Shallow water model documentation

Milan Kloewer **version of February 3, 2017**

This documentation aims to summarize a methodology how the shallow water equations can be discretized and solved numerically with finite differences. The corresponding numerical model can be found at www.github.com/milankl/swm.

1 Physical model

1.1 General model

The shallow water equations of interest are (see Gill?)

$$\partial_t u + u\partial_x u + v\partial_y u - fv = -g\partial_x \eta + F_x + M_x \tag{1}$$
$$\partial_t v + u\partial_x v + v\partial_y v + fu = -g\partial_y \eta + F_y + M_y \tag{2}$$
$$\partial_t \eta + \partial_x(uh) + \partial_y(vh) = 0 \tag{3}$$

with

$\mathbf{u} = (u, v) = (u(x, y, t), v(x, y, t))$	horizontal velocity vector
$\eta = \eta(x, y, t)$	surface displacement
$h = h(x, y, t) = \eta + H$	layer thickness
$H = H(x, y)$	undisturbed layer thickness
$f = f(y)$	coriolis parameter
$g = \text{const}$	gravitational acceleration
$\mathbf{F} = (F_x, F_y) = (F_x(x, y, t), F_y(x, y, t))$	forcing vector
$\mathbf{M} = (M_x, M_y) = (M_x(u, v, h), M_y(u, v, h))$	lateral mixing of momentum, diffusion, or friction term[1]

and differential operators

$$\partial_x = \frac{\partial}{\partial x}, \partial_y = \frac{\partial}{\partial y}, \partial_t = \frac{\partial}{\partial t}$$

on the domain $\mathcal{D} = (0, L_x) \times (0, L_y)$ of width (or east-west extent) L_x and north-south extent L_y and with cartesian coordinates x, y and time t. The initial conditions are

$$u(t = 0) = u_0(x, y), \quad v(t = 0) = v_0(x, y), \quad h(t = 0) = h_0(x, y) \tag{4}$$

The kinematic boundary condition states that there is no flow through the boundary, i.e.

$$u(x = 0) = u(x = L_x) = 0 \tag{5}$$
$$v(y = 0) = v(y = L_y) = 0, \tag{6}$$

additionally we require no gradient of η across the boundary, hence

$$\partial_x \eta(x = 0) = \partial_x \eta(x = L_x) = 0 \tag{7}$$
$$\partial_y \eta(y = 0) = \partial_y \eta(y = L_y) = 0. \tag{8}$$

NO-SLIP FREE-SLIP

The prognostic variables so far are u, v, η, however, as η only appears in gradients, also h can be used as prognostic variable. We can separate from the advective terms in equation 1,2 the spatial gradient of kinetic energy. In combination with the pressure gradient term we introduce the Bernoulli potential P as

$$P = \frac{1}{2}(u^2 + v^2) + gh \tag{9}$$

Furthermore, with introducing the relative vorticity $\zeta = \partial_x v - \partial_y u$ the potential vorticity can be defined as

$$q = \frac{f + \zeta}{h} \tag{10}$$

and the equations 1,2 and 3 then become

$$\partial_t u = qhv - \partial_x P + F_x + M_x \tag{11}$$
$$\partial_t v = -qhu - \partial_y P + F_y + M_y \tag{12}$$
$$\partial_t h = -\partial_x(uh) - \partial_y(vh) \tag{13}$$

which are the equations solved by the numerical model as described in the next section.

1.2 Double gyre set up

In order to simulate mid-latitudinal dynamics we choose the physical parameters of the previous section as

$$g = 10 \text{ m/s} \tag{14}$$
$$H = 500 \text{ m} \tag{15}$$
$$L_x = L_y = 3840 \text{ km} \tag{16}$$
$$F_y = 0 \tag{17}$$
$$F_x = \frac{F_0}{\rho_0 H}\left[\cos\left(2\pi(\frac{y}{L_y} - \frac{1}{2})\right) + 2\sin\left(2\pi(\frac{y}{L_y} - \frac{1}{2})\right)\right] \tag{18}$$
$$F_0 = 0.12 \text{ Pa} \tag{19}$$
$$\rho_0 = 1000 \text{ kgm}^{-3} \tag{20}$$

and start from rest, so that the initial conditions become

$$u_0 = v_0 = 0, \quad h_0 = H \tag{21}$$

2 Spatial discretization

2.1 Grid

The domain $\mathcal{D}$ is divided into $n_x \times n_y$ grid cells, evenly spaced, so that each grid cell has the side length $\Delta x = \frac{L_x}{n_x}$ in x-direction and similarly Δy in y-direction. The spatial discretization then follows the ideas of the Arakawa C-grid, a staggered grid, where h_i

