix der Kostenfunktion am Minimum approximiert werden. Für die Bestimmung der größten Eigenwerte einer Matrix muß diese jedoch nicht explizit vorliegen, es ist ausreichend, ihr Produkt mit einem beliebigen Vektor  $u$  ausrechnen zu können. Eine entsprechende Prozedur kann durch Differenzieren einer Prozedur zur Berechnung der Richtungsableitung generiert werden:

$$(\nabla^2 J) \cdot u = \nabla (\nabla J \cdot u) \quad (73)$$

Der Rechenaufwand für das Produkt beträgt etwa das 4- bis 10-fache der Modellintegration. Damit ist es möglich zumindest einige gut bestimmte Richtungen im Raum der Kontrollvariablen zu identifizieren.

In der Vergangenheit hat die aufwendige Erstellung des adjungierten Modells die Anwendung der adjungierten Methode eingeschränkt. Die Situation hat sich mit der Möglichkeit der automatischen Erzeugung von adjungierten Modellen wesentlich verbessert. Zur Zeit gibt es zwei Programme, die den Rückwärtsmodus der automatischen Differentiation von Algorithmen anwenden um ein adjungiertes Modell zu generieren: *Odyssée* (Rostaing, Dalmas und Galligo 1993) und *AMC* (Giering 1992). Mehrere umfangreiche adjungierte Modelle wurden bereits weitgehend automatisch erstellt (*ARPEGE/IFS*, *HOPE*, *WAM*, *TM2*). In naher Zukunft wird dies wahrscheinlich ein Standardwerkzeug sein, wie etwa das symbolische Differenzieren oder die automatische Vektorisierung von Programmen.

Die adjungierte Methode der Datenassimilation ist durch die Aufstellung und Minimierung einer Kostenfunktion ein sehr objektives Verfahren. Die einfache Einbeziehung von unterschiedlichsten Datensätzen und 'a priori'-Informationen durch Addition von Termen in der Kostenfunktion machen diese Methode sehr flexibel. Sie wird mit der Verbesserung der Modelle und der Erweiterung der Rechnerkapazitäten eine immer wichtigere Rolle spielen.

## Danksagung

Mein Dank gilt insbesondere meinem Betreuer Ernst Maier-Reimer, er ist immer eine reiche Quelle an Informationen und Anregungen. Als Autor des benutzten Ozeanmodells gab er mir unverzichtbare Hinweise über dessen interne Vorgänge.

Herrn Prof. Hasselmann danke ich, die Arbeit am Max-Planck-Institut für Meteorologie mit seinen sehr guten Arbeitsmitteln durchführen und die große Freiheit den Forscherdrang ausleben zu können. Für die Übernahme des Erstgutachtens sei Herrn Prof. Zahel gedankt.

Die Diskussionen über Datenassimilation und artverwandte Themen während vieler Spaziergänge mit Martin Fischer in der grünen Umgebung des Instituts waren immer wieder Quelle neuer Ideen und Anregungen. Mit Thomas Kaminski habe ich viele Aspekte des manuellen und automatischen Adjungierens erörtert, dies war für die Aufstellung von Regeln sehr wichtig.

Ich danke Marko Laudam für die Erstellung einiger Bilder und Christian Eckert für die Durchsicht der Arbeit.

Für die anregende und fruchtbare Atmosphäre am Institut möchte ich allen Kollegen danken.

Nicht zuletzt gilt mein herzlicher Dank meiner Freundin Dagmar Voigtlaender für die Unterstützung auch in schwierigen Phasen.

