

Kanal in der β -Ebene: Gitterpunktmodell

TM2 Numerik
Wintersemester 2010/2011

Veronika Wolf, 1048360
Felix Dietzsch, 1473979

9. MÄRZ 2011

Inhaltsverzeichnis

1. Problemstellung	3
2. Analytische Lösung	4
3. Numerische Lösung der barotropen Vorticitygleichung	6
3.1. Helmholtzgleichung	6
3.2. Der Laplaceoperator	6
3.3. Das sukzessive (Über-)Relaxationsverfahren	7
3.4. Zeitliche Integrationsverfahren	10
3.5. Umsetzung der analytischen Lösung	11
4. Ergebnisse	13
4.1. Vorhersage einer einzelnen Rossbywelle	13
4.2. Vorhersage eines gegebenen Anfangsfeldes	17
A. Quellcode	20
A.1. Präambel	20
A.2. Deklaration der Testfunktion	20
A.3. Einlesen des Anfangsfeldes	21
A.4. Integration über die Zeit	22
A.5. Sukzessive Überrelaxation	23
A.6. Berechnung der Restfunktion F	25
A.7. Euler-Schritt	26
A.8. Leapfrog-Schritt	26
Literaturverzeichnis	28

1. Problemstellung

Das Numerikpraktikum fand im Wintersemester 2010/2011 des Studiengangs M.Sc. Meteorologie statt. Ziel war es, mittels eines selbstprogrammierten Modells eine 24-Stunden-Vorhersage für ein gegebenes Geopotentialfeld zu erstellen. „Neben seiner Bedeutung für konzeptionelle und analytische Studien der groß- und synoptischskaligen Zirkulation ist das barotrope Modell von historischem Interesse für die numerische Meteorologie, da die ersten erfolgreichen Versuche zur numerischen Wettervorhersage mit einem barotropen Modell (der (quasi-)geostrophisch divergenzfreien Version) durchgeführt wurden.“¹ Die genaue Aufgabenstellung lautete wie folgt:

Lösen Sie die barotrope Vorticitygleichung

$$\frac{\partial}{\partial t}(\nabla^2 - H)\psi + \mathcal{J}(\psi, \nabla^2\psi) + \beta \frac{\partial\psi}{\partial x} = 0 \quad (1.1)$$

in einem Kanal auf der β -Ebene mit periodischen Randbedingungen in x -Richtung und festen Rändern bei $y = 0, y = L_y$. Zur Lösung der auftretenden Helmholtzgleichung soll ein Relaxationsverfahren verwendet werden. Das Anfangsfeld wird aus beobachteten Daten abgeleitet vorgegeben. Insbesondere enthält es 65×21 Gitterpunktwerte der 500 hPa-Stromfunktion für das Breitenband $0 \leq \lambda \leq 360$ und $65 \geq \varphi \geq 25$, das heißt $\Delta\lambda = 5.625$, $\Delta\varphi = 2.0$. Dies entspricht einem Gitterpunktabstand von $\Delta y = 222.53$ km, während Δx zwischen 264.50 und 567.23 km variiert. Dieses Problem kann nur durch Einführung von Maßstabsfaktoren in (1.1) gelöst werden. Hier soll allerdings darauf verzichtet werden und der Einfachheit halber mit einem konstanten $\Delta x = 442.55$ (das ist der Wert für die Berührungsbreite $\varphi = 45^\circ$) gerechnet werden.

1. Überprüfen Sie zunächst die (1.1) zugrundeliegende Rossbywellentheorie durch Vorhersage einer einzigen Rossbywelle mit ganzzahligen Wellenzahlen k und l .

$$\begin{aligned} \psi_l^k(x = m\Delta x, y = n\Delta y) &= \\ &= A \left(\sin\left(\frac{2\pi km}{64}\right) + \cos\left(\frac{2\pi km}{64}\right) \right) \sin\left(\frac{\pi ln}{20}\right) - Un\Delta y \end{aligned} \quad (1.2)$$

2. Führen Sie sodann eine Vorhersage mit dem gegebenen Anfangsfeld über 24 h durch.

$$H = \frac{1}{R_d^2}, \quad R_d = 1000 \text{ km}, \quad \beta = 1.62 \cdot 10^{-11} \text{ m}^{-1} \text{ s}^{-1}$$

¹vgl. http://www.mi.uni-hamburg.de/uploads/media/uebung_barotrop.pdf, Seite 1 (22.02.2011)

2. Analytische Lösung

Zu lösen ist die partielle Differentialgleichung (1.1):

$$\frac{\partial}{\partial t}(\nabla^2 - H)\psi + \mathcal{J}(\psi, \nabla^2\psi) + \beta \frac{\partial\psi}{\partial x} = 0$$

Hierbei bezeichnet $\mathcal{J}(\psi, \nabla^2\psi)$ den Jacobioperator

$$\mathcal{J}(\psi, \nabla^2\psi) = \frac{\partial\psi}{\partial x} \frac{\partial}{\partial y} \nabla^2\psi - \frac{\partial\psi}{\partial y} \frac{\partial}{\partial x} \nabla^2\psi \quad (2.1)$$

Zunächst wird (1.1) linearisiert mit dem Ansatz

$$\psi = \bar{\psi} + \psi' \quad (2.2)$$

Der Grundstrom $\bar{\psi}$ in (2.2) wird definiert als

$$\bar{\psi} = -\bar{U} \cdot y \quad (2.3)$$

mit der zonalen Grundströmungsgeschwindigkeit $\bar{U} = \text{const.}$ Das Integrationsgebiet ist ein Kanal in der β -Ebene. Die Randbedingungen in x -Richtung sind dabei periodisch, d.h. es gilt

$$\psi(x=0) = \psi(x=L \cdot x) \quad (2.4)$$

Die Ränder in y -Richtung sind fest:

$$v(y=0) = v(y=L \cdot y) = 0 \quad (2.5)$$

Wegen $v = \frac{\partial\psi}{\partial x}$ gilt somit $\psi(y=0) = \text{const.}$ Daraus folgt

$$\psi'(y=0) = \psi'(y=L \cdot y) = 0 \quad (2.6)$$

Während der Linearisierung fallen die Grundstromterme mit $\bar{\psi}$ heraus, da diese alleine die Gleichung erfüllen. Weiterhin werden Produkte der Störgröße ψ' mit Null genähert. Übrig bleiben gemischte Terme aus $\bar{\psi}$ und ψ' . Die resultierende linearisierte Gleichung (1.1) lautet dann

$$\frac{\partial}{\partial t} \nabla^2 \psi' - H \frac{\partial \psi'}{\partial t} - \bar{U} \frac{\partial}{\partial x} \nabla^2 \psi' + \beta \frac{\partial \psi'}{\partial x} = 0 \quad (2.7)$$

Zur Lösung von (2.7) wird der Ansatz

$$\psi' = A \cdot e^{i(k \cdot x - \omega \cdot t)} \sin(ly) \quad (2.8)$$

gewählt unter der Berücksichtigung der Randbedingungen (2.4)-(2.6). Der Parameter A beeinflusst die Amplitude der Störung ψ' . Einsetzen in (2.7) liefert die Dispersionsrelation

$$\omega = \frac{\bar{U}k(k^2 + l^2) - \beta k}{H + k^2 + l^2} \quad (2.9)$$

Weiterhin erhält man die Phasengeschwindigkeit in x -Richtung mit

$$c_x = \frac{\omega}{k} = \frac{\bar{U}(k^2 + l^2) - \beta}{H + k^2 + l^2} \quad (2.10)$$

Zu beachten ist, dass k und l in diesem Fall die auf das Gitternetz skalierten Wellenzahlen darstellen. Die tatsächlichen Wellenzahlen k^* und l^* erhält man aus der Beziehung

$$k^* = \frac{2\pi k}{\Delta x \cdot I}; \quad l^* = \frac{\pi l}{\Delta y \cdot J} \quad (2.11)$$

mit der Anzahl der Gitterpunkte I bzw. J in x - bzw. y -Richtung. Die vollständige Lösung der barotropen Vorticitygleichung lautet somit

$$\psi = \bar{\psi} + \psi' = -\bar{U} \cdot y + A[\sin(k^* \cdot x - \omega^* \cdot t) + \cos(k^* \cdot x - \omega^* \cdot t)] \sin(l^* \cdot y) \quad (2.12)$$

Mit dieser Lösung kann der Zustand einer Rossbywelle zu jedem Zeitpunkt t berechnet werden. Sie wird verwendet, um die numerische Lösung mit der tatsächlichen Lösung zu vergleichen (vgl. Kapitel 3.5).

