Meteorologische Modellierung

Introduction to Numerical Modeling

(of the Global Atmospheric Circulation)

Part II (2^{nd} Semester)

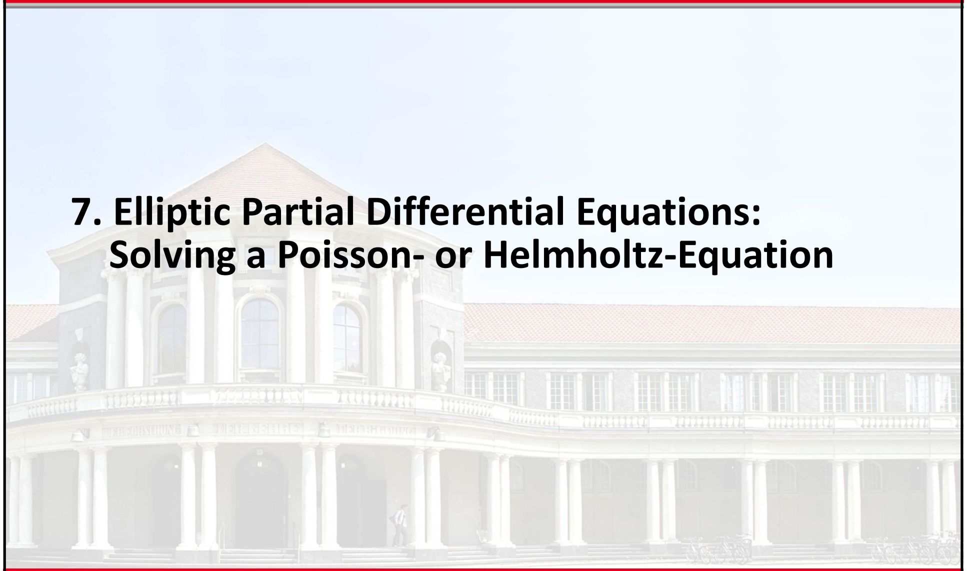
Frank Lunkeit

Outline

7. Elliptic Partial Differential Equations: Solving a Poisson- or Helmholtz-Equation

8. An example: qg barotropic channel (weather prediction)

Introduction to Numerical Modeling



7. Elliptic Partial Differential Equations: Solving a Poisson- or Helmholtz-Equation

Elliptic Partial Differential Equations: Solving a Poisson- or Helmholtz-Equation

$$a\frac{\partial^2 T}{\partial t^2}+b\frac{\partial^2 T}{\partial t \partial x}+c\frac{\partial^2 T}{\partial x^2}+d\frac{\partial T}{\partial t}+e\frac{\partial T}{\partial x}+f=0$$

hyperbolic:	b^2-$4ac > 0$
parabolic:	b^2-$4ac = 0$
elliptic:	b^2-$4ac < 0$

Elliptic equations: boundary value problem

Laplace: $\nabla^2\Theta = 0$

Poisson: $\nabla^2\Theta = G(x,y)$

Helmholtz: $(\nabla^2 + \lambda)\Theta = G(x,y)$

G, λ known; Θ wanted (e.g. G=vorticity; Θ =streamfunction)

Solving a Poisson-Equation: Spectral Method

$$\nabla^2\Theta = G(x,y) \qquad (1)$$

Start: Simple form of the inverse Laplace operator in spectral space (Fourier):

$$\hat{\Theta}(k,l) = \nabla^{-2}\hat{G}(k,l) = -K^{-2}\hat{G}(k,l) = -\frac{1}{k^2+l^2}\hat{G}(k,l) \qquad (2)$$

with $\hat{G}(k,l), \hat{\Theta}(k,l)$ = spectral transformed of $G(x,y)$ and Θ(x,y)

k = wave number in x

l = wave number in y

Note: Trigonometric functions (Fourier, plane) or Legendere polynomials (sphere) are Eigenfunctions of the Laplace operator, i.e. $\nabla^2\mathrm{E} = \lambda\mathrm{E}$