sits in the middle of the i-th grid cell at position (x_i, y_i), i.e. $h_i = h(x_i, y_i)$. In contrast, u, v and q are shifted

$$u_i = u(x_i + \frac{\Delta x}{2}, y_i) \tag{22}$$

$$v_i = u(x_i, y_i + \frac{\Delta y}{2}) \tag{23}$$

$$q_i = q(x_i + \frac{\Delta x}{2}, y_i + \frac{\Delta y}{2}). \tag{24}$$

Hence, we distinguish between 4 different grids: (i) the T-grid, for h or tracers, (ii) the u-grid, (iii) the v-grid, (iv) the q-grid. Not for all grid cells it is necessary to evaluate u or v, as they might vanish due to the kinematic boundary condition. The grids therefore carry a different amount of grid points. Let N_T, N_u, N_v, N_q be the total number of grid points on the respective grids then

$$N_T = n_x n_y, \qquad N_u = (n_x - 1)n_y \tag{25}$$

$$N_q = (n_x + 1)(n_y + 1), \qquad N_v = n_x(n_y - 1) \tag{26}$$

The n_x-th column of u-points vanish, as does the n_y-th row of v-points. However, there is no boundary condition for q, which makes it necessary to evaluate the q-grid for all points within the domain $\mathcal{D}$.

Choosing one index for the grid points (instead of two) leads to the advantage that every scalar variable can be represented as a vector. Numbering the grids row-first as can be seen in Fig. 1, leads to the following vector-representation $\mathtt{u}, \mathtt{v}, \mathtt{h}, \mathtt{q}$ of u, v, h and q

$$\mathtt{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N_u} \end{pmatrix}, \quad \mathtt{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N_v} \end{pmatrix}, \quad \mathtt{h} = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_{N_T} \end{pmatrix}, \quad \mathtt{q} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_{N_q} \end{pmatrix}. \tag{27}$$

In the special case of $n_x = n_y$ the vectors $\mathtt{u}, \mathtt{v}$ are of same size, but in general $\mathtt{u}, \mathtt{v}, \mathtt{h}, \mathtt{q}$ all differ in their sizes.

2.2 Gradients

Representing the model variables in vector-form, as discussed above, enables us to think of a gradient ∂ as a linear map between two vector spaces V_1, V_2. Hence, any gradient ∂ in that sense can be written as a matrix $\mathbf{G}$ which is multiplied with a vector $\mathtt{z}$ representing one of the model's variables:

$$\partial : V_1 \to V_2, \mathtt{z} \to \mathbf{G}\mathtt{z} \tag{28}$$

Having 4 different grids, we have to deal with 4 different vector spaces V_u, V_v, V_T, V_q. Let K be the algebraic field of the floating-point numbers with a given precision (e.g. 64bit), then $V_u = K^{N_u}, V_v = K^{N_v}, V_T = K^{N_T}$ and $V_q = K^{N_q}$. For simplicity we might consider $K = \mathbb{R}$, i.e. the real numbers.

In the following we will use a notation where the subscript denotes the direction of the derivative, hence x or y, and the superscript u, v, T, q denotes the vector space, where ∂ is mapping from. For the x-derivative on the T-grid

$$\partial_x^T : V_T \to V_2, \mathtt{z} \to \mathtt{G}_x^T \mathtt{z} \tag{29}$$

