2DH LSWE

2020/4/27

NTU ESOE 5136

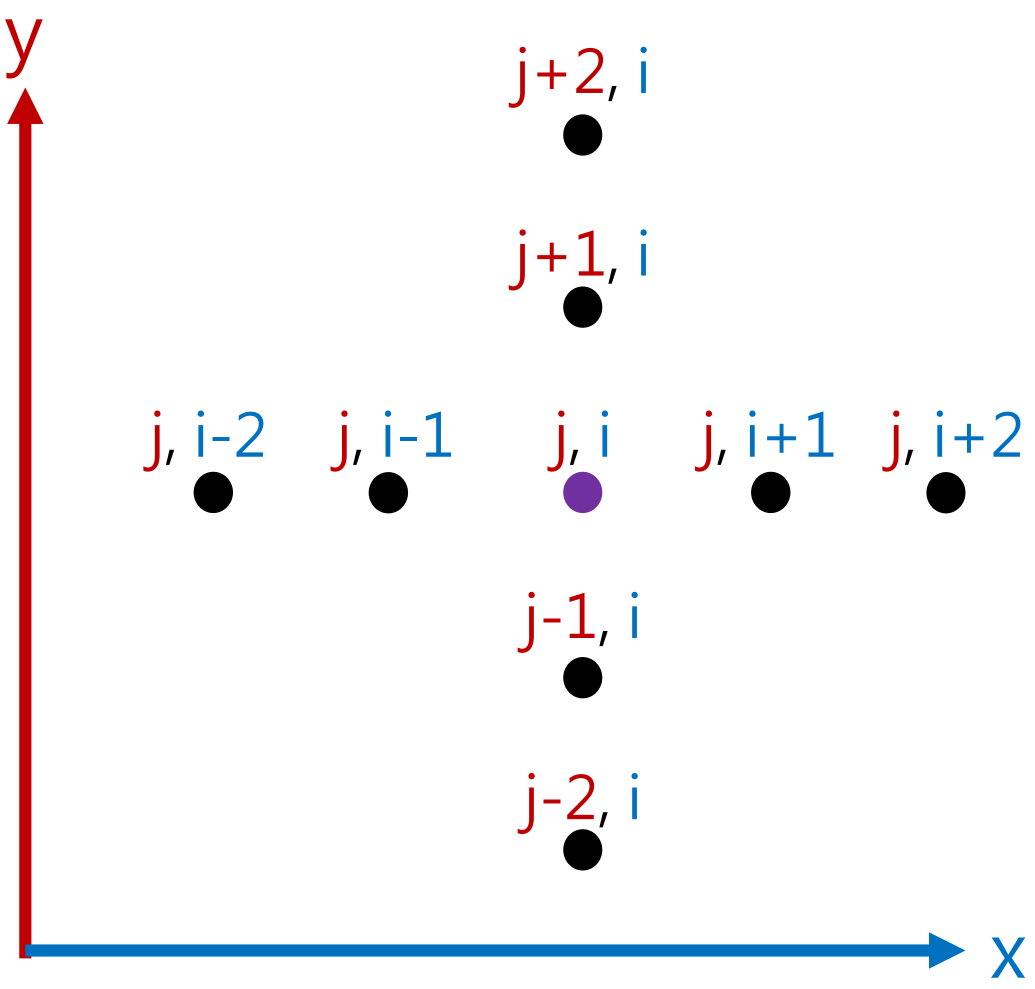
Content

1. governing equations
2. discretization
3. numerical scheme
4. boundary conditions
5. sponge layer
6. storage management in 2DH

Governing equations

$$
\begin{cases}
\dfrac{\partial \eta}{\partial t} + \dfrac{\partial U h}{\partial x} + \dfrac{\partial V h}{\partial y} = 0 \\[2ex]
\dfrac{\partial U}{\partial t} = -g \dfrac{\partial \eta}{\partial x} \\[2ex]
\dfrac{\partial V}{\partial t} = -g \dfrac{\partial \eta}{\partial y}
\end{cases}
$$

Discretization



$$\eta(x, y, t) \Rightarrow \eta_{j,i}^{(n)}$$

$$U(x, y, t) \Rightarrow U_{j,i}^{(n)}$$

$$V(x, y, t) \Rightarrow V_{j,i}^{(n)}$$

$$h(x, y, t) \Rightarrow h_{j,i}$$

4^{th}-order central difference in space

1. $$\left(\frac{\partial \eta}{\partial x}\right)_{j,i}^{(n)} \simeq \frac{-\eta_{j,i+2}^{(n)}+8\eta_{j,i+1}^{(n)}-8\eta_{j,i-1}^{(n)}+\eta_{j,i-2}^{(n)}}{12\Delta x}$$