=> **Solution of (1):**

- transform $G(x,y)$ to spectral space
- compute $\hat{\Theta}(k,l)$ from (2)
- transform $\hat{\Theta}(k,l)$ to grid point space

Solving a Poisson-Equation: Grid Point Method

Start: discretization of $\nabla^2\Theta = G(x,y)$

x →, y ↑

6	2	5
3	0	1
7	4	8

grid point nomination

For point 0:

$$(\nabla^2\Theta)_0 - G_0 = \frac{1}{\Delta x^2}(\Theta_1 + \Theta_3 - 2\Theta_0) + \frac{1}{\Delta y^2}(\Theta_2 + \Theta_4 - 2\Theta_0) - G_0$$

$$\overset{!}{=} 0$$

NOTE: grid points interdependent -> no direct solution!

Solving a Poisson-Equation: Grid Point Method: Jacobi Method

Discrete Poisson-equation:

$$\nabla^2\Theta_0 - G_0 = \frac{1}{\Delta x^2}(\Theta_1+\Theta_3-2\Theta_0)+\frac{1}{\Delta y^2}(\Theta_2+\Theta_4-2\Theta_0)-G_0=0 \quad (1)$$

Solution by Iteration (Jacobi Method):

1. choose initial field (e.g. Θ = 0. or Θ = old (known) values)
2. compute error ε_0 for each grid point: $\varepsilon_0 = \nabla^2\Theta_0 - G_0$ (computed from (1))

 (example: with Θ = 0: $\varepsilon = G$)

3. correct each Θ_0 with error ε_0: $\Theta'_0 = \Theta_0 + \varepsilon_0/(\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2})$

$$= (\frac{1}{\Delta x^2}(\Theta_1+\Theta_3)+\frac{1}{\Delta y^2}(\Theta_2+\Theta_4)-G_0)/(\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2})$$

4. continue with 2. until the error is sufficiently small (given an adequate error norm)

Problem: very slow and, therefore, not feasible!

Solving a Poisson-Equation: Gauß-Seidel Method and SOR

Jacobi Method: $\Theta'_0 = (\frac{1}{\Delta x^2}(\Theta_1+\Theta_3)+\frac{1}{\Delta y^2}(\Theta_2+\Theta_4)-G_0)/(\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2})$

Improvement: Gauß-Seidel Method

$$\Theta'_0 = (\frac{1}{\Delta x^2}(\Theta_1+\Theta'_3)+\frac{1}{\Delta y^2}(\Theta_2+\Theta'_4)-G_0)/(\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2})$$

similar to Jacobi but using already known (computed) new values (Θ'_4, Θ'_3) advantage: feasible (fast enough)

6	2	5
3	0	1
7	4	8

grid points

More improvement: 'Successive Over-Relaxation' (SOR)

$$\Theta'_0 = \Theta_0 + \omega\varepsilon_0/(\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2})$$

$$= \Theta_0 + \omega(\frac{1}{\Delta x^2}(\Theta_1+\Theta'_3-2\Theta_0)+\frac{1}{\Delta y^2}(\Theta_2+\Theta'_4-2\Theta_0)-G_0)/(\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2})$$

similar to Gauß-Seidel but ‚over correct ' with $1 < \omega < 2$

advantage: fast   disadvantage: ω needs to be chosen by 'try and error'

Solving a Poisson-Equation: Grid Point Method

Performance (NxN Grid):

Jacobi: N^2 *p/2* iterations to decrease the initial error by factor 10^p

Gauß-Seidel: N^2 *p/4* iterations to decrease the initial error by factor 10^p (0.5 Jacobi).