3. Numerische Lösung der barotropen Vorticitygleichung

3.1. Helmholtzgleichung

Die barotrope Vorticitygleichung ist eine elliptische Differentialgleichung des Typs

$$\nabla^2 G - HG + F = 0 \quad (3.1)$$

Eine Gleichung solchen Typs wird als Helmholtzgleichung bezeichnet. Ein Vergleich mit der barotropen Vorticitygleichung (1.1) zeigt, dass $\frac{\partial \psi}{\partial t}$ dem gesuchten G entspricht¹. H ist der Rossbyparameter und hier mit dem Deformationsradius $\frac{1}{R_d^2}$ definiert. R_d ist mit 1000 km gegeben. Das Absolutglied F ist ebenfalls gegeben durch

$$F = \beta \frac{\partial \psi}{\partial x} + J(\psi, \nabla^2 \psi) \quad (3.2)$$

Um die Helmholtzgleichung numerisch zu lösen, verwendet man das Relaxationsverfahren (vgl. Kapitel 3.3).

3.2. Der Laplaceoperator

Die Ableitung einer Funktion F an einem Punkt i und der Schrittweite Δx lässt sich mittels Vorwärtsdifferenz bestimmen:

$$\frac{dF}{dx} = \frac{F_{i+1} - F_i}{\Delta x} \quad (3.3)$$

Die zweite Ableitung lässt sich damit ebenfalls herleiten. Es ist

$$\begin{aligned} \frac{d^2 F}{dx^2} &= \frac{d}{dx} \frac{dF}{dx} \\ &= \frac{\frac{F_{i+1} - F_i}{\Delta x} - \frac{F_i - F_{i-1}}{\Delta x}}{\Delta x} \\ &= \frac{F_{i+1} - 2F_i + F_{i-1}}{(\Delta x)^2} \end{aligned} \quad (3.4)$$

¹Der Satz von Schwarz ermöglicht die Vertauschung von Laplaceoperator und Zeitableitung in (1.1).

Mit der Definition des Laplaceoperators

$$\nabla^2 = \frac{d^2}{dx^2} + \frac{d^2}{dy^2} \quad (3.5)$$

lässt sich dieser somit in finiten Differenzen ausdrücken als

$$\nabla^2 F(i, j) = \frac{F_{i+1} - 2F_i + F_{i-1}}{(\Delta x)^2} + \frac{F_{j+1} - 2F_j + F_{j-1}}{(\Delta y)^2} \quad (3.6)$$

Aufgrund der Tatsache, dass im Gitternetz des Modells $\Delta x \neq \Delta y$, findet Gleichung (3.6) auch in dieser Form Anwendung.

3.3. Das sukzessive (Über-)Relaxationsverfahren

Das Relaxationsverfahren wird eingesetzt, um eine Lösung für G in Gleichung (3.1) zu erhalten. Im betrachteten Fall wird also $\frac{\partial \psi}{\partial t}$ bestimmt. Zunächst wird die Gleichung in Differenzenquotienten aufgestellt. Dabei werden für räumliche Ableitungen in F zentrierte Differenzenquotienten verwendet. Man erhält mit $\frac{\partial \psi}{\partial t} = G$:

$$\underbrace{\beta \frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta x}}_{=\beta \frac{\partial \psi}{\partial x}} + \underbrace{\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta x} \frac{J_{i,j+1} - J_{i,j-1}}{2\Delta y} - \frac{J_{i+1,j} - J_{i-1,j}}{2\Delta x} \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2\Delta y}}_{\text{Jacobioperator } \mathcal{J}(\psi, J) \text{ mit } J = \nabla^2 \psi} = 0 \quad (3.7)$$

$=F$

Die Funktion F ist bekannt und lässt sich aus dem bekannten Anfangsfeld ψ berechnen. Der Jacobioperator steht nur formal in der Gleichung, da er sich zu Null ergibt und somit auch weggelassen werden kann. Multiplizieren der Gleichung mit dem Hauptnenner von (3.7) ergibt

$$(G_{i+1,j}^m - 2G_{i,j}^m + G_{i-1,j}^m)(\Delta y)^2 + (G_{i,j+1}^m - 2G_{i,j}^m + G_{i,j-1}^m)(\Delta x)^2 - HG_{i,j}^m(\Delta x)^2(\Delta y)^2 + F_{i,j}(\Delta x)^2(\Delta y)^2 = 0 \quad (3.8)$$

Dabei bezeichnet m den Iterationsschritt. G^m ist damit die m -te Schätzung von G . Das Verfahren wird mit einer „geratenen“ Anfangslösung initialisiert. In diesem Fall wird G Null gesetzt. Man erhält eine Lösung von (3.8), die von Null abweicht. Diese Abweichung wird als *Residuum* $R_{i,j}^m$ an der Stelle (i, j) für den m -ten Iterationsschritt definiert. Das Residuum ist also die Abweichung der numerischen von der analytischen Lösung der Gleichung. Mit dem Residuum lautet (3.8) nun

$$(G_{i+1,j}^m - 2G_{i,j}^m + G_{i-1,j}^m)(\Delta y)^2 + (G_{i,j+1}^m - 2G_{i,j}^m + G_{i,j-1}^m)(\Delta x)^2 - HG_{i,j}^m(\Delta x)^2(\Delta y)^2 + F_{i,j}(\Delta x)^2(\Delta y)^2 = R_{i,j}^m \quad (3.9)$$

Ziel ist es, das Residuum auf einen hinreichend kleinen Wert zu bringen, der als Lösung akzeptiert werden kann. Dazu führt man einen verbesserten Wert des nächsten Iterationsschrittes $G_{i,j}^{m+1}$ ein, für den das Residuum in (3.9) Null wird:

$$(G_{i+1,j}^m - 2G_{i,j}^{m+1} + G_{i-1,j}^m)(\Delta y)^2 + (G_{i,j+1}^m - 2G_{i,j}^{m+1} + G_{i,j-1}^m)(\Delta x)^2 - HG_{i,j}^{m+1}(\Delta x)^2(\Delta y)^2 + F_{i,j}(\Delta x)^2(\Delta y)^2 = 0 \quad (3.10)$$

Subtrahiert man (3.10) von (3.9), so erhält man eine Gleichung, mit der sich G des nächsten Iterationsschrittes bestimmen lässt:

$$G_{i,j}^{m+1} = G_{i,j}^m + \frac{R_{i,j}^m}{2((\Delta x)^2 + (\Delta y)^2) + H(\Delta x)^2(\Delta y)^2} \quad (3.11)$$

Um $G_{i,j}^{m+1}$ zu bestimmen, benötigt man die vier umliegenden Gitterpunkte. Es ergibt sich das Problem, dass bei zeilen- und spaltenweiser Iteration Werte dieser Gitterpunkte schon die Werte des $m + 1$ -ten Iterationsschrittes besitzen, während andere noch den Wert des m -ten Iterationsschrittes haben (vgl. Abb. 3.1).

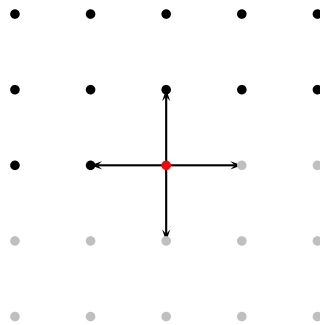


Abbildung 3.1.: Zeilenweise Iteration: Der Wert des roten Punktes wird aus Werten berechnet, die sowohl aus dem $m + 1$ -ten Iterationsschritt (schwarze Punkte) als auch aus dem m -ten Iterationsschritt (graue Punkte) stammen

Um dieses Problem zu umgehen, unterteilt man das Gitter in „gerade“ und „ungerade“ Felder, analog zu einem Schachbrett. Im ersten Schritt werden die geraden Felder berechnet. Im folgenden Schritt werden die ungeraden Felder aus den geraden Feldern berechnet und im

darauf folgenden Schritt die geraden Felder wieder aus den aktuellen ungeraden Feldern (vgl. Abb. 3.2).