2. $$\left(\frac{\partial \eta}{\partial y}\right)_{j,i}^{(n)} \simeq \frac{-\eta_{j+2,i}^{(n)}+8\eta_{j+1,i}^{(n)}-8\eta_{j-1,i}^{(n)}+\eta_{j-2,i}^{(n)}}{12\Delta y}$$

3. $$\left(\frac{\partial hU}{\partial x}\right)_{j,i}^{(n)} \simeq \frac{-h_{j,i+2}U_{j,i+2}^{(n)}+8h_{j,i+1}U_{j,i+1}^{(n)}-8h_{j,i-1}U_{j,i-1}^{(n)}+h_{j,i-2}U_{j,i-2}^{(n)}}{12\Delta x}$$

4. $$\left(\frac{\partial hV}{\partial y}\right)_{j,i}^{(n)} \simeq \frac{-h_{j+2,i}V_{j+2,i}^{(n)}+8h_{j+1,i}V_{j+1,i}^{(n)}-8h_{j-1,i}V_{j-1,i}^{(n)}+h_{j-2,i}V_{j-2,i}^{(n)}}{12\Delta y}$$

3rd-order SSP-RK in time

$$
\begin{cases}
\phi^{(*)} = \phi^{(n)} + \Delta t \mathcal{L}(\phi^{(n)}, t^{(n)}) \\
\\
\phi^{(**)} = \dfrac{3}{4}\phi^{(n)} + \dfrac{1}{4}\phi^{(*)} + \dfrac{1}{4}\Delta t \mathcal{L}(\phi^{(*)}, t^{(n)} + \Delta t) \\
\\
\phi^{(n+1)} = \dfrac{1}{3}\phi^{(n)} + \dfrac{2}{3}\phi^{(**)} + \dfrac{2}{3}\Delta t \mathcal{L}(\phi^{(**)}, t^{(n)} + \dfrac{1}{2}\Delta t)
\end{cases}
$$

Numerical scheme – first round

First round:

$$\eta_{j,i}^{(*)} = \eta_{j,i}^{(n)} - \frac{\Delta t}{12\Delta x}\Big(- U_{j,i+2}^{(n)} h_{j,i+2} + 8U_{j,i+1}^{(n)} h_{j,i+1} - 8U_{j,i-1}^{(n)} h_{j,i-1} + U_{j,i-2}^{(n)} h_{j,i-2}\Big)$$

$$- \frac{\Delta t}{12\Delta y}\Big(- V_{j+2,i}^{(n)} h_{j+2,i} + 8U_{j+1,i}^{(n)} h_{j+1,i} - 8U_{j-1,i}^{(n)} h_{j-1,i} + U_{j-2,i}^{(n)} h_{j-2,i}\Big)$$

$$U_{j,i}^{(*)} = U_{j,i}^{(n)} - \frac{\Delta t}{12\Delta x} g\Big(- \eta_{j,i+2}^{(n)} + 8\eta_{j,i+1}^{(n)} - 8\eta_{j,i-1}^{(n)} + \eta_{j,i-2}^{(n)}\Big)$$

$$V_{j,i}^{(*)} = V_{j,i}^{(n)} - \frac{\Delta t}{12\Delta y} g\Big(- \eta_{j+2,i}^{(n)} + 8\eta_{j+1,i}^{(n)} - 8\eta_{j-1,i}^{(n)} + \eta_{j-2,i}^{(n)}\Big)$$

Numerical scheme – second round

Second round:

$$\eta_{j,i}^{(**)} = \frac{3}{4}\eta_{j,i}^{(n)} + \frac{1}{4}\eta_{j,i}^{(*)} - \frac{1}{4}\frac{\Delta t}{12\Delta x}\Big(- U_{j,i+2}^{(*)}h_{j,i+2} + 8U_{j,i+1}^{(*)}h_{j,i+1} - 8U_{j,i-1}^{(*)}h_{j,i-1} + U_{j,i-2}^{(*)}h_{j,i-2}\Big)$$

$$-\frac{1}{4}\frac{\Delta t}{12\Delta y}\Big(- V_{j+2,i}^{(*)}h_{j+2,i} + 8V_{j+1,i}^{(*)}h_{j+1,i} - 8V_{j-1,i}^{(*)}h_{j-1,i} + V_{j-2,i}^{(*)}h_{j-2,i}\Big)$$

$$U_{j,i}^{(**)} = \frac{3}{4}U_{j,i}^{(n)} + \frac{1}{4}U_{j,i}^{(*)} - \frac{1}{4}\frac{\Delta t}{12\Delta x}g\Big(- \eta_{j,i+2}^{(*)} + 8\eta_{j,i+1}^{(*)} - 8\eta_{j,i-1}^{(*)} + \eta_{j,i-2}^{(*)}\Big)$$

$$V_{j,i}^{(**)} = \frac{3}{4}V_{j,i}^{(n)} + \frac{1}{4}V_{j,i}^{(*)} - \frac{1}{4}\frac{\Delta t}{12\Delta y}g\Big(- \eta_{j+2,i}^{(*)} + 8\eta_{j+1,i}^{(*)} - 8\eta_{j-1,i}^{(*)} + \eta_{j-2,i}^{(*)}\Big)$$