SOR: *N p/3* iterations to decrease the initial error by factor 10^p (about 1/N Gauß-Seidel)

Further improvement: multigrid methods

Ansatz: Faster convergence of iteration for larger scales

=> **Multigrid method** (example):

1. interpolate $G(x,y)$ and $\Theta(x,y)$ to a coarse grid
2. solve (iterate) equation on coarse grid (e.g. by SOR)
3. interpolate solution from 2 to finer grid
4. solve (iterate) equation on finer grid (e.g. by SOR)
5. repeat 1 to 4 until final resolution (grid) and accuracy is reached

Elliptic Partial Differential Equations

Summary

• Elliptic equations: Laplace, Poisson, Helmholz

• Spectral method: Eigenfunctions

• Grid point methods: Jacobi, Gauß-Seidel, SOR

• Multigrid method

Introduction to Numerical Modeling

8. An Example: QG Barotropic Channel Model (Weather Prediction)

The Barotropic Model

- First (functioning) numerical weather prediction model

(Charney, J. G., Fjortoft, R. and von Neumann, J. 1950. Numerical integration of the barotropic vorticity equation. *Tellus, 2*, 237–54.)

- Simple model for idealized (conceptional) studies

The Barotropic Model: Equations

Homogeneous incompressible fluid, constant density, hydrostatic equilibrium, Cartesian coordinates, z-system, barotropic, ...

Equation of motion:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv = -\frac{1}{\rho}\frac{\partial P}{\partial x}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu = -\frac{1}{\rho}\frac{\partial P}{\partial y}$$

Hydrostatic equation:

$$\frac{\partial P}{\partial z} = -g\rho$$

Continuity equation:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

The Barotropic Model: Transformations/Approximations

1 ‚primitive' shallow water equations

A) Integrating the hydrostatic equation and replacing the pressure gradient in the equation of motion:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv = -\frac{\partial gh}{\partial x}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu = -\frac{\partial gh}{\partial y}$$

B) Integrating the continuity equation (0 to h) and using boundary conditions for w (here: no bottom topography):

$$\frac{\partial gh}{\partial t} + u\frac{\partial gh}{\partial x} + v\frac{\partial gh}{\partial y} = -gh\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)$$

The Barotropic Model: Transformations/Approximations

2 ‚quasi-geostrophic' shallow water equations

A) QG-approximation of the equation of motion (ß-plain):

$$\frac{\partial u_g}{\partial t} + u_g \frac{\partial u_g}{\partial x} + v_g \frac{\partial u_g}{\partial y} - f_0 v_a - \beta y v_g = -\frac{\partial g h_a}{\partial x}$$

$$\frac{\partial v_g}{\partial t} + u_g \frac{\partial v_g}{\partial x} + v_g \frac{\partial v_g}{\partial y} + f_0 u_a + \beta y u_g = -\frac{\partial g h_a}{\partial y}$$

Or: QG vorticity equation:

$$\frac{\partial \zeta_g}{\partial t} + u_g \frac{\partial \zeta_g}{\partial x} + v_g \frac{\partial \zeta_g}{\partial y} + \beta v_g = -f_0 \left(\frac{\partial u_a}{\partial x} + \frac{\partial v_a}{\partial y}\right)$$

B) QG-approximation of the continuity equation

$$\frac{\partial g h_g}{\partial t} + u_g \frac{\partial g h_g}{\partial x} + v_g \frac{\partial g h_g}{\partial y} = -g h_0 \left(\frac{\partial u_a}{\partial x} + \frac{\partial v_a}{\partial y}\right)$$

The Barotropic Model: Transformations/Approximations

2 ‚quasi-geostrophic' shallow water equations

with geostrophic streamfunction $\psi = g h_g / f_0$:

$$u_g = -\frac{\partial \Psi}{\partial y}$$

$$v_g = \frac{\partial \Psi}{\partial x}$$

$$\zeta_g = \nabla^2 \Psi = \frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial y^2}$$

$$\frac{\partial (\nabla^2 - \lambda^{-2})\Psi}{\partial t} = -\frac{\partial \Psi}{\partial x}\frac{\partial (\nabla^2 - \lambda^{-2})\Psi}{\partial y} + \frac{\partial \Psi}{\partial y}\frac{\partial (\nabla^2 - \lambda^{-2})\Psi}{\partial x} - \beta \frac{\partial \Psi}{\partial x}$$

$$= -J(\Psi, (\nabla^2 - \lambda^{-2})\Psi) - \beta \frac{\partial \Psi}{\partial x}$$

with λ = Rossby Radius of Deformation = $(g h_0)^{1/2} / f_0$

The Barotropic Model: Transformations/Approximations

3 non-divergent shallow water equations

(QG-)vorticity equation (non divergent):

$$\frac{\partial \zeta_g}{\partial t} + u_g \frac{\partial \zeta_g}{\partial x} + v_g \frac{\partial \zeta_g}{\partial y} + \beta v_g = 0$$