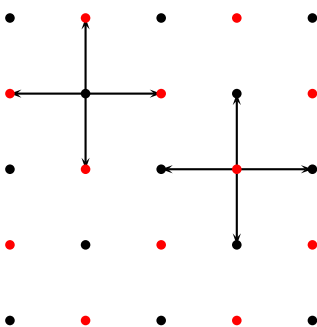


Abbildung 3.2.: Schachbrettmuster, das im Relaxationsverfahren verwendet wird. Werte der roten Felder berechnen sich aus denen der aktuellen schwarzen Felder und umgekehrt.

Das sukzessive Relaxationsverfahren hat die Eigenschaft, nur sehr langsam zu konvergieren. Aus diesem Grunde führt man in (3.11) einen Faktor α ein, um das Verfahren zu beschleunigen:

$$G_{i,j}^{m+1} = G_{i,j}^m + \alpha \frac{R_{i,j}^m}{2((\Delta x)^2 + (\Delta y)^2) + H(\Delta x)^2(\Delta y)^2}, \quad 1 \leq \alpha \leq 2 \quad (3.12)$$

Dieses Verfahren wird als Sukzessive Überrelaxation bezeichnet (engl. SOR, successive over relaxation). Der Wert von α ist unter anderem abhängig von der Form und den Koeffizienten der zu lösenden Gleichung. Weiterhin hängt es von der Anzahl der Gitterpunkte und den gewählten Gitterpunktabständen ab, für welche α das Verfahren konvergiert. α kann optimal gewählt werden mit

$$\alpha = \frac{2}{1 - \sqrt{1 - \rho_{\text{Jacobi}}^2}} \quad (3.13)$$

ρ_{Jacobi} bezeichnet dabei den Jacobiradius, in welchem die dem Relaxationsverfahren zugrundeliegende Jacobi-Iteration konvergiert. Es ist

$$\rho_{\text{Jacobi}} = \frac{\cos \frac{\pi}{I} + \left(\frac{\Delta x}{\Delta y}\right)^2 \cos \frac{\pi}{J}}{1 + \left(\frac{\Delta x}{\Delta y}\right)^2} \quad (3.14)$$

mit der Anzahl der Gitterpunkte I und J in x - bzw. y - Richtung und den Gitterpunktabständen Δx und Δy .

Die Kombination der Wahl des Parameters α mit der Anwendung des Schachbrettalgorithmus führt zu einem initialen Anwachsen des Fehlers um den Faktor 20. Erst nach einer bestimmten Anzahl von Iterationen, die in der Größenordnung von I liegt, konvergiert das Verfahren. Dieses Problem wird durch eine alternative Wahl von α gelöst. Es kommt die

Chebyshev-Beschleunigung zur Anwendung. Hierbei wird α nach jeder halben Iteration im Schachbrettalgorithmus neu gewählt:

$$\begin{aligned}\alpha_0 &= 1 \\ \alpha_{1/2} &= \frac{1}{1 - \frac{\rho_{\text{jacobi}}^2}{2}} \\ \alpha_{n+1/2} &= \frac{1}{1 - \frac{\alpha_n \rho_{\text{jacobi}}^2}{4}}, \quad n = k/2, k \in \mathbb{N}_+\end{aligned}$$

Für unendlich viele Iterationen ergibt sich dann

$$\alpha_\infty = \alpha_{\text{optimal}}$$

Der Algorithmus für das SOR-Verfahren ist in Press et al., Seite 860 gegeben gewesen und musste noch an das Problem angepasst werden².

3.4. Zeitliche Integrationsverfahren

Nachdem $G = \frac{\partial \psi}{\partial t}$ mittels des Relaxationsverfahrens bestimmt wurde, muss noch numerisch nach der Zeit integriert werden, um ψ zu erhalten. Dabei kommt das Leapfrogschema zur Anwendung. Zur Berechnung von ψ^{t+1} muss aber ein Eulerschritt verwendet werden, da ψ^{-1} nicht bekannt ist. Das Eulerverfahren ist analog zur Vorwärtsdifferenz:

$$\frac{\partial \psi}{\partial t} = G = \frac{\psi^{t+1} - \psi^t}{\Delta t} \quad (3.15)$$

Damit ergibt sich ψ^{t+1} aus

$$\psi^{t+1} = \psi^t + G \cdot \Delta t \quad (3.16)$$

Das Leapfrogverfahren ist genauer, da es auf der zentrierten Differenz basiert:

$$\frac{\partial \psi}{\partial t} = G = \frac{\psi^{t+1} - \psi^{t-1}}{2\Delta t} \quad (3.17)$$

Dies lässt sich auch schreiben als

$$G = \frac{\psi^{t+2} - \psi^t}{2\Delta t}, \quad (3.18)$$

so dass sich der übernächste Zeitschritt aus dem aktuellen ergibt mit

$$\psi^{t+2} = \psi^t + G \cdot 2\Delta t \quad (3.19)$$

²vgl. Anhang

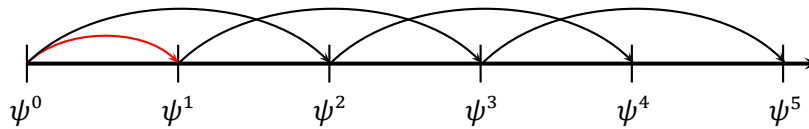


Abbildung 3.3.: Schema zum Euler- (rot) und Leapfrog-Verfahren (schwarz)

Es zeigt sich, dass die Genauigkeit der einzelnen Zeitschritte alterniert, da die Hälfte der Felder ψ direkt aus ψ^0 berechnet wurden und die andere Hälfte auf ψ^1 basiert, welches erst mit dem ungenaueren Eulerverfahren ermittelt werden musste (vgl. Abb. 3.3). Um den Fehler klein zu halten, kann man für den Eulerschritt einen kleineren als den ursprünglich gewählten Zeitschritt Δt verwenden. Es ist darauf zu achten, dass ein Vielfaches des Euler-Zeitschrittes wieder den originalen Zeitschritt ergibt. Für die folgenden Leapfrogschritte kann dann der Zeitschritt wieder an das ursprünglich gewählte Δt angepasst werden. In unserem Modell haben wir auf eine solche Maßnahme verzichtet und generell einen hinreichend kleinen Zeitschritt von $\Delta t = 1800$ s gewählt. Die heutige Rechentechnik erlaubt dies ohne signifikanten Zuwachs der Rechenzeit bei einer solchen Modellgröße.

3.5. Umsetzung der analytischen Lösung

Gegeben ist die Lösung (2.12). Diese kann numerisch direkt mit der ebenfalls gegebenen Testfunktion umgesetzt werden:

$$\psi = -\bar{U} \cdot n \cdot \Delta y + A[\sin(k^* \cdot m \cdot \Delta x - \omega^* t) + \cos(k^* \cdot m \cdot \Delta x - \omega^* t)] \sin(l^* \cdot n \cdot \Delta y) \quad (3.20)$$

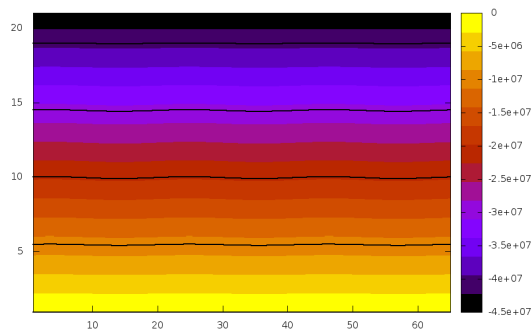
Hierbei bezeichnen m und n die Iterationsschritte in x - und y -Richtung. k^* und l^* sind die modifizierten Wellenzahlen aus Gleichung (2.11). ω^* ist die Phasengeschwindigkeit, die sich aus den modifizierten Wellenzahlen errechnet:

$$\omega^* = \frac{\bar{U}k^*(k^{*2} + l^{*2}) - \beta k^*}{H + k^{*2} + l^{*2}} \quad (3.21)$$