Numerical scheme – third round

Third round:

$$\eta_{j,i}^{(n+1)} = \frac{1}{3}\eta_{j,i}^{(n)} + \frac{2}{3}\eta_{j,i}^{(**)} - \frac{2}{3}\frac{\Delta t}{12\Delta x}\Big(- U_{j,i+2}^{(**)}h_{j,i+2} + 8U_{j,i+1}^{(**)}h_{j,i+1} - 8U_{j,i-1}^{(**)}h_{j,i-1} + U_{j,i-2}^{(**)}h_{j,i-2}\Big)$$

$$- \frac{2}{3}\frac{\Delta t}{12\Delta y}\Big(- V_{j+2,i}^{(**)}h_{j+2,i} + 8V_{j+1,i}^{(**)}h_{j+1,i} - 8V_{j-1,i}^{(**)}h_{j-1,i} + V_{j-2,i}^{(**)}h_{j-2,i}\Big)$$

$$U_{j,i}^{(n+1)} = \frac{1}{3}U_{j,i}^{(n)} + \frac{2}{3}U_{j,i}^{(**)} - \frac{2}{3}\frac{\Delta t}{12\Delta x}g\Big(- \eta_{j,i+2}^{(**)} + 8\eta_{j,i+1}^{(**)} - 8\eta_{j,i-1}^{(**)} + \eta_{j,i-2}^{(**)}\Big)$$

$$V_{j,i}^{(n+1)} = \frac{1}{3}V_{j,i}^{(n)} + \frac{2}{3}V_{j,i}^{(**)} - \frac{2}{3}\frac{\Delta t}{12\Delta y}g\Big(- \eta_{j+2,i}^{(**)} + 8\eta_{j+1,i}^{(**)} - 8\eta_{j-1,i}^{(**)} + \eta_{j-2,i}^{(**)}\Big)$$

CFL condition

1. $\Delta t \leq C_{CFL} \frac{\min(\Delta x, \Delta y)}{\sqrt{g h_{max}}}$

2. C_{CFL} in 2DH is usually smaller

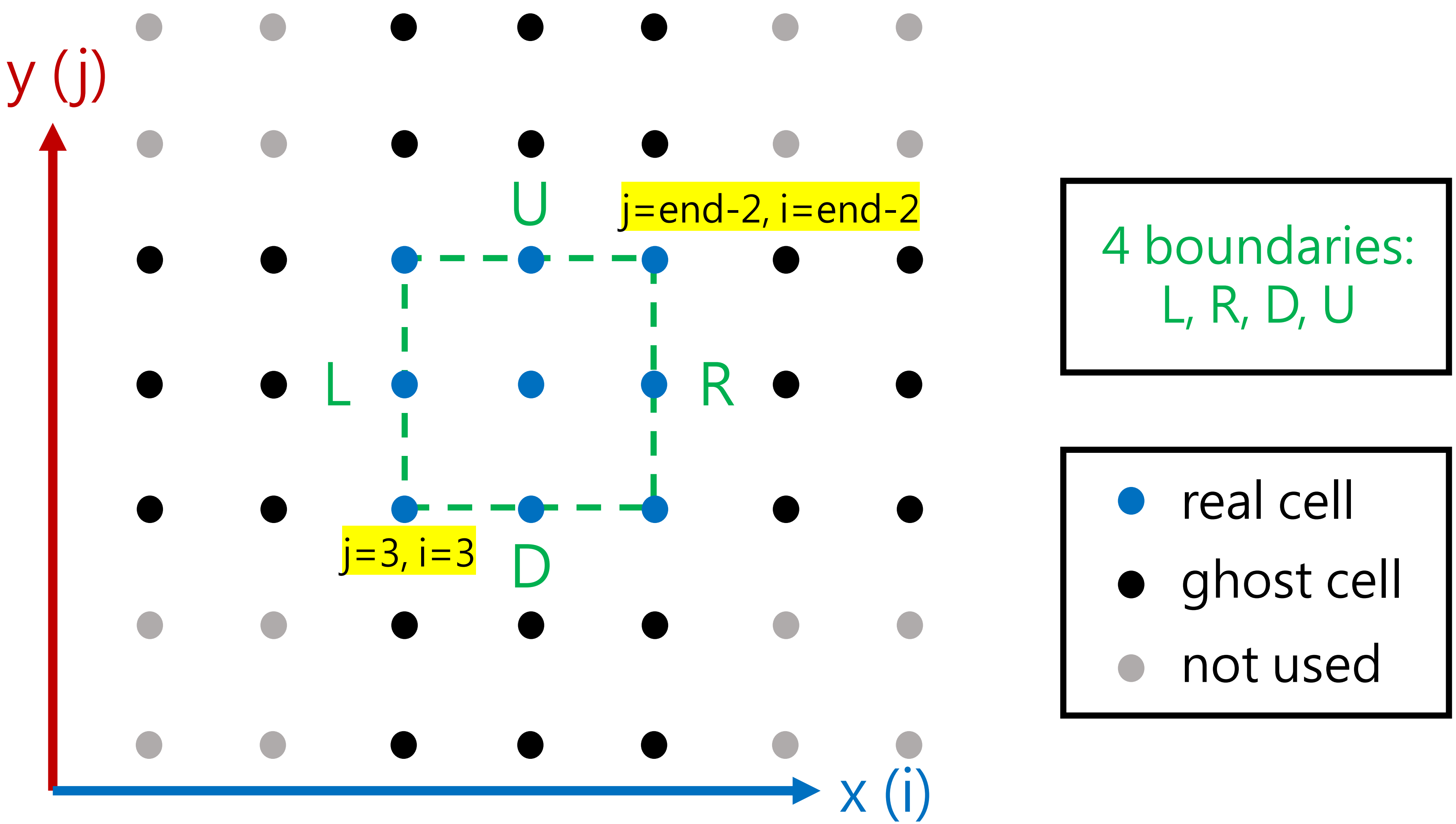
3. in my tests, $C_{CFL} = 0.9$ is still stable