## Literatur

- Anderson, D. und Willebrand, J., eds (1989), *Ocean Circulation Models: Combining Data and Dynamics*, D. Reidel Publishing Company, Dordrecht.
- Anthes, R. (1974), 'Data assimilation and initialization of hurricane prediction models', *Journal of Atmosphere Science* **31**, 702–718.
- Arakawa, A. (1966), 'Computational design for long term numerical integration of the equations of fluid motion : Two-dimensional incompressible flow. Part I', *Journal of Computational Physics* **1**, 119–143.
- Barnett, T., Latif, M., Kirk, E. und Roeckner, E. (1991), 'On ENSO Physics', *Journal of Climate* **4**(5), 487–515.
- Barzel, G. (1994), Optimierung eines Seegangmodells mit der adjungierten Methode, PhD thesis, Max-Planck-Institut für Meteorologie, Hamburg, Germany.
- Baur, W. und Strassen, V. (1983), 'The complexity of partial derivatives', *Theoretical Computer Science* **22**, 317–330.
- Bengtsson, L., Ghil, M. und Kallen, E., eds (1981), *Dynamic Meteorology: Data Assimilation Methods*, Springer Verlag, New York.
- Bennet, W. (1989), 'The Kalman smoother for a linear quasi-geostrophic model of ocean circulation', *Dynamics of Atmospheres and Oceans* **13**, 219–267.
- Bischof, C. und Hovland, P. (1991), Using ADIFOR to compute dense and sparse jacobians, Technical report ANL/MCS-TM-158, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Ave., Argonne.
- Bryden, H. (1973), 'New polynomials for thermal expansion , adiabatic temperature gradient and potential temperature gradient of sea water', *Deep-Sea Research* **20**, 401–408.
- Buizza, R., Tribba, J., Molteni, F. und Palmer, T. (1993), 'Computation of optimal unstable structures for a numerical weather prediction model', *Tellus A*(45), 388–407.
- Christianson, B. (1993), Reverse accumulation and attractive fixed points, Internal report, School of Information Sciences, University of Hertfordshire, England.

- Courtier, P. und Talagrand, O. (1987), 'Variational assimilation of meteorological observations with the adjoint vorticity equation Part II : Numerical results', *Quarterly Journal of the Royal Meteorological Society* **113**, 1329–1347.
- Daley, R., ed. (1991), *Atmospheric data analysis*, Cambridge University Press.
- Fischer, M. und Latif, M. (1995), 'Assimilation of temperature and sea level observations into a primitive equation model of the tropical Pacific', *Journal of the Meteorological Society* **6**, 31–46.
- Ghil, M., Kohn, S., Tavantis, J., Bube, K. und Isaacson, E. (1981), Application of estimation theory to numerical weather prediction, in M. G. L. Bengtsson und E. Kallen, eds, 'Dynamic Meteorology: Data Assimilation Methods', Springer Verlag, New York.
- Giering, R. (1992), *Adjoint Model Compiler, users manual*, Max-Planck-Institut für Meteorologie.
- Giering, R. und Kaminski, T. (1996a), Recipes for Adjoint Code Construction, Technical Report 212, Max-Planck-Institut für Meteorologie.
- Giering, R. und Kaminski, T. (1996b), 'Recipes for Adjoint Code Construction'. submitted to ACM Trans. On Math. Software.
- Griewank, A. (1989), On automatic differentiation, in M. Iri und K. Tanabe, eds, 'Mathematical Programming: Recent Developments and Applications', Kluwer Akademic Publishers, Dordrecht, pp. 83–108.
- Griewank, A. (1992), 'Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation', *Optimization Methods and Software* **1**, 35–54.
- Griewank, A. und Reese, S. (1991), On the calculation of Jacobian matrices by the Markowitz rule, in A. Griewank und G. Corliess, eds, 'Automatic Differentiation of Algorithms: Theory, Implementation and Application', SIAM, Philadelphia, Penn, pp. 191–201.
- Hasselmann, K. (1982), 'An ocean model for climate variability studies', *Prog. Oceanog.* **11**, 69–92.

