

1-D Simulation of Western Boundary Current

Chuning Wang

February 21, 2017

1 Model setup

1.1 Vorticity equation

The numerical model in this report aims to solve the 1-D vorticity equation

$$\frac{\partial}{\partial t}\phi_{xx} + \beta\phi_x = WC - \gamma\phi_{xx} \quad (1)$$

where ϕ is the stream function, the terms on the left side are the 'storage term' and ' β effect term', while terms on the right side are wind curl and friction, respectively. To solve this equation numerically, the domain is set to a 1-D grid consists of $X + 1$ points, and the spatial derivation terms must be discretized with finite difference method:

$$(\phi_x)_i^n = \frac{\phi_{i+1}^n - \phi_{i-1}^n}{2\Delta x} \quad (2)$$

$$(\phi_{xx})_i^n = \frac{\phi_{i+1}^n - 2\phi_i^n + \phi_{i-1}^n}{\Delta x^2} \quad (3)$$

and temporal derivation can be discretized with either forward scheme

$$\frac{\partial}{\partial t}(\phi_{xx})_i = \frac{(\phi_{xx})_i^{n+1} - (\phi_{xx})_i^n}{\Delta t} \quad (4)$$

or leap frog scheme

$$\frac{\partial}{\partial t}(\phi_{xx})_i = \frac{(\phi_{xx})_i^{n+1} - (\phi_{xx})_i^{n-1}}{2\Delta t} \quad (5)$$

The superscript n and subscript i are indices for time steps and spatial grid points, respectively. Substitute Eq 4 into Eq 1 and rearrange terms

$$(\phi_{xx})_i^{n+1} = (\phi_{xx})_i^n + \Delta t(-\beta(\phi_x)_i^n + WC - \gamma(\phi_{xx})_i^n) \quad (6)$$

then substitute Eq 3 into Eq 6 for time step $n + 1$, the forward scheme, discretized 1-D vorticity equation becomes

$$\phi_{i+1}^{n+1} - 2\phi_i^{n+1} + \phi_{i-1}^{n+1} = \Delta x^2(\phi_{xx})_i^n + \Delta x^2 \Delta t(-\beta(\phi_x)_i^n + WC - \gamma(\phi_{xx})_i^n) \quad (7)$$

or using leap frog scheme

$$\phi_{i+1}^{n+1} - 2\phi_i^{n+1} + \phi_{i-1}^{n+1} = \Delta x^2(\phi_{xx})_i^{n-1} + 2\Delta x^2 \Delta t(-\beta(\phi_x)_i^n + WC - \gamma(\phi_{xx})_i^n) \quad (8)$$

For a 1-D grid consists of $X + 1$ points, Eq 7 and 8 satisfy everywhere except at boundaries $i = 0$ or $i = X$.

1.2 Boundary condition

To close the system and complete the linear equations, boundary condition are required. For this particular exercise, we use two sets of boundary conditions: first of which is ‘no net transport’

$$\phi = 0 \quad @ \quad x = 0, 1 \quad (9)$$

and second is for velocity v at each boundary, which is chosen from either non-slip

$$\phi_x = 0 \quad @ \quad x = 0, 1 \quad (10)$$

or free-slip

$$\phi_{xx} = 0 \quad @ \quad x = 0, 1 \quad (11)$$

Discretizing Eq 9, 10 and 11 for time step $n + 1$ gives

$$\phi_0^{n+1} = 0, \quad \phi_X^{n+1} = 0 \quad (12)$$

$$\phi_{-1}^{n+1} - \phi_1^{n+1} = 0, \phi_{X-1}^{n+1} - \phi_{X+1}^{n+1} = 0 \quad (13)$$

$$\phi_{-1}^{n+1} - 2\phi_0^{n+1} + \phi_1^{n+1} = 0, \phi_{X-1}^{n+1} - 2\phi_X^{n+1} + \phi_{X+1}^{n+1} = 0 \quad (14)$$

where $i = -1$ and $i = X + 1$ are two imaginary grid points brought into the system to satisfy the second set of boundary condition.

1.3 Matrix construction

At this point, we have constructed a linear equation system with $X + 3$ equations and $X + 3$ unknowns. It can be expressed with a matrix form

$$A\phi^{n+1} = B \quad (15)$$

where A is a sparse matrix dependent on the choice of the second boundary condition and B is a 1-D array dependent on the choice of temporal discretization scheme. An example of A with non-slip condition is

$$A = \begin{bmatrix} 1 & 0 & -1 & & & & & \\ 0 & 1 & 0 & & & & & \\ & 1 & -2 & 1 & & & & \\ & & 1 & -2 & 1 & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & 1 & -2 & 1 & \\ & & & & & 1 & -2 & 1 \\ & & & & & & 0 & 1 & 0 \\ & & & & & & 1 & 0 & -1 \end{bmatrix} \quad (16)$$

and B with forward scheme is

$$B = \begin{bmatrix} 0 \\ 0 \\ \Delta x^2(\phi_{xx})_1^n + \Delta x^2 \Delta t(-\beta(\phi_x)_1^n + WC - \gamma(\phi_{xx})_1^n) \\ \Delta x^2(\phi_{xx})_2^n + \Delta x^2 \Delta t(-\beta(\phi_x)_2^n + WC - \gamma(\phi_{xx})_2^n) \\ \vdots \\ \Delta x^2(\phi_{xx})_{X-1}^n + \Delta x^2 \Delta t(-\beta(\phi_x)_{X-1}^n + WC - \gamma(\phi_{xx})_{X-1}^n) \\ \Delta x^2(\phi_{xx})_{X-2}^n + \Delta x^2 \Delta t(-\beta(\phi_x)_{X-2}^n + WC - \gamma(\phi_{xx})_{X-2}^n) \\ 0 \\ 0 \end{bmatrix} \quad (17)$$

The unknown ϕ^{n+1} is also a 1-D array

$$\phi^{n+1} = \begin{bmatrix} \phi_{-1}^{n+1} \\ \phi_0^{n+1} \\ \phi_1^{n+1} \\ \phi_2^{n+1} \\ \vdots \\ \phi_{X-2}^{n+1} \\ \phi_{X-1}^{n+1} \\ \phi_X^{n+1} \\ \phi_{X+1}^{n+1} \end{bmatrix} \quad (18)$$

Iteratively solve Eq 16 for $n = n+1$ to move forward in temporal domain. Update velocity $v = \phi_x$ and vorticity $\zeta = \phi_{xx}$ for each step and store the solution.

1.4 Numerical tips

- When N is a very large number, matrix A becomes larger accordingly and solving the system eats up a lot of memory. However, since A is a sparse matrix, Python and MATLAB both offer special algorithms to solve the equation system much more efficiently. Make sure to use these algorithms and save some time.
- Forward scheme and leap frog scheme both works in this case. Here we solve the equation with both schemes, and average them to get a

'mixed' solution. Note that to use leap frog scheme two previous time steps are required - thus the first step (from $n = 0$ to $n = 1$) can only be solved with forward scheme.

2 Results

A Source code