Boundary conditions

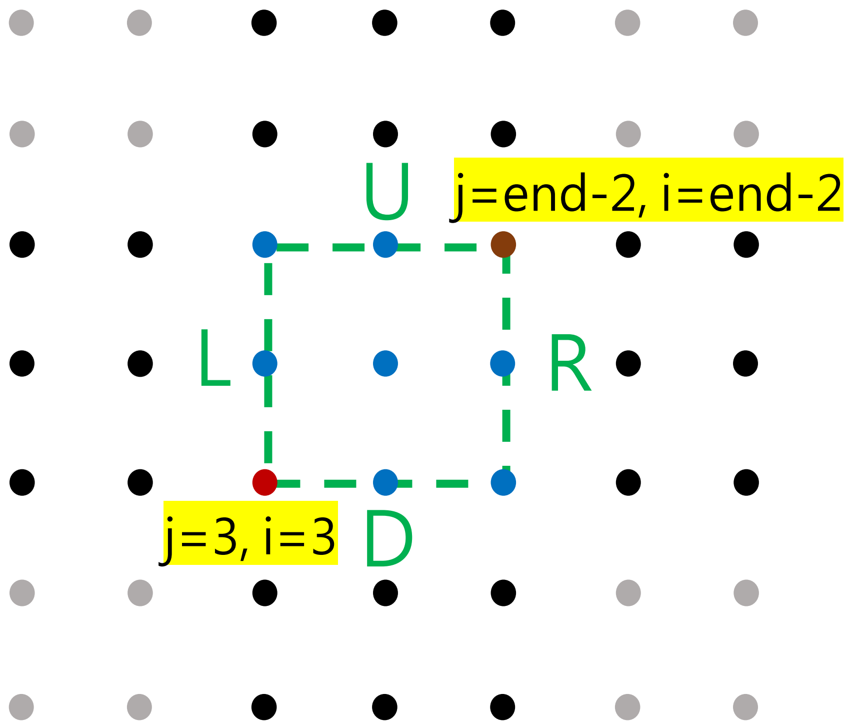
- at least 4 boundaries in 2DH
- BC choices:

1. periodic – robust and simple
2. mirror – robust and simple
3. open – tricky → sponge layer is easier?
4. wall – tricky → mirror is easier?