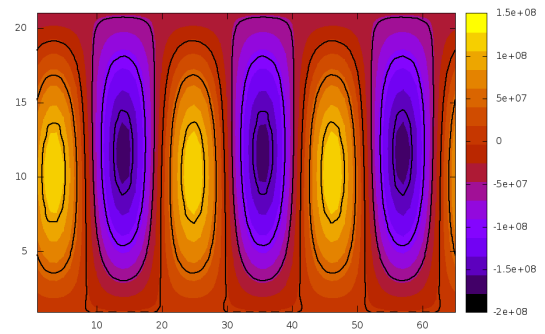
Die Phasengeschwindigkeit in x -Richtung beträgt somit

$$c_x^* = \frac{\bar{U}(k^{*2} + l^{*2}) - \beta}{H + k^{*2} + l^{*2}} \quad (3.22)$$

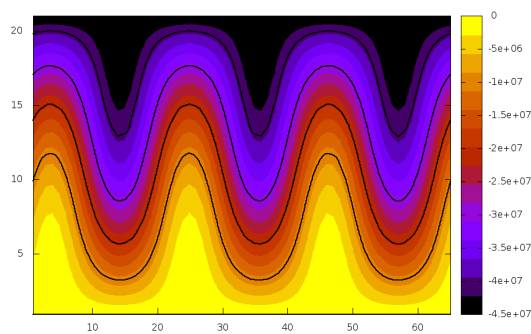
Damit lässt sich die Verlagerungsgeschwindigkeit in Einheiten von Gitternetzpunkten pro Zeit angeben. Dies ist der Überprüfung der Vorhersage dienlich. Von Bedeutung ist die Wahl der Amplitude A . Diese beeinflusst die Größenordnung des Störungsterms gegenüber dem Grundstromterm $\bar{U} \cdot j \cdot \Delta y$. Ist A zu klein, sind die Rossbywellen zu flach, um aussagekräftige Ergebnisse zu erhalten. Ist A dagegen zu groß, überlagern sich die Wellen derart, dass sich abgeschlossene Gebiete anstelle von Wellen bilden. Es zeigt sich empirisch, dass eine Wahl von $A = 10^6$ günstig ist (vgl. Abb. 3.4).



(a) $A = 10^4$



(b) $A = 10^7$



(c) $A = 10^6$

Abbildung 3.4.: Ergebnisse bei zu kleiner, zu großer und korrekter Wahl der Amplitude A

4. Ergebnisse

4.1. Vorhersage einer einzelnen Rossbywelle

Um das Programm zu testen wird die in der Aufgabenstellung gegebene Testfunktion (1.2)

$$\psi = A \left[\sin\left(\frac{2\pi km}{64}\right) + \cos\left(\frac{2\pi km}{64}\right) \right] \sin\left(\frac{\pi ln}{20}\right) - \bar{U}n \cdot \Delta y$$

genutzt. Es ist klar, dass es zu Unterschieden zwischen der numerischen und der analytischen Lösung kommt. Dieser Unterschied sollte nach 24 Stunden aber noch gering sein. Für den Vergleich wurden die Wellenzahlen $k = 2$ und $l = 1$ festgelegt. Der Grundstrom wurde mit $\bar{U} = 5 \text{ m s}^{-1}$ gewählt. Das Ergebnis ist in Abbildung 4.1 dargestellt. Aus der Phasengeschwindigkeit (3.22) erhält man eine Verlagerung von -1.46 Gitterpunkten, das heißt die Welle bewegt sich retrograd von Ost nach West bzw. in der Abbildung von rechts nach links. Ein positives Vorzeichen entspräche einer progressiven Bewegung von West nach Ost.

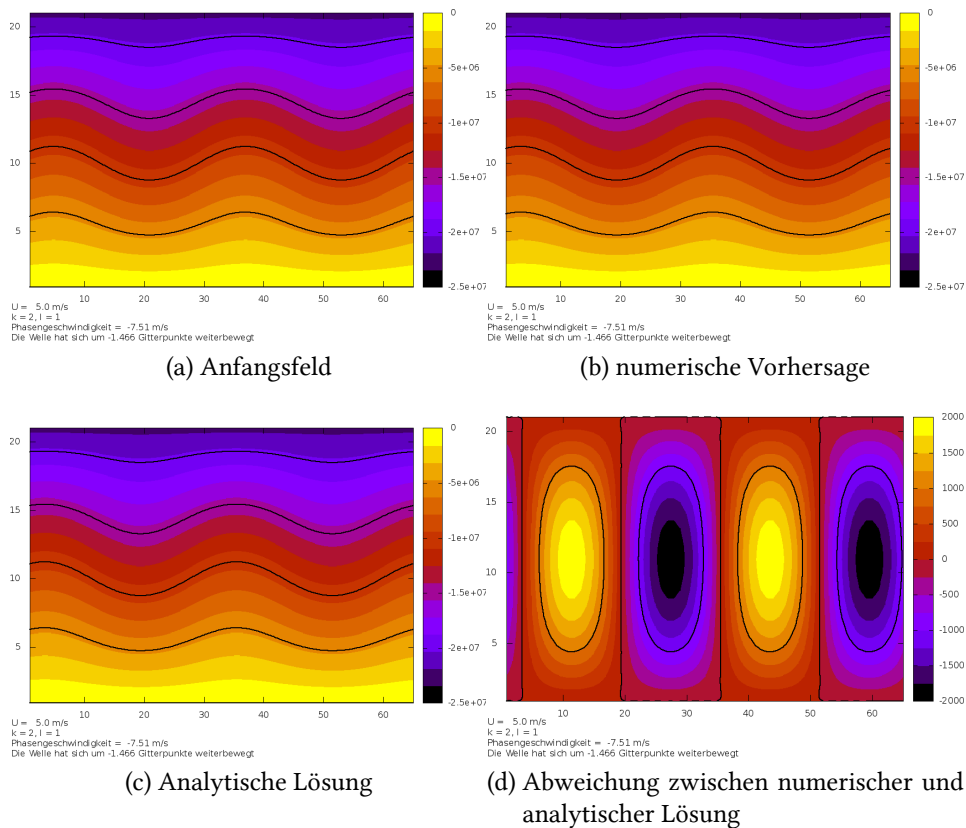


Abbildung 4.1.: Vorhersage einer Rossbywelle mit Wellenzahlen $k = 2, l = 1$

Sowohl bei der numerischen als auch der analytischen Lösung ist die Verlagerung der Welle so groß wie berechnet. In der grafischen Darstellung der Differenz zwischen numerischer und analytischer Lösung (Abb. 4.1d). Die geringste Abweichung liegt an den Amplituden der Wellen vor, die größten Abweichungen befinden sich dazwischen. Insgesamt liegt der Fehler in der Größenordnung von 0.1%. Dies lässt sich durch einen Skalenvergleich bestimmen, in dem man die dargestellten Werte der berechneten Welle und der Differenz ins Verhältnis setzt.

Anhand eines weiteren Beispiels ($k = 7, l = 4, \bar{U} = 15 \text{ m s}^{-1}$) soll gezeigt werden, dass der zonal gemittelte Zonalwind und die gemittelte Vorticity im zeitlichen Verlauf konstant bleiben. Dies sind wesentliche Eigenschaften des Modells, die sich aus der Barotropie und der Tatsache, dass in einem Kanal mit festen Rändern gerechnet wird, ergeben. Zunächst ist das berechnete Feld dargestellt (Abb. 4.2). Die Dispersionsgeschwindigkeit ist positiv und betragsmäßig höher als im ersten Beispiel. Dies ist durch die Wahl einer hohen Grundstromgeschwindigkeit sowie größerer Wellenzahlen das erwartete Resultat.