Or:

$$\frac{\partial \nabla^2 \Psi}{\partial t} = -\frac{\partial \Psi}{\partial x}\frac{\partial \nabla^2 \Psi}{\partial y} + \frac{\partial \Psi}{\partial y}\frac{\partial \nabla^2 \Psi}{\partial x} - \beta \frac{\partial \Psi}{\partial x}$$

$$= -J(\Psi, \nabla^2 \Psi) - \beta \frac{\partial \Psi}{\partial x}$$

The Barotropic Model: Transformations/Approximations

4 equivalent barotropic model

Assumption: absolute value of velocity may change with hight but not the direction: $(u,v) = A(p)(\langle u \rangle(x,y,t), \langle v \rangle(x,y,t))$

$$\langle B \rangle = \frac{1}{p_s}\int_0^{p_s} B dp$$

It follows:

$$\frac{\partial(\nabla^2 - A(p_s)\lambda^{-2})\Psi^*}{\partial t} = -J(\Psi^*, \nabla^2 \Psi^*) - \beta \frac{\partial \Psi^*}{\partial x}$$

with $\psi^* = \langle A^2 \rangle \psi$; $\langle A \rangle = 1$

Valid for the equivalent barotropic level p^* with: $A(p^*) = \langle A^2 \rangle$

(typically 600-500hPa; minimum divergence)

Summary
The Barotropic Model: Equations

'primitive' equations:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv = -\frac{\partial gh}{\partial x}$$
$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu = -\frac{\partial gh}{\partial y}$$
$$\frac{\partial gh}{\partial t} + u\frac{\partial gh}{\partial x} + v\frac{\partial gh}{\partial y} = -gh\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)$$

Quasi-Geostrophic:
$$\frac{\partial(\nabla^2 - \lambda^{-2})\Psi}{\partial t} = -J(\Psi, (\nabla^2 - \lambda^{-2})\Psi) - \beta\frac{\partial \Psi}{\partial x}$$

Non-divergent:
$$\frac{\partial \nabla^2 \Psi}{\partial t} = -J(\Psi, \nabla^2\Psi) - \beta\frac{\partial \Psi}{\partial x}$$

Equivalent barotropic:
$$\frac{\partial(\nabla^2 - A(p_s)\lambda^{-2})\Psi^*}{\partial t} = -J(\Psi^*, \nabla^2\Psi^*) - \beta\frac{\partial \Psi^*}{\partial x}$$

From Equations to Numerical Model:
Model Design

A) The equation(s):
Here: barotropic non-divergent

$$\frac{\partial \nabla^2 \Psi}{\partial t} = -J(\Psi, \nabla^2\Psi) - \beta\frac{\partial \Psi}{\partial x}$$

B) The numerical method - general
Here: grid point method

C) The numerical method – specific

variables, operators, grid, discretizations, work flow, boundary conditions, etc.

Barotropic non-divergent Model: Numerics

$$\frac{\partial \nabla^2 \Psi}{\partial t} = -J(\Psi, \nabla^2 \Psi) - \beta \frac{\partial \Psi}{\partial x} \Leftrightarrow \frac{\partial \Psi}{\partial t} = \nabla^{-2} \left[-J(\Psi, \nabla^2 \Psi) - \beta \frac{\partial \Psi}{\partial x} \right]$$