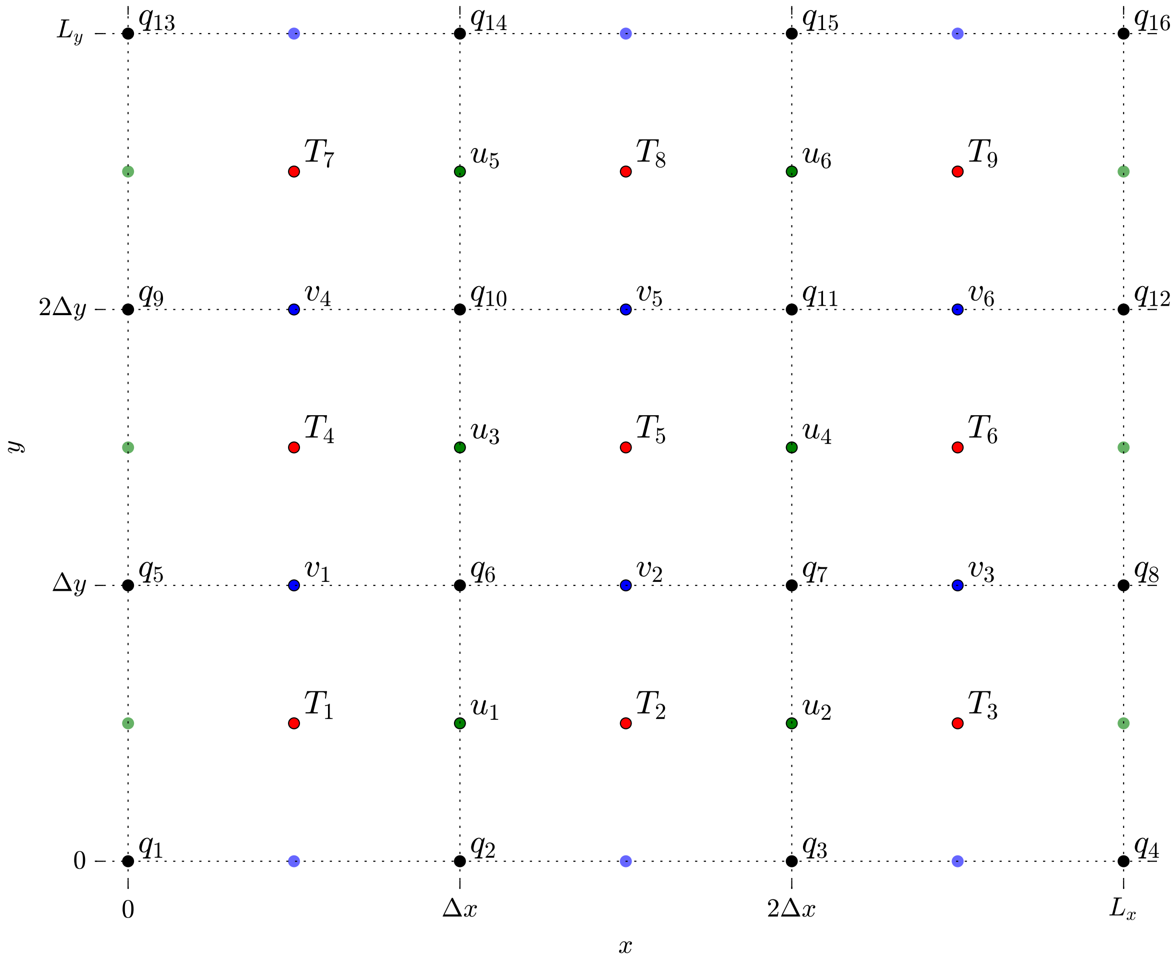

Figure 1: Grid cell numbering following the ideas of an Arakawa C-grid. The values of u, v vanish at the faint grid nodes on the boundary (kinematic boundary condition, see equation 5 and 6) and therefore these grid nodes are not explicitly resolved.

(using centred finite differences we will later identify that here $V_2 = V_u$.)

All gradients are now approximated by centred finite differences and

$$\mathtt{p} = \frac{1}{2}\left(\mathtt{I}_u^T(\mathtt{u}^2) + \mathtt{I}_v^T(\mathtt{v}^2)\right) + g\mathtt{h} \tag{30}$$

$$\mathtt{q} = \frac{\mathtt{f}_q + \mathtt{G}_x^v\mathtt{v} - \mathtt{G}_y^u\mathtt{u}}{\mathtt{I}_T^q\mathtt{h}} \tag{31}$$

$$\partial_t u = \mathtt{I}_q^u\mathtt{q} * \mathtt{I}_v^u(v * \mathtt{I}_T^v\mathtt{h}) - \mathtt{G}_x^T\mathtt{p} + \mathtt{F}_x + \nu_B\mathtt{L}_u^2\mathtt{u} \tag{32}$$

$$\partial_t v = -\mathtt{I}_q^v\mathtt{q} * \mathtt{I}_u^v(u * \mathtt{I}_T^u\mathtt{h}) - \mathtt{G}_y^T\mathtt{p} + \nu_B\mathtt{L}_v^2\mathtt{v} \tag{33}$$

$$\partial_t h = -\mathtt{G}_x^u(\mathtt{u} * \mathtt{I}_T^u\mathtt{h}) - \mathtt{G}_y^v(\mathtt{v} * \mathtt{I}_T^v\mathtt{h}) \tag{34}$$

3 Time discretization