Ghost cells in 2DH



Periodic BC



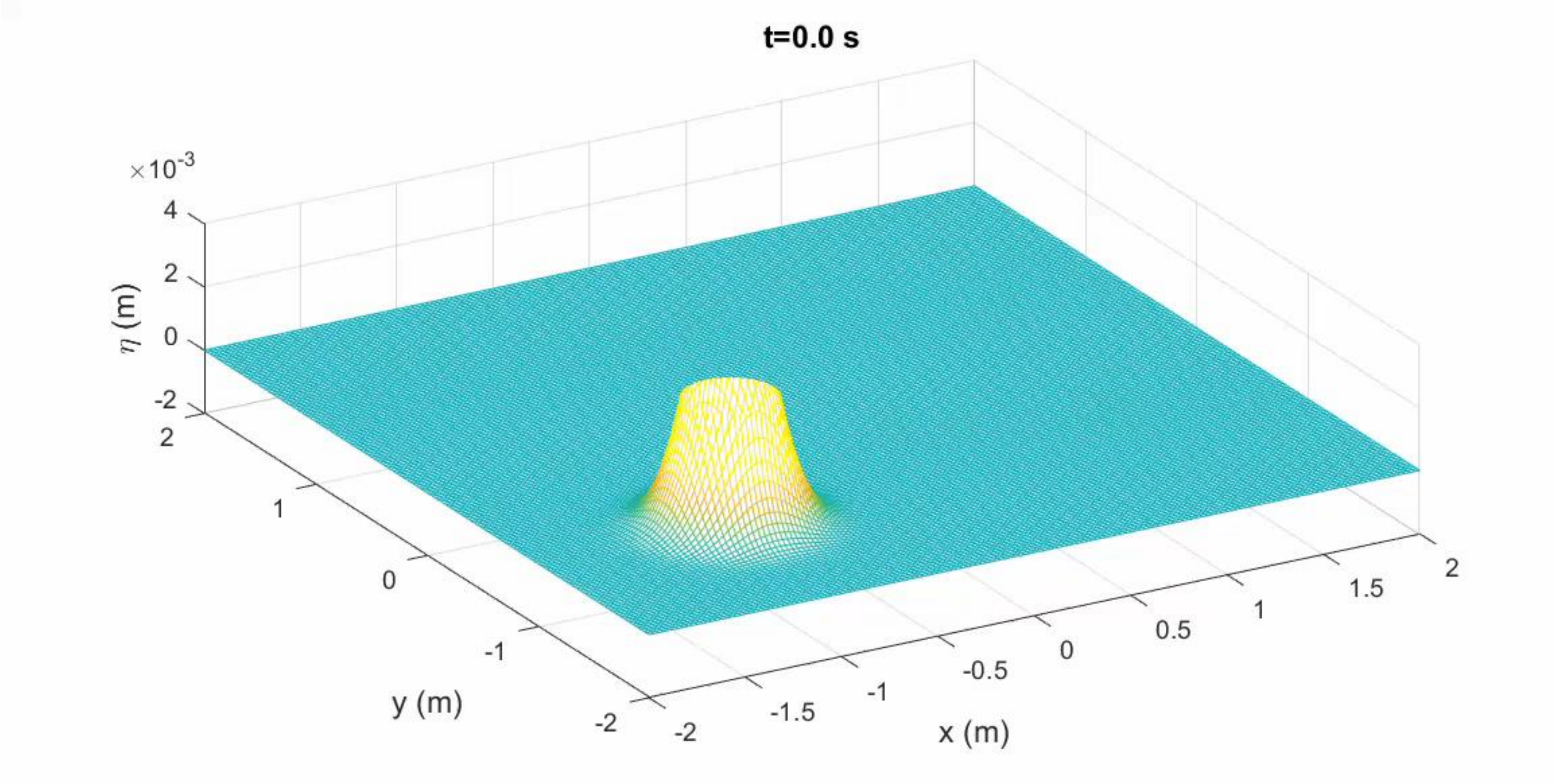
"L" boundary

$$\phi_{j=3,i=1} = \phi_{j=3,i=end-4}$$
$$\phi_{j=3,i=2} = \phi_{j=3,i=end-3}$$

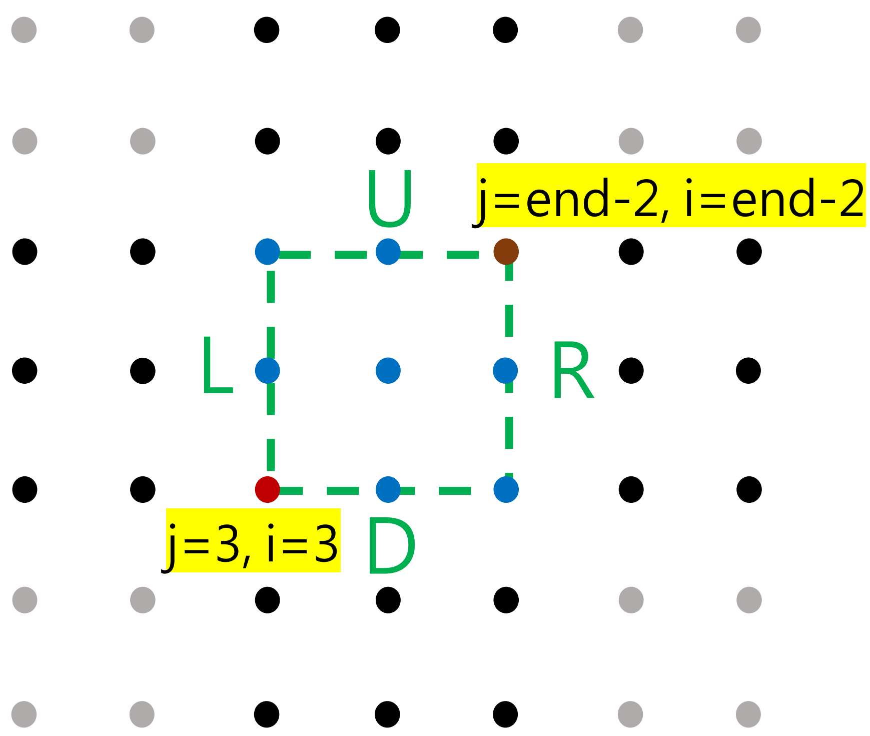
"D" boundary

$$\phi_{j=1,i=3} = \phi_{j=end-4,i=3}$$
$$\phi_{j=2,i=3} = \phi_{j=end-3,i=3}$$

Periodic BC – example



Mirror BC



"L" boundary

$$\eta_{j=3,i=1} = \eta_{j=3,i=5}$$