Prognostic variable: Streamfunction Ψ

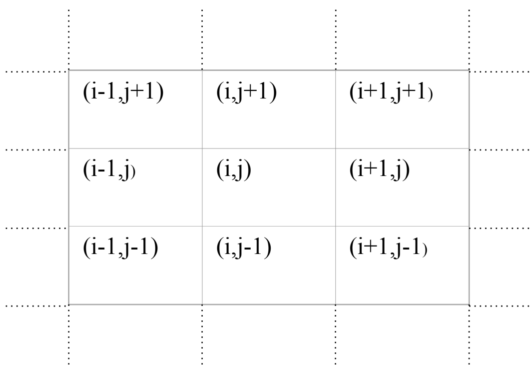
Operators:

- Derivation in time: $\frac{\partial \Psi}{\partial t}$
- Derivation in space: $\beta \frac{\partial \Psi}{\partial x}$
- Jacobi-Operator: $J(\Psi, \nabla^2 \Psi)$
- Laplace-Operator and its inverse: $\Psi \leftrightarrow \nabla^2 \Psi$

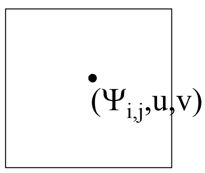
Barotropic non-divergent Model: Numerics

The Grid:
Lon-Lat grid; one prognostic variable only (Ψ) -> Arakawa A

The Grid:



Gridbox (i,j)



Barotropic non-divergent Model: Numerics Discretizations

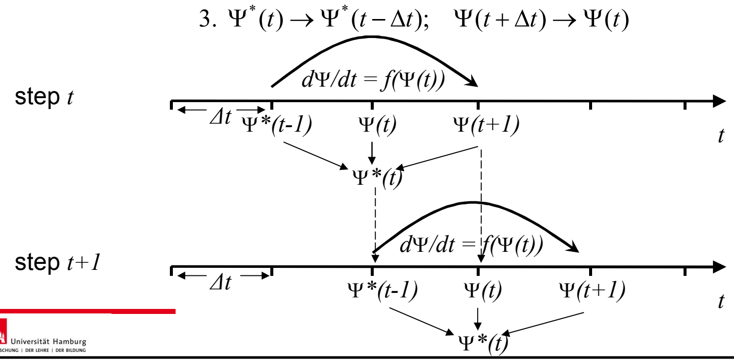
Time derivative: Leapfrog with Robert-Asselin filter

(Three level scheme; conservative (non-dissipative) system; Computation of Ψ*(t) (and* Ψ*(t+Δt))* by weighted averages of Ψ*(t+Δt)*, Ψ*(t)* and Ψ*(t-Δt)*)

Courant-Friedrich-Levy criterion: $(u\Delta t/\Delta x)^2 < (1-\gamma)^2$

Calculation role:

1. $\Psi(t+\Delta t) = \Psi^*(t-\Delta t) + 2\Delta t \cdot f(\Psi(t))$
2. $\Psi^*(t) = \Psi(t) + \gamma(\Psi^*(t-\Delta t) - 2\Psi(t) + \Psi(t+\Delta t))$ (γ = *filter const.*)
3. $\Psi^*(t) \to \Psi^*(t-\Delta t); \quad \Psi(t+\Delta t) \to \Psi(t)$ *(e.g. γ = 0.1)*



Barotropic non-divergent Model: Numerics Discretizations

Derivation in space: centered differences

$$\frac{\partial \Psi}{\partial x} = \frac{1}{2\Delta x}(\Psi_1 - \Psi_3)$$

$$\frac{\partial \Psi}{\partial y} = \frac{1}{2\Delta y}(\Psi_2 - \Psi_4)$$

Grid:

6 = (i-1,j+1)	2 = (i,j+1)	5 = (i+1,j+1)
3 = (i-1,j)	0 = (i,j)	1 = (i+1,j)
7 = (i-1,j-1)	4 = (i,j-1)	8 = (i+1,j-1)

Laplacian: centered differences

$$\nabla^2\Psi(i,j) = \nabla^2\Psi_0 = \frac{1}{\Delta x^2}(\Psi_1 + \Psi_3 - 2\Psi_0) + \frac{1}{\Delta y^2}(\Psi_2 + \Psi_4 - 2\Psi_0)$$