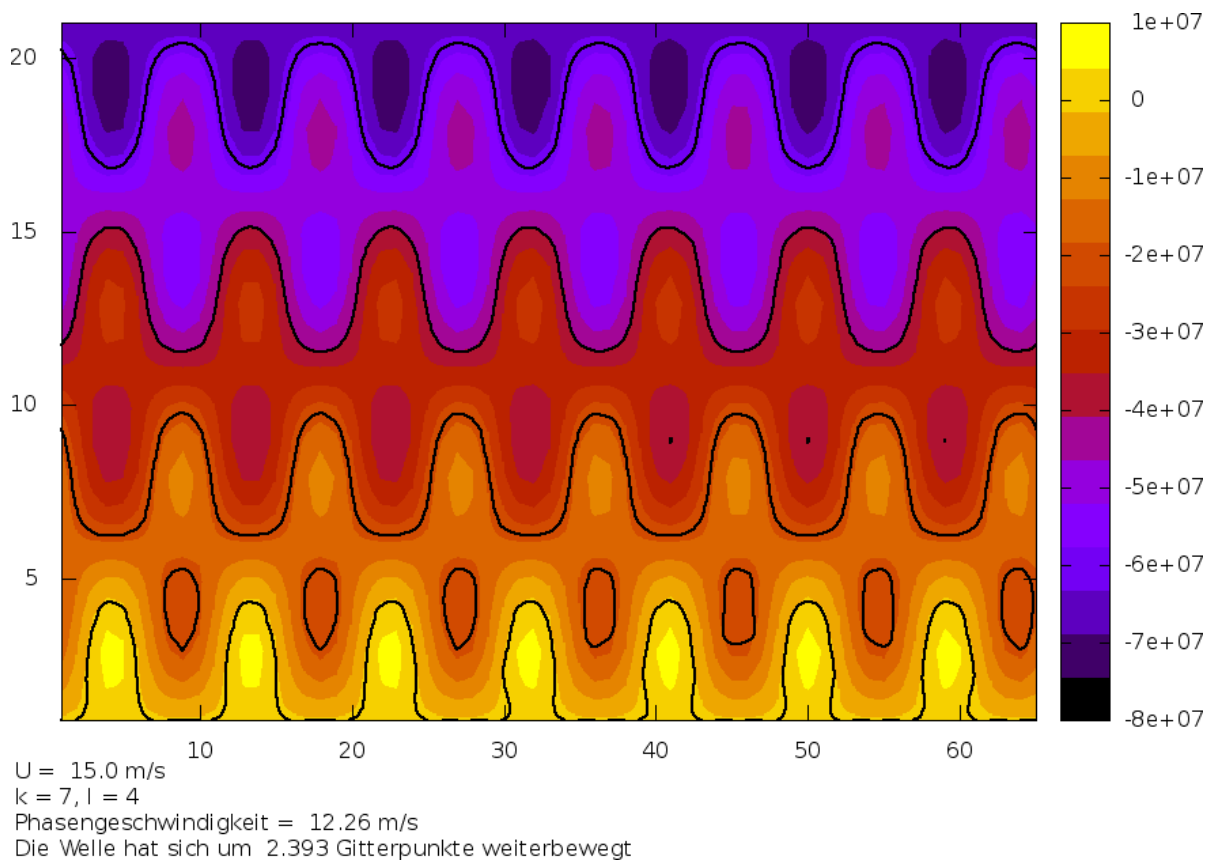


Abbildung 4.2.: 24 h-Vorhersage mit den Parametern $k = 4, l = 7, U = 15 \text{ m s}^{-1}$

Der Zonalwind berechnet sich mit $U = -\frac{\partial\psi}{\partial y}$. Es muss demnach gelten:

$$\frac{1}{dx} \int -\frac{\partial\psi}{\partial y} dx = \text{const.} \quad (4.1)$$

In der folgenden Abbildung (4.3) ist der zeitliche Verlauf des zonal gemittelten Zonalwindes dargestellt.

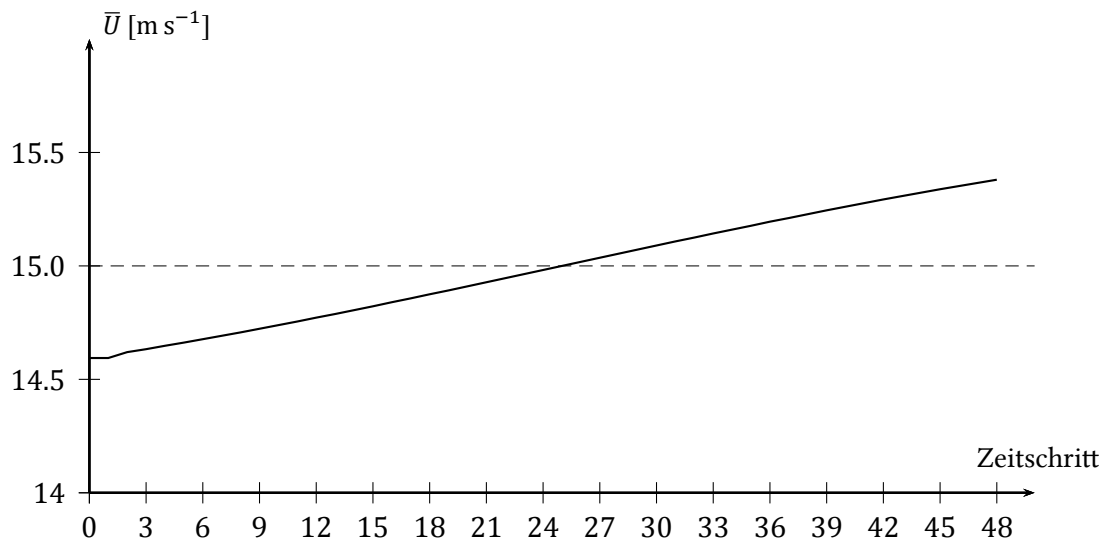


Abbildung 4.3.: Zonal gemittelter Zonalwind auf einem festen Breitenkreis im zeitlichen Verlauf

Es zeigt sich eine symmetrische Abweichung von der eingestellten Grundstromgeschwindigkeit. Grund dafür ist die Zonalbewegung der Rossbywellen. Bei einer Fortsetzung (d.h. Erhöhung der Zeitschritte) ergäbe sich demnach eine Sinuskurve mit dem Mittelwert \bar{U} . Dies lässt sich einfach nachweisen, indem man die Integrationszeit vervielfacht und sich dann die Kurve anschaut.

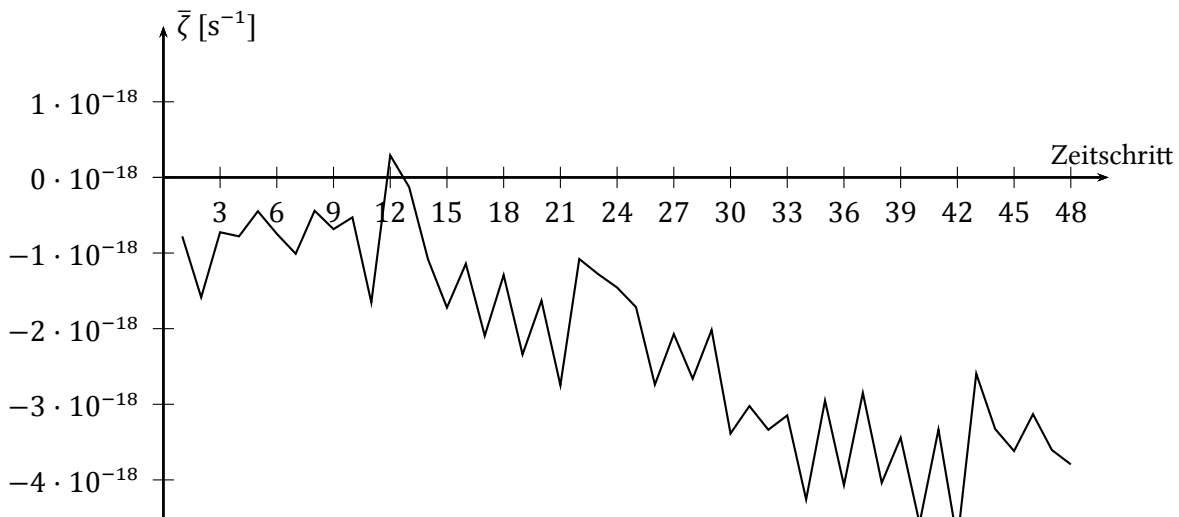
Die Rossbywellentheorie besagt, dass bei barotropen Verhältnissen kein Vorticityzuwachs oder -abbau stattfindet, das heißt, es gilt

$$\frac{\partial \zeta}{\partial t} = 0 \quad (4.2)$$

Wegen $\zeta = \nabla^2 \psi$ lässt sich auch diese Eigenschaft überprüfen. Die gemittelte Vorticity

$$\bar{\zeta} = \int \zeta \, dx \, dy \quad (4.3)$$

ist in Abbildung 4.4 dargestellt:

Abbildung 4.4.: Gemittelte Vorticity $\bar{\zeta}$ im zeitlichen Verlauf

Die Abweichung von Null wird durch numerisches Rauschen verursacht und führt analog zum Zonalwind zu einer Sinusschwingung um Null. Ohnehin ist sie derart klein (Größenordnung 10^{-18}), dass sie vernachlässigt werden kann.