- Hellermann, S. und Rosenstein, M. (1983), 'Normal monthly windstress over the world ocean with error estimates', *Journal of Physical Oceanography* **13**, 1093–1104.
- Heras, M., Burgers, G. und Janssen, P. (1995), 'Wave data assimilation in the WAM wave model', *Journal of the Meteorological Society* **6**, 77–85.
- Holland, W. und Malanotte-Rizzoli, P. (1989), 'Assimilation of altimeter data into an ocean model. Space versus time resolution studies', *Journal of Physical Oceanography* **19**, 15507–1534.
- Hurlburt, H. (1986), 'Dynamic transfer of simulated altimeter data into an ocean model', *Journal of Geophysical Research* **91**, 2372–2400.
- Kalman, R. (1960), 'A new approach to linear filtering and prediction problems.', *J. Basic Eng.* **82D**, 35–45.
- Lanczos, C. (1957), 'Applied analysis', *Prentice Hall*.
- Large, W. und Pond, S. (1982), 'Sensible latent heat flux measurements over the ocean', *Journal of Physical Oceanography* **12**, 464–482.
- Le Dimet, F. und Talagrand, O. (1986), 'Variational algorithms for analysis and assimilation of meteorological observations: Theoretical aspects.', *Tellus A*(38), 97–110.
- Legler, D. und O'Brien, J. (1984), Atlas of tropical pacific wind stress climatology 1971-1980, Technical report, Florida State University, Dept. of Meteo.
- Lewis, J. und Derber, J. (1985), 'The use of adjoint equations to solve a variational adjustment problem with advective constraints.', *Tellus A*(37), 309–322.
- Long, R. und Thacker, W. (1989), 'Data assimilation into a numerical equatorial ocean model. Part I : The model and assimilation algorithm.', *Dynamics of Atmospheres and Oceans* **13**, 379–412.
- Lorenc, A. (1981), 'A global tree-dimensional multi-variate statistical interpolation scheme', *Monthly Weather Review* **109**, 701–721.
- Maier-Reimer, E., Mikolajewicz, U. und Hasselmann, K. (1993), 'Mean circulation of the Hamburg LSG OGCM and its sensitivity to the thermohaline surface forcing', *Journal of Physical Oceanography* **23**, 731–757.

- Oberhuber, J. (1988), An atlas based on the COADS data set: the budgets of heat buoyancy and turbulent kinetic energy at the surface of the global ocean, Technical Report 15, Max-Planck-Institut für Meteorologie, Hamburg, Germany.
- Reynolds, R. (1988), 'A real time global sea surface temperature analysis', *Journal of Climate* **1**, 75–86.
- Rostaing, N., Dalmas, S. und Galligo, A. (1993), 'Automatic differentiation in Odyssee', *Tellus* pp. 558–568.
- Sasaki, Y. (1970), 'Some basic formalisms in numerical variational analysis', *Monthly Weather Review* **98**(12), 875–883.
- Schiller, A. (1993), Assimilation hydrographischer Daten in ein Zirkulationsmodell des Atlantiks auf der Grundlage des adjungierten Verfahrens, PhD thesis, Institut für Meereskunde, Christian-Albrechts-Universität, Kiel, Germany.
- Schröter, J. (1989), Driving of non-linear time dependent ocean models by observations of transient tracer - a problem of constrained optimization., in D. Anderson und J. Willebrand, eds, 'Ocean Circulation Models: Combining Data and Dynamics', Kluwer Academic Publishers, pp. 257–285.
- Schröter, J. (1992), 'Variational assimilation of GEOSAT data into an eddy-resolving model of the gulf-stream extension area', *Journal of Physical Oceanography* **23**, 925–953.
- Shaw, D. (1987), 'The 1984/85 revisions of the ECMWF assimilation system', *Quarterly Journal of the Royal Meteorological Society* **113**, 1048–1066.
- Talagrand, O. (1991), The use of adjoint equations in numerical modeling of the atmospheric circulation, in A. Griewank und G. Corliess, eds, 'Automatic Differentiation of Algorithms: Theory, Implementation and Application', SIAM, Philadelphia, Penn, pp. 169–180.
- Talagrand, O. und Courtier, P. (1987), 'Variational assimilation of meteorological observations with the adjoint vorticity equation Part I : Theory', *Quarterly Journal of the Royal Meteorological Society* **113**, 1311–1328.
- Thacker, W. (1987), Three lectures on fitting numerical models to observations., Technical report, GKSS Forschungszentrum Geesthacht GmbH, Geesthacht, Federal Republic of Germany.