Barotropic non-divergent Model: Numerics Discretizations

Jacobi-Operator: $J(\Psi, \nabla^2\Psi)$

Grid:

6 = (i-1,j+1)	2 = (i,j+1)	5 = (i+1,j+1)
3 = (i-1,j)	0 = (i,j)	1 = (i+1,j)
7 = (i-1,j-1)	4 = (i,j-1)	8 = (i+1,j-1)

analytically
$$
\begin{aligned}
J(a,b) &= \frac{\partial a}{\partial x}\frac{\partial b}{\partial y} - \frac{\partial a}{\partial y}\frac{\partial b}{\partial x} \\
&= \frac{\partial}{\partial x}\left(a\frac{\partial b}{\partial y}\right) - \frac{\partial}{\partial y}\left(a\frac{\partial b}{\partial x}\right) \\
&= \frac{\partial}{\partial y}\left(b\frac{\partial a}{\partial x}\right) - \frac{\partial}{\partial x}\left(b\frac{\partial a}{\partial y}\right)
\end{aligned}
$$

numerically:
$$
\begin{aligned}
J_1 &= \frac{1}{4\Delta x\Delta y}\{(a_1-a_3)(b_2-b_4)-(a_2-a_4)(b_1-b_3)\} \\
J_2 &= \frac{1}{4\Delta x\Delta y}\{a_1(b_5-b_8)-a_3(b_6-b_7)-a_2(b_5-b_6)+a_4(b_8-b_7)\} \\
J_3 &= \frac{1}{4\Delta x\Delta y}\{b_2(a_5-a_6)-b_4(a_8-a_7)-b_1(a_5-a_8)+b_3(a_6-a_7)\}
\end{aligned}
$$

$J=(J_1+J_2+J_3)/3$ (Arakawa `66) enstrophy and energy conserving

Barotropic non-divergent Model: Numerics Discretizations

Inverse Laplacian (solution of a Poisson-equation): $\nabla^2\Theta = G(x,y)$

e.g. **'Successive Over-Relaxation' (SOR)**

Grid:

6= (i-1,j+1)	2 = (i,j+1)	5= i+1,j+1)
3 = (i-1,j)	0 = (i,j)	1 = (i+1,j)
7 = (i-1,j-1)	4 = (i,j-1)	8= (i+1,j-1)

Discrete Laplacian:

$$\nabla^2\Theta_0 - G_0 = \frac{1}{\Delta x^2}(\Theta_1+\Theta_3-2\Theta_0)+\frac{1}{\Delta y^2}(\Theta_2+\Theta_4-2\Theta_0)-G_0 = 0$$

Iterative solution:

1. estimate the error: $\varepsilon_0 = \nabla^2\Theta_0 - G_0$

2. (over-) correct the error (using already new values):

$$
\begin{aligned}
\Theta'_0 &= \Theta_0 + \omega\varepsilon_0 / \left(\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2}\right) \\
&= \Theta_0 + \omega\left(\frac{1}{\Delta x^2}(\Theta_1+\Theta'_3-2\Theta_0)+\frac{1}{\Delta y^2}(\Theta_2+\Theta'_4-2\Theta_0)-G_0\right) / \left(\frac{2}{\Delta x^2}+\frac{2}{\Delta y^2}\right)
\end{aligned}
$$

Barotropic non-divergent Model: Numerics

Work flow:

$$\frac{\partial \nabla^2 \Psi}{\partial t} = -J(\Psi, \nabla^2 \Psi) - \beta \frac{\partial \Psi}{\partial x}$$

1. **Initialization**
   a) define grid
   b) set boundary conditions
   c) set initial conditions
2. **Time loop**
   a) compute the right hand side (tendency)
   b) Inverse Laplacian = solving a Poisson-equation
   c) compute ψ at new time step
3. **Finalization**
   a) write restart files