Da nun nachgewiesen ist, dass das numerische Lösungsverfahren nur sehr gering von dem analytischen Lösungsverfahren abweicht und sich das berechnete Feld wie erwartet verhält, erschließt sich, dass das Programm korrekt arbeitet und so eine Prognose mit gegebenem Anfangsfeld möglich ist.

4.2. Vorhersage eines gegebenen Anfangsfeldes

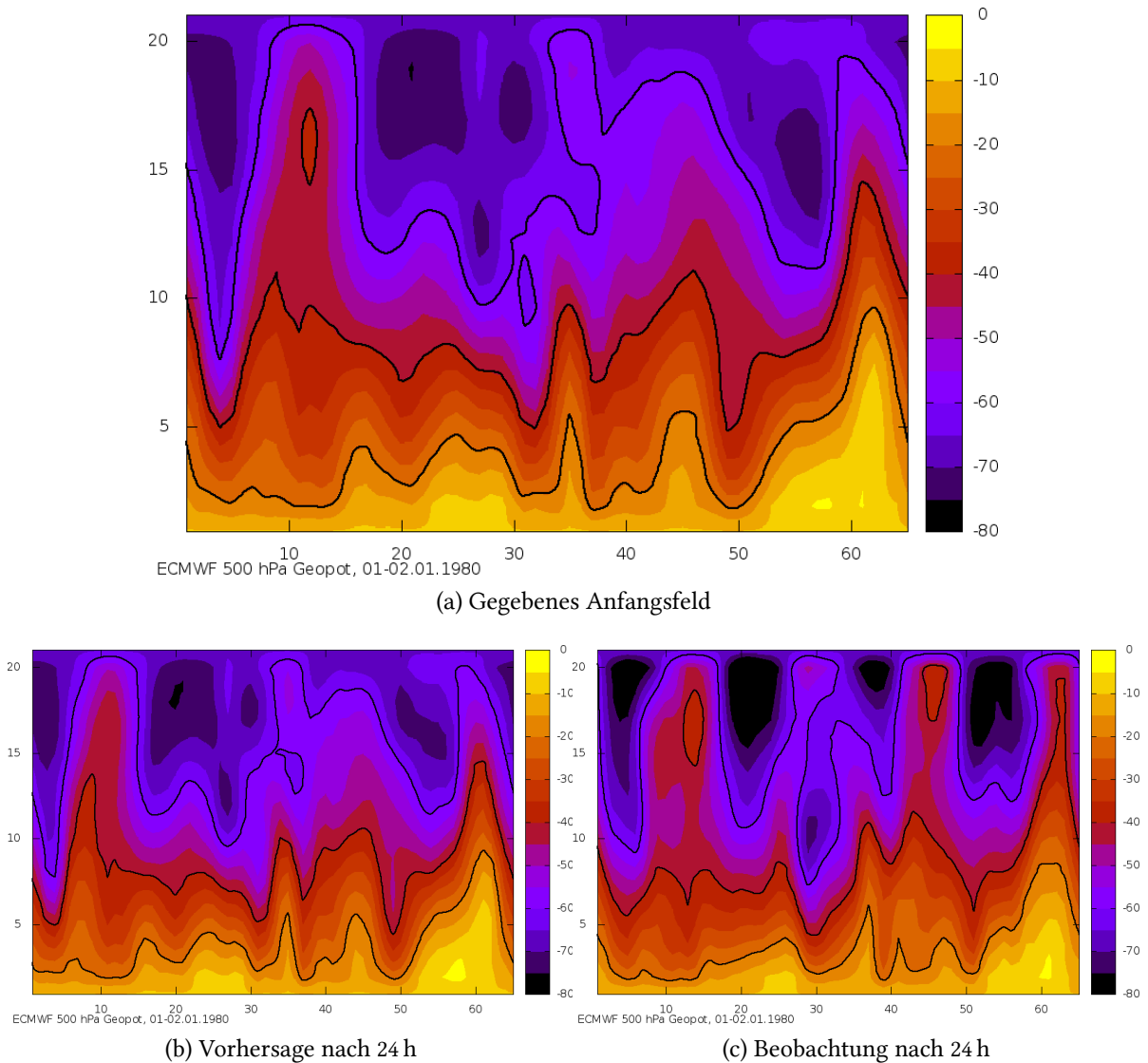


Abbildung 4.5.: 24 h-Vorhersage des gegebenen Anfangsfeldes

In Abbildung 4.5 sind das gegebene Anfangsfeld, die Vorhersage nach 24 Stunden sowie die Analyse nach 24 Stunden dargestellt. Im Gegensatz zu der Vorhersage mittels Testfunktion, bei der man die analytische Lösung praktisch als Beobachtung zu diesem Zeitpunkt ansehen kann, erkennt man deutliche Unterschiede zwischen prognostiziertem und beobachtetem Feld. Dies ist auch zu erwarten, da ein barotropes Einschichtenmodell in einem Kanal auf der β -Ebene die real ablaufenden, komplexen atmosphärischen Prozesse nicht ausreichend erfassen kann.

In Abbildung 4.5 ist zu erkennen, dass sich bei der numerischen Berechnung das Feld nach

Westen, aber in der Realität nach Osten fortbewegte. Des Weiteren sieht man, dass die Veränderung des Feldes (Keilbildung und Austrognung) viel stärker eintraf als numerisch berechnet. Ein weiterer Vergleich ist mit Hilfe des zonal gemittelten Zonalwindes möglich. In Abbildung 4.6 erkennt man, dass der zonal gemittelte Zonalwind eine Nord-Südverteilung aufweist. Der zonal gemittelte Zonalwind nimmt von Süd nach Nord zu. Ein Grund dafür ist die höhere Wellenaktivität in den nördlichen Breiten an der winterlichen Polarfront. Die Ausgangswerte des gegebenen Feldes und die Werte der numerischen Lösung nach 24 Stunden sind fast deckungsgleich. Dies liegt an unseren Annahmen des Modells, dass der zonal gemittelte Zonalwind konstant ist. Dies ist in der Realität aber nicht der Fall. So weicht der tatsächliche zonal gemittelte Zonalwind vom numerisch berechneten ab.