- Thacker, W. (1988), 'Fitting models to inadequate data by enforcing spatial and temporal smoothness.', *Journal of Geophysical Research* **93**, 10655–10665.
- Thacker, W. (1989), 'The role of the Hessian Matrix in fitting Models to measurements.', *Journal of Geophysical Research* **94**, 6177–6196.
- Thacker, W. (1991), Automatic differentiation from an oceanographer's perspective, in A. Griewank und G. Corliess, eds, 'Automatic Differentiation of Algorithms: Theory, Implementation and Application', SIAM, Philadelphia, Penn, pp. 191–201.
- Tziperman, E. und Thacker, W. (1989), 'An optimal control/adjoint equation approach to studying the ocean general circulation', *Journal of Physical Oceanography* **19**, 1471–1485.
- Tziperman, E., Thacker, W., Long, R. und Hwang, S.-M. (1992), 'Ocean data analysis using a general circulation model, I, simulations', *Journal of Physical Oceanography* **22**, 1434–1457.
- Tziperman, E., Thacker, W., Long, R., Hwang, S.-M. und Rintoul, S. (1992), 'Ocean data analysis using a general circulation model, II, North Atlantic model', *Journal of Physical Oceanography* **22**, 1458–1485.
- UNESCO (1983), Eleventh report of the joint panel on oceanographic tables and standards., UNESCO Technical Papers in Marine Science 36, UNESCO, Paris.
- Wenzel, M. und Schröter, J. (1995), Finding a cyclo steady state of a global ocean circulation model by data assimilation, in prep.
- Woodruff, S., Sluts, R., Jenne, R. und P.M.Steurer (1987), 'A Comprehensive Ocean-Atmosphere Data Set', *Bull. Am. Met. Soc.* **68**(10), 1239–1250.
- Woods, J. (1985), 'The world ocean circulation experiment', *Nature* **31**(6011), 501–511.

- EXAMENSARBEIT Nr. 14**  
März 1992  
**Windinduzierte interannuale Variabilität in der Warmwassersphäre von 1981 bis 1987**  
**Teil II: Fluktuationen im Kohlenstoffkreislauf**  
Arne M. E. Winguth
- EXAMENSARBEIT Nr. 15**  
Dezember 1992  
**Fernerkundung der optischen Eigenschaften von Seen mit Landsat 5 Thematic Mapper**  
Waltraud Manschke
- EXAMENSARBEIT Nr. 16**  
Juli 1993  
**Baroklin instabile Wellen der Atmosphäre:  
Empirisch abgeleitete Moden im Vergleich zu quasi-geostrophischer Theorie**  
Reiner Schnur
- EXAMENSARBEIT Nr. 17**  
September 1993  
**Messung des Wasserdampfes in der konvektiven Grenzschicht mit DIAL und RADAR-RASS**  
Christoph Senff
- EXAMENSARBEIT Nr. 18**  
November 1993  
**Zur saisonalen Variation des ozeanischen Kohlendioxidpartialdrucks**  
Katharina D. Kurz
- EXAMENSARBEIT Nr. 19**  
Januar 1994  
**Dynamisch-stochastische Vorhersage-Experimente mit Modellen der allgemeinen Zirkulation für einen Zeitraum von 10 Tagen bis 100 Jahren**  
Ulrich Cubasch
- EXAMENSARBEIT Nr. 20**  
Februar 1994  
**Deposition radioaktiver Tracer in einem Transportmodell der Atmosphäre**  
Stefan Rehfeld
- EXAMENSARBEIT Nr. 21**  
Februar 1994  
**Der Einfluß vulkanischen Aerosols und simultaner Temperaturanomalien der tropischen Meeresoberfläche ( El Niño ) auf das Klimasystem**  
Ingo Kirchner
- EXAMENSARBEIT Nr. 22**  
Juni 1994  
**ENSO induzierte Variabilität im Indischen Ozean**  
Andreas Villwock
- EXAMENSARBEIT Nr. 23**  
Juni 1994  
**Untersuchungen zu Mechanismus und Vorhersagbarkeit von ENSO mit einem vereinfachten gekoppelten Ozean - Atmosphäre - Modell**  
Moritz Flügel
- EXAMENSARBEIT Nr. 24**  
Juli 1994  
**Nordatlantische Variabilität in einem Ozeanzirkulationsmodell im Zeitbereich von Dekaden**  
Ralf Weiß
- EXAMENSARBEIT Nr.25**  
August 1994  
**Inverse Modellierung des atmosphärischen Methan-Kreislaufs unter Verwendung eines drei-dimensionalen Modells des Transports und der Chemie der Troposphäre**  
Ralf Hein
- EXAMENSARBEIT Nr. 26**  
August 1994  
**Optimierung eines Seegangmodells mit der Adjungierten Methode**  
Georg Barzel
- EXAMENSARBEIT Nr. 27**  
März 1995  
**Stabile Wasserisotope im allgemeinen Zirkulationsmodell ECHAM**  
Georg Hoffmann