Summary: Numerics

Equation: *barotropic non-divergent* $$\frac{\partial \nabla^2 \Psi}{\partial t} = -J(\Psi, \nabla^2 \Psi) - \beta \frac{\partial \Psi}{\partial x}$$

Method: Grid point

Grid: Arakawa A

Prognostic variable: streamfunction

Time stepping: Leapfrog with filter

Differentials in space: central differences

Jacobi operator: energy and enstrophy conserving (Arakawa)

Inverse Laplacian: 'Successive Over-Relaxation' (SOR)

From the Design to the Code: The FORTRAN program

Modular structure:

(sets of) subroutines for individual model parts:

- input
- output
- grid definition
- derivation in space
- Laplacian
- Jacobi operator
- Leapfrog
- Euler
- 'Successive Over-Relaxation' (SOR)
- boundary conditions
- initial conditions
- ...

organized and called by a main program

From the Design to the Code: The FORTRAN program

How to program?

- use ‚modules' for global parameters/variables
- use a name convention (real/integer; global/local, etc)
- document/comment your code (the more the better)
- try to be flexible (use parameters etc.)
- test as frequent as possible

From the Design to the Code: The FORTRAN program

How to test?

- use simple structures with know solutions (sin,cos) to check the derivatives
- use analytic solution to check the dynamics e.g. Rossby-Haurwitz wave

R-H wave:

only one wave number k and l in x- and y-direction: $\Psi(x,y,t) = \Psi_0 e^{i(kx+ly-\omega t)}$

$$\Rightarrow J(\Psi,\nabla^2\Psi) = 0$$

$$\Rightarrow \frac{\partial \nabla^2 \Psi}{\partial t} = -\beta \frac{\partial \Psi}{\partial x}$$

$$\Rightarrow c = -\frac{\beta}{K^2}$$

westward propagating, amplitude and wavenumber conserving

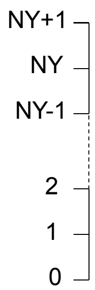
The Barotropic Model: Boundary Conditions

General: at each boundary (East, West, North, South) two boundary conditions are needed: u and v or streamfunction Ψ and vorticity ξ

Our model: non divergent barotropic vorticity equation, grid point method, A-grid, Ψ

The grid: Example: one longitude:

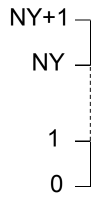
Grid Point j=



Here: 0,NY+1 = Channel boundaries, 1,2,...NY = ‚aktive'

i.e. boundary conditions at GP 0 and NY+1 needed

Boundary Conditions: Examples



a) Prescribed from data

b) Cyclic: $A_0 = A_{NY}$ and $A_{NY+1} = A_1$ (with A= u, v or Ψ, ξ)

c) No flow across boundary: $v_0 = v_{NY+1} = 0$, i.e. $\Psi_0 = \Psi_{NY+1}$ = const in x, as $v_{0,NY+1} = \left(\frac{\partial \Psi}{\partial x}\right)_{0,NY+1}$

and

c1) $u_0 = u_1$ and $u_{NY+1} = u_{NY}$ (**full slip boundary condition**), i.e. $\xi_{0,NY+1} = 0$,

as $\xi_{0,NY+1} = -\left(\frac{\partial u}{\partial y}\right)_{0,NY+1} \overset{!}{=} 0$

c2) $u_{0,NY+1} = 0$ (**no slip boundary condition**), i.e. (in our model) $\xi_{0,NY+1} = 2\frac{\Psi_{1,NY+1} - \Psi_{0,NY}}{(\Delta y)^2}$

as $\xi_j = -\frac{u_{j+1/2} - u_{j-1/2}}{\Delta y}$ and $u_{-1/2,NY+3/2} \overset{!}{=} -u_{1/2,NY+1/2}$ and $u_{1/2,NY+1/2} = -\frac{\Psi_{1,NY+1} - \Psi_{0,NY}}{\Delta y}$

Boundary Conditions: Summary

- Prescribed
- Cyclic
- Perpendicular: No flow across boundary
- Tangential: No slip and full slip