$$\eta_{j=3,i=2} = \eta_{j=3,i=4}$$

$$U_{j=3,i=1} = -U_{j=3,i=5}$$
$$U_{j=3,i=2} = -U_{j=3,i=4}$$

$$V_{j=3,i=1} = V_{j=3,i=5}$$
$$V_{j=3,i=2} = V_{j=3,i=4}$$

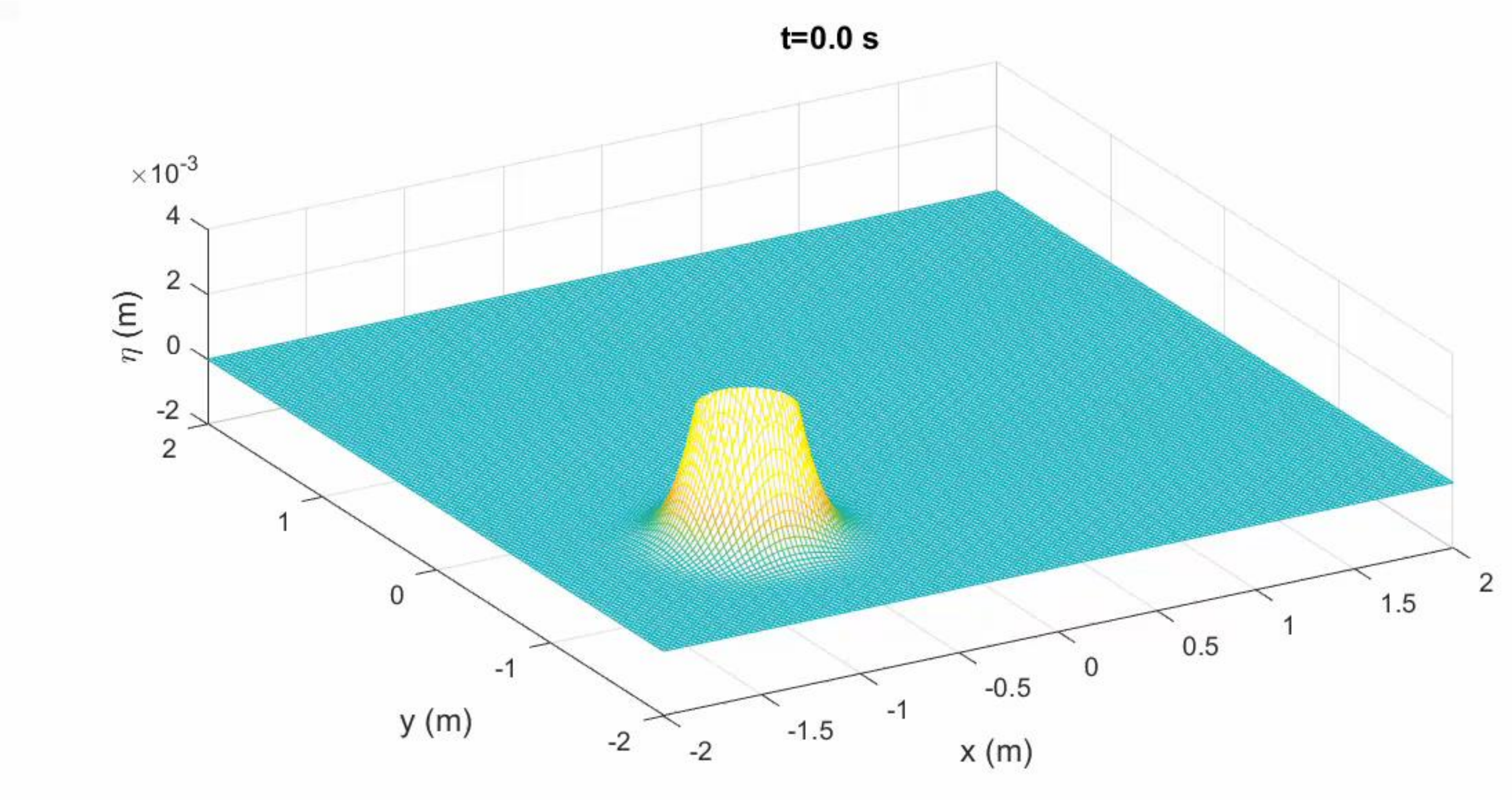
"D" boundary

$$\eta_{j=1,i=3} = \eta_{j=5,i=3}$$
$$\eta_{j=2,i=3} = \eta_{j=4,i=3}$$

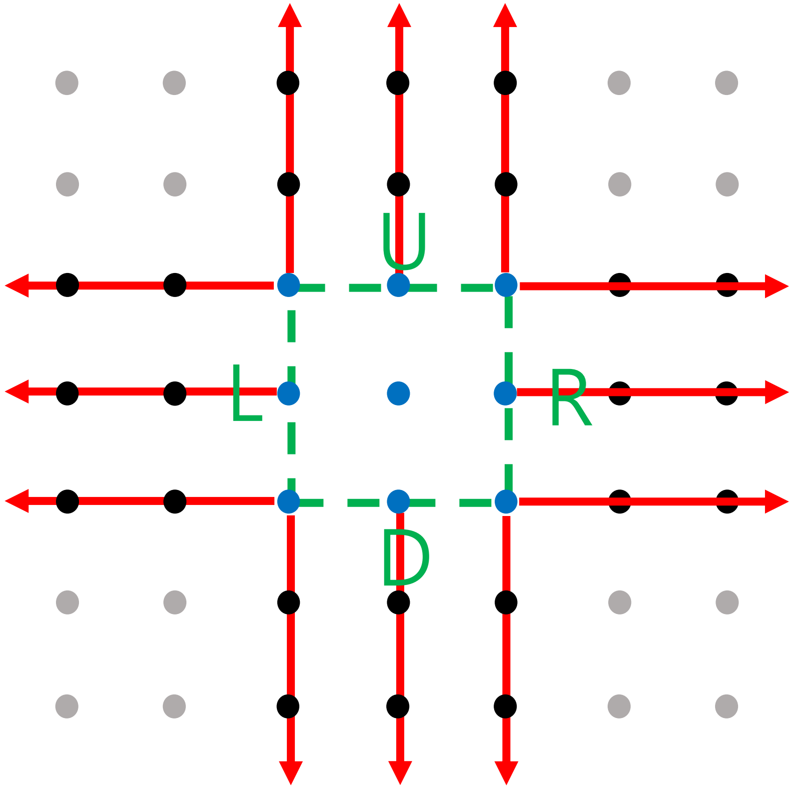
$$U_{j=1,i=3} = U_{j=5,i=3}$$
$$U_{j=2,i=3} = U_{j=4,i=3}$$

$$V_{j=1,i=3} = -V_{j=5,i=3}$$
$$V_{j=2,i=3} = -V_{j=4,i=3}$$

Mirror BC – example



Open BC – the 1DH method



ASSUME waves travel in these directions

PROBLEM: waves can travel in different directions…

"L" boundary

$$\eta_{j,i=2} = \eta_{j,i=3} - \frac{\Delta x}{C_{j,i=3}} \left(\frac{\partial \eta}{\partial t}\right)_{j,i=3}$$

$$\eta_{j,i=1} = \eta_{j,i=3} - \frac{2\Delta x}{C_{j,i=3}} \left(\frac{\partial \eta}{\partial t}\right)_{j,i=3}$$