- EXAMENSARBEIT Nr. 28**  
Mai 1995  
**The Atmospheric Response to North Atlantic Sea Surface Temperature Anomalies in GCM Experiments**  
Viacheslav V. Kharin
- EXAMENSARBEIT Nr. 29**  
Juni 1995  
**Einbindung der regionalen troposphärischen Chemie in die Hamburger Klimamodellumgebung: Modellrechnungen und Vergleich mit Beobachtungsdaten**  
Bärbel Langmann
- EXAMENSARBEIT Nr. 30**  
Juli 1995  
**Messung des Ozonflusses in der unteren Troposphäre mit einem neuen Ozon-DIAL-System und einem Radar-RASS**  
Thorsten Schaberl
- EXAMENSARBEIT Nr. 31**  
November 1995  
**Untersuchungen troposphärischen Ozons mit DIAL**  
Gunther C. Grabbe
- EXAMENSARBEIT Nr. 32**  
November 1995  
**Die Schneedecke als Komponente des Klimasystems und ihre Modellierung**  
Bettina Loth
- EXAMENSARBEIT Nr. 33**  
Dezember 1995  
**Der Einfluß des El Niño/Southern Oscillation Phänomens auf die atmosphärische Zirkulation in den mittleren Breiten der Nordhalbkugel: Beobachtungen und Modellsimulationen**  
Wilhelm May
- EXAMENSARBEIT Nr. 34**  
Dezember 1995  
**DIAL-Messungen von vertikalen Wasserdampfverteilungen Ein Lasersystem für Wasserdampf- und Temperaturmessungen in der Troposphäre**  
Volker Wulfmeyer
- EXAMENSARBEIT Nr. 35**  
Januar 1996  
**Der Einfluß von Datenassimilation auf ENSO Simulationen und Vorhersagen**  
Martin Fischer
- EXAMENSARBEIT Nr. 36**  
Februar 1996  
**Bodengebundene Fernerkundung von Eiswolken im nahen Infrarot**  
- Margarita Betancor Gothe
- EXAMENSARBEIT Nr. 37**  
März 1996  
**Turbulenzcharakteristiken von Golfstromtrajektorien in einem quasigeostrophischen Zirkulationsmodell basierend auf Lagrange'schen Vorticitytracern**  
Miguel Zorita Calvo
- EXAMENSARBEIT Nr. 38**  
Mai 1996  
**Entwicklung eines Kopplungsverfahrens zur Reduzierung der Rechenzeit von Atmosphäre-Ozean-Modellen**  
Reinhard Voß
- EXAMENSARBEIT Nr. 39**  
Mai 1996  
**Kombination von Ensemble-Klimavorhersagen am Beispiel ENSO**  
Sven Marcus Metzger (März 1995)
- EXAMENSARBEIT Nr.40**  
Juni 1996  
**Der Einfluß der quasi-zweijährigen Oszillation auf die allgemeine Zirkulation: Modellsimulationen mit ECHAM4**  
Marco Andrea Giorgetta
- EXAMENSARBEIT Nr. 41**  
Juli 1996  
**Sensitivität des Modellklimas eines globalen Zirkulationsmodells der Atmosphäre gegenüber Änderungen der Wolkenmikrophysik**  
Ulrike Lohmann
- EXAMENSARBEIT Nr. 42**  
July 1996  
**Simulation der Kohlenstoffdynamik der globalen Landbiosphäre mit SILVAN-Modellbeschreibung und Ergebnisse**  
Jörg Kaduk

**EXAMENSARBEIT Nr. 43**  
Oktober 1996

**Zur Dynamik des interhemisphärischen CO<sub>2</sub>-Transports im Ozean**  
Christine Weber

**EXAMENSARBEIT Nr. 44**  
November 1996

**Erstellung eines adjungierten Modells zur Assimilierung von Daten in ein Modell der globalen ozeanischen Zirkulation**  
Ralf Giering