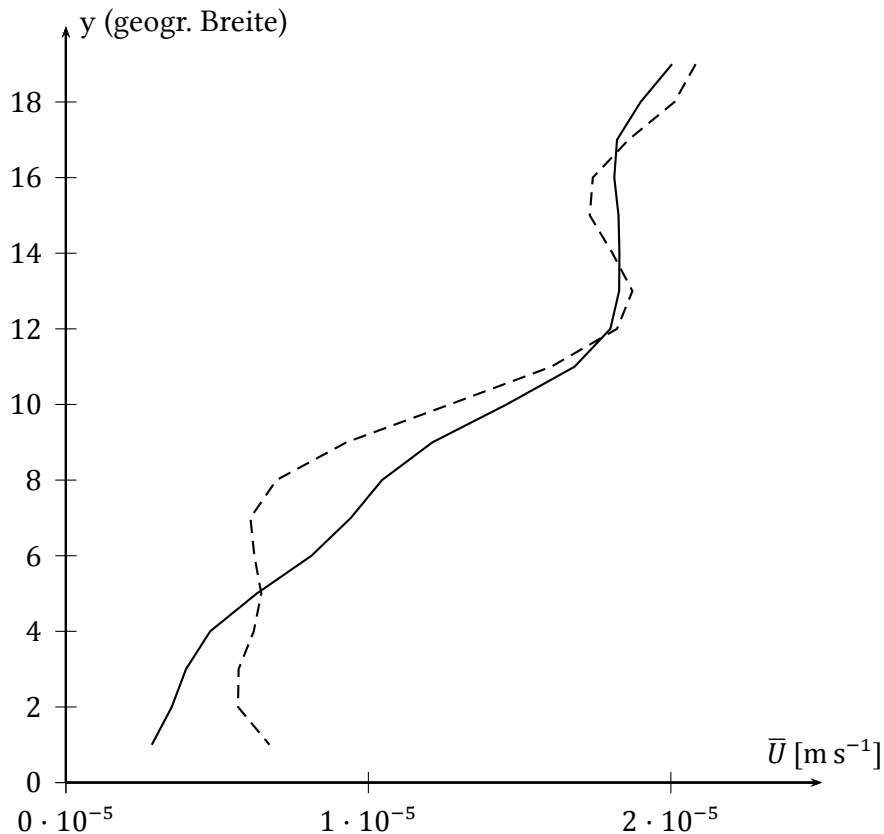
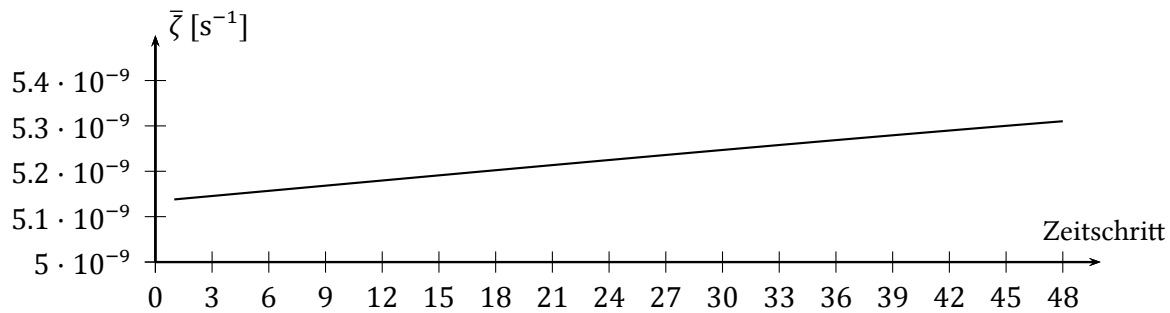


Abbildung 4.6.: Vergleich der gemittelten Zonalwinde für Vorhersage und Analyse. Durchgezogene Linie: Vorhersage. Unterbrochene Linie: Analyse.

Aufgrund der meridional verschieden großen Zonalwinde und der damit verbundenen Scherung ist nun eine mittlere Vorticity zu erwarten, die verschieden von Null ist. Wie Abbildung 4.7 zeigt, ist dies auch der Fall.

Abbildung 4.7.: Gemittelte Vorticity $\bar{\zeta}$ im zeitlichen Verlauf

Auch hier kommt es durch numerisches Rauschen zu einer Sinusschwingung um den genauen Mittelwert, die den Anstieg erklärt. Ein Vergleich mit dem analysierten Feld zeigt auch hier die Abweichung der Vorhersage. In der Analyse ist $\bar{\zeta} = 1.36 \cdot 10^{-8} s^{-1}$.

Die Abweichungen zwischen numerischen und beobachteten Werten resultieren aus mehreren Annahmen, die zu Fehlern führen.

1. Ein ungenaues Gitter und die Betrachtung von nur einer Schicht der Atmosphäre führen zu einer schlechten Auflösung.
2. Durch die Annahme der Barotropie fehlt ein wesentlicher Antrieb (Baroklinität) der atmosphärischen Zirkulation.
3. Durch die Festlegung auf einen Kanal werden die Entwicklungen an den Rändern nicht berücksichtigt. Dies ist vor allem im Norden gerade im Winter zu erkennen, da sich in der Frontalzone intensive Entwicklungen abspielen. Dies ist z.B. an der Austrogung im Gitterpunkt $x = 20$ gut zu erkennen. Die beobachteten Werte sind um einiges geringer als die vorhergesagten, was an einer Advektion von außerhalb der Kanalgrenzen liegt.

Zusammenfassend kann man sagen, dass dieses einfache Modell sich nicht zur genauen Wettervorhersage eignet, da die atmosphärischen Entwicklungen zu komplex sind und mit diesen Annahmen die atmosphärischen Prozesse nicht ausreichend erfasst werden können. Es ist lediglich möglich vereinfachte theoretische Überlegungen zu überprüfen. Historisch gesehen war es trotz dessen ein großer Erfolg.

A. Quellcode

Im Folgenden werden alle erforderlichen Programmteile gegliedert nach ihrer Funktion aufgeführt.

A.1. Präambel

```
parameter (imax=65, jmax=20)
integer k, l, dt
double precision beta, dx, dy, h, a, pi, u, rjac, kk, ll, w, c
double precision f(0:imax,0:jmax), g(0:imax,0:jmax),
& psi(0:imax,0:jmax), psi2(0:imax,0:jmax), psi3(0:imax,0:jmax),
& psi_a(0:imax,0:jmax)
character(25) dummy
character(2) datei
pi = 4.d0*datan(1.d0)
beta = 1.62d-11
H = 1/(1d6)**2
dx = 442550d0
dy = 222530d0
dt = 1800
rjac = (cos(pi/(imax-1))+dx/dy)**2*cos(pi/jmax)/(1+(dx/dy)**2)
```

A.2. Deklaration der Testfunktion

```
k = 7
l = 4
a = 1d7
u = 15.d0
kk = 2*pi*k/(dx*(imax-1))
ll = pi*l/(dy*(jmax))
w = (u*kk*(kk**2+ll**2)-beta*kk)/(kk**2+ll**2+H)
c = w/kk

do i=0,imax
```

```

do j=0,jmax
  psi(i,j)=a*(cos(2*pi*k*i/(imax-1))+sin(2*pi*k*i/(imax-1)))*
&   sin(pi*l*j/jmax)-u*j*dy
  psi_a(i,j)=a*(cos(kk*i*dx-w*86400)+sin(kk*i*dx
&   -w*86400))*sin(l1*j*dy)-u*j*dy
  g(i,j) = 0.d0
enddo
enddo

open(11,file='psi.dat')
open(13,file='psi_a.dat')
do j=0,jmax
  write(11,'(66(G12.5,X))') (psi(i,j),i=0,imax)
  write(13,'(66(G12.5,X))') (psi_a(i,j),i=0,imax)
enddo
close(11)
close(13)

write(*,*) 'U =',u, 'm/s'
write(*,*) 'k =', k, ', l =', l
write(*,*) 'Phasengeschwindigkeit =',c,'m/s'
write(*,*) 'Die Welle hat sich um', c*86400/dx
&,'Gitterpunkte weiterbewegt'

```

A.3. Einlesen des Anfangsfeldes

```

open(15,file='SF500.800101.NH.bc') ! Datei zum Auslesen oeffnen
do i=1,2
  read(15,*,end=100) dummy ! Erste zwei Zeilen in der Datei ueberspringen
enddo
! Auslesen der Werte
do i = 0,20
  k = 0
  do j = 1,9 !Letzte Zeile (10) in i-Richtung hat nur 2 Eintraege
    read(15,*,end=100) (psi(l,i),l=k,k+6)
    k = k+7
  enddo
  read(15,*) psi(63,i), psi(64,i)
enddo
100 close(15)

do j = 0,jmax

```

```

        psi(imax,j) = psi(1,j)
        psi(0,j) = psi(imax-1,j)
    enddo

    open(11,file='psi.dat')
    do j=0,jmax
        write(11,'(66(G12.5,X))') (psi(i,j),i=0,imax)
    enddo
    close(11)

    open(16,file='SF500.800102.NH.bc') ! Datei zum Auslesen oeffnen
    do i=1,2
        read(16,*,end=101) dummy ! Erste zwei Zeilen in der Datei ueberspringen
    enddo
    ! Auslesen der Werte
    do i = 0,20
        k = 0
        do j = 1,9 !Letzte Zeile (10) in i-Richtung hat nur 2 Eintraege
            read(16,*,end=100) (psi_a(1,i),l=k,k+6)
            k = k+7
        enddo
        read(16,*) psi_a(63,i), psi_a(64,i)
    enddo
101 close(16)

    open(13,file='psi_a.dat')
    do j=0,jmax
        write(13,'(66(G12.5,X))') (psi_a(i,j),i=0,imax)
    enddo
    close(13)

```

A.4. Integration über die Zeit

!====Erster Zeitschritt mit Euler=====

```

    call rest(psi,f,dx,dy,imax,jmax,0)
    call sor(f,g,dx,dy,imax,jmax,rjac)
    call euler(psi,psi2,g,imax,jmax,dt)

    open(20,file='psi_0.dat')
    do j=0,jmax
        write(20,'(66(G12.5,X))') (psi2(i,j),i=0,imax)
    enddo

```

```

        enddo
        close(20)

!=====Restliche Integration mit Leapfrog=====

        do m=1,47
            call rest(psi2,f,dx,dy,imax,jmax,m)
            call sor(f,g,dx,dy,imax,jmax,rjac)
            call leapfrog(psi,psi3,g,imax,jmax,dt)

            do i=0,imax
                do j=0,jmax
                    psi(i,j) = psi2(i,j)
                    psi2(i,j) = psi3(i,j)
                enddo
            enddo
            write(datei,'(I2.2)') m
            open(20,file='psi_//datei//'.dat')
            do j=0,jmax
                write(20,'(66(G12.5,X))') (psi2(i,j),i=0,imax)
! Schreibe alle Felder in jeweils eine einzelne Datei wg. Plot
            enddo
            close(20)
        enddo

        open(12,file='psi2.dat')
        open(14,file='psi_diff.dat')
        do j=0,jmax
! Schreibe berechnetes Feld in eine Datei
            write(12,'(66(G12.5,X))') (psi2(i,j),i=0,imax)
! Schreibe Differenz zw. num. u. anal. Lsg. in Datei
            write(14,'(66(G12.5,X))') (psi2(i,j)-psi_a(i,j),i=0,imax)
        enddo

        end program

```

A.5. Sukzessive Überrelaxation

```

subroutine sor(f,g,dx,dy,imax,jmax,rjac)

integer jmax, maxits, ipass, i, isw, j, jsw, n

```

```

double precision rjac, f(0:imax,0:jmax),g(0:imax,0:jmax),eps,
& anorm, anormf, omega, resid, h, dx, dy
parameter(maxits=1000, eps=1.d-5)

h = 1.d0/(1.d6)**2
anormf = 0.d0
do i=2,imax-1
  do j=2,jmax-1
    anormf=anormf+abs(f(i,j))
  enddo
enddo
omega = 1.d0
do n=1,maxits
  anorm=0.d0
  isw=1
  do ipass=1,2
    jsw=isw
    do i=1,imax-1
      do j=jsw,jmax-1,2
        resid=(g(i+1,j)+g(i-1,j)-2*g(i,j))*dy**2
&          +(g(i,j-1)+g(i,j+1)-2*g(i,j))*dx**2
&          -h*g(i,j)*dx**2*dy**2
&          -f(i,j)*dx**2*dy**2
        anorm=anorm+abs(resid)/(dx**2*dy**2)
        g(i,j) = g(i,j)+omega
&          *resid/(2*dx**2+2*dy**2
&          +h*dx**2*dy**2)
      enddo
      jsw=3-jsw
    enddo
    isw=3-isw
    if(n.eq.1.and.ipass.eq.1) then
      omega=1.d0/(1.d0-.5d0*rjac**2)
    else
      omega=1.d0/(1.d0-.25d0*rjac**2*omega)
    endif
  enddo
  do j=0,jmax
    g(0,j) = g(imax-1,j)
    g(imax,j) = g(1,j)
  enddo
  if(anorm.lt.eps*anormf) then
    write(*,*) 'SOR konvergiert nach',n,'Schritten.'
    return
  endif
enddo

```



```

    endif
  enddo
write(*,*) 'MAXITS exceeded!'
end subroutine

```

A.6. Berechnung der Restfunktion F

```

subroutine Rest(psi,f,dx,dy,imax,jmax,time)

double precision psi(0:imax,0:jmax), f(0:imax,0:jmax), dx, dy,
& beta, laplace(0:imax,0:jmax), jacobi(0:imax,0:jmax)
integer imax,jmax,time

beta = 1.62d-11

do i=0,imax
  do j=0,jmax
    laplace(i,j) = 0.d0
  enddo
enddo

! Berechne Laplace
do i = 1,imax-1
  do j = 1,jmax-1
    laplace(i,j) = (psi(i-1,j)+psi(i+1,j)-2*psi(i,j))/(dx*dx) +
& (psi(i,j-1)+psi(i,j+1)-2*psi(i,j))/(dy*dy)
  enddo
enddo

do j=0,jmax
  laplace(0,j) = laplace(imax-1,j)
  laplace(imax,j) = laplace(1,j)
enddo

! Berechne den Jacobioperator
do i=1,imax-1
  do j=1,jmax-1
    jacobi(i,j)=((psi(i+1,j)-psi(i-1,j))/(2*dx))
& *((laplace(i,j+1)-laplace(i,j-1))/(2*dy))
& -(((psi(i,j+1)-psi(i,j-1))/(2*dy))
& *((laplace(i+1,j)-laplace(i-1,j))/(2*dx)))
  enddo

```

```
    enddo

    do i=1,imax-1
        do j=1,jmax-1
            f(i,j) = -beta*(psi(i+1,j)-psi(i-1,j))/(2.d0*dx)
&            -jacobi(i,j)
        enddo
    enddo

    do j=0,jmax
        f(0,j) = f(imax-1,j)
        f(imax,j) = f(1,j)
    enddo

end subroutine
```

A.7. Euler-Schritt

```
subroutine euler(psi,psi2,g,imax,jmax,dt)

double precision psi(0:imax,0:jmax), psi2(0:imax,0:jmax),
& g(0:imax,0:jmax)
integer dt, imax, jmax

do i=0,imax-1
    do j=0,jmax
        psi2(i,j) = psi(i,j) + dt*g(i,j)
    enddo
enddo

do j=0,jmax
    psi2(0,j) = psi2(imax-1,j)
    psi2(imax,j) = psi2(1,j)
enddo

end subroutine
```

A.8. Leapfrog-Schritt

```
subroutine leapfrog(psi,psi2,g,imax,jmax,dt)
```

```
double precision psi(0:imax,0:jmax),psi2(0:imax,0:jmax),
& g(0:imax,0:jmax)
integer dt, imax, jmax

do i=0,imax-1
  do j=0,jmax
    psi2(i,j) = psi(i,j) + 2*dt*g(i,j)
  enddo
enddo

do j=0,jmax
  psi2(0,j) = psi2(imax-1,j)
  psi2(imax,j) = psi2(1,j)
enddo

end subroutine
```

Literaturverzeichnis

[Haltiner und Williams 1980] HALTINER, George J. ; WILLIAMS, Roger T.: *Numerical Prediction and Dynamic Meteorology, 2nd Edition*. 2. Wiley, 5 1980. -- ISBN 9780471059714

[Press u. a. 1992] PRESS, William H. ; FLANNERY, Brian P. ; TEUKOLSKY, Saul A. ; VETTERLING, William T.: *Numerical Recipes in Fortran 77: The Art of Scientific Computing*. 2. Cambridge University Press, 9 1992. -- ISBN 9780521430647

[Universität Hamburg 2011] UNIVERSITÄT HAMBURG: *Übung zur Meteorologischen Modellierung (Numerik): Das barotrope Modell*. 2011. -- www.mi.uni-hamburg.de/uploads/media/uebung_barotrop.pdf (22.02.2011)