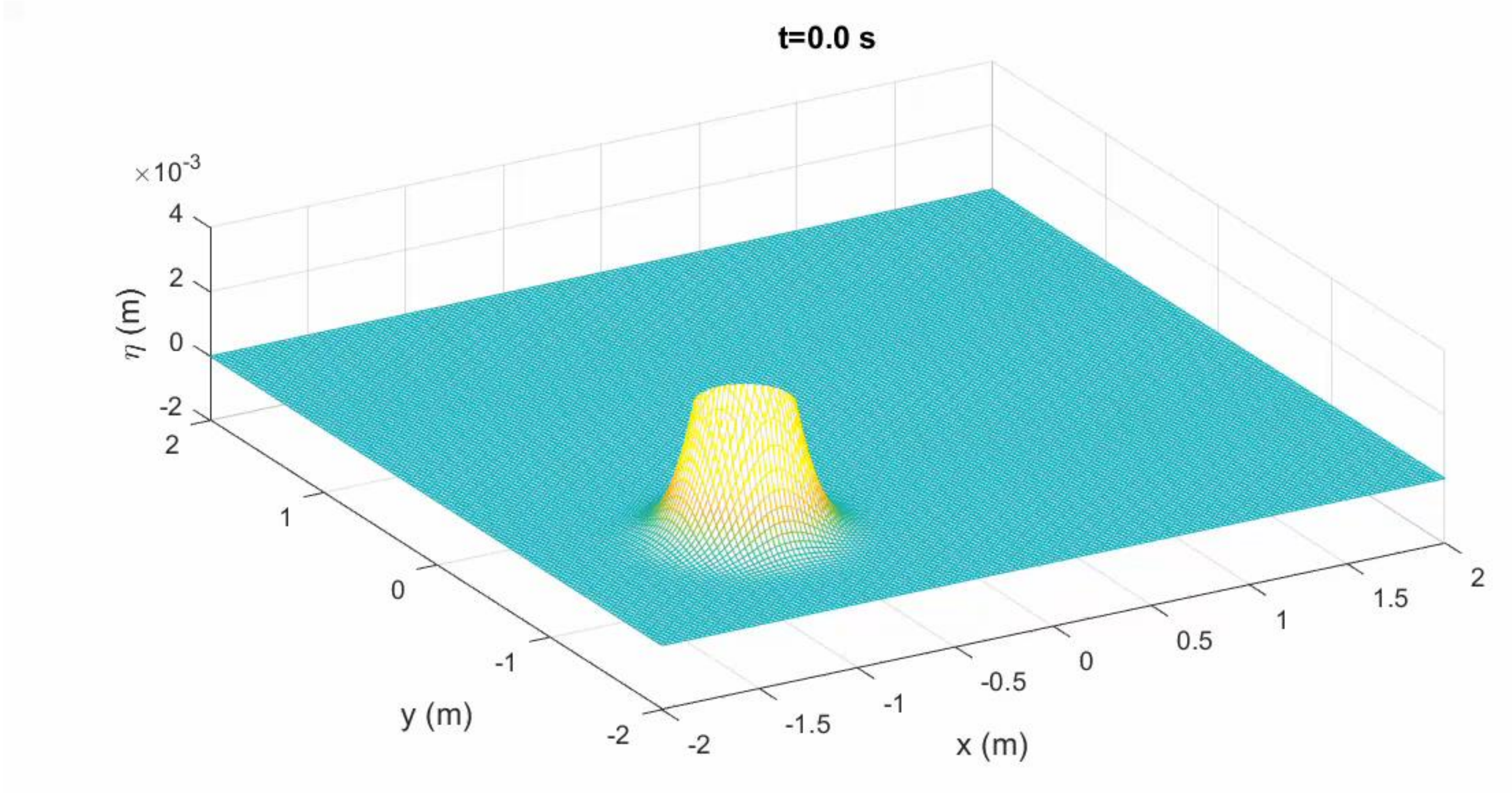
$$U_{j,i=2} = -\frac{C_{j,i=2}}{h_{j,i=2}} \eta_{j,i=2}$$

$$U_{j,i=1} = -\frac{C_{j,i=1}}{h_{j,i=1}} \eta_{j,i=1}$$

$$V_{j,i=2} = V_{j,i=4}$$

$$V_{j,i=1} = V_{j,i=5}$$

Open BC – the 1DH method – example



Open BC – more accurate methods

1. allow for different incident angles θ_i to the boundary
2. how to detect θ_i for different waves?
3. more assumptions and computation
4. see *Van Dongeren & Svendsen (1997)* for example

5. not always perfect
6. maybe just use sponge layer...
7. see *Lavelle & Thacker (2007)* for more discussion

Wall BC (bad, do not use)

"L" boundary (x=0)

$$U = 0$$

$$\frac{\partial \eta}{\partial x} = 0$$

(momentum equation)

$$\frac{\partial V}{\partial x} = 0$$

(assumed)

$$\frac{\partial U}{\partial x} = -\frac{1}{h}\left(\frac{\partial \eta}{\partial t} + \frac{\partial Vh}{\partial y}\right)$$

(mass equation)

$$\eta_{j,i=2} = \eta_{j,i=4}$$
$$\eta_{j,i=1} = \eta_{j,i=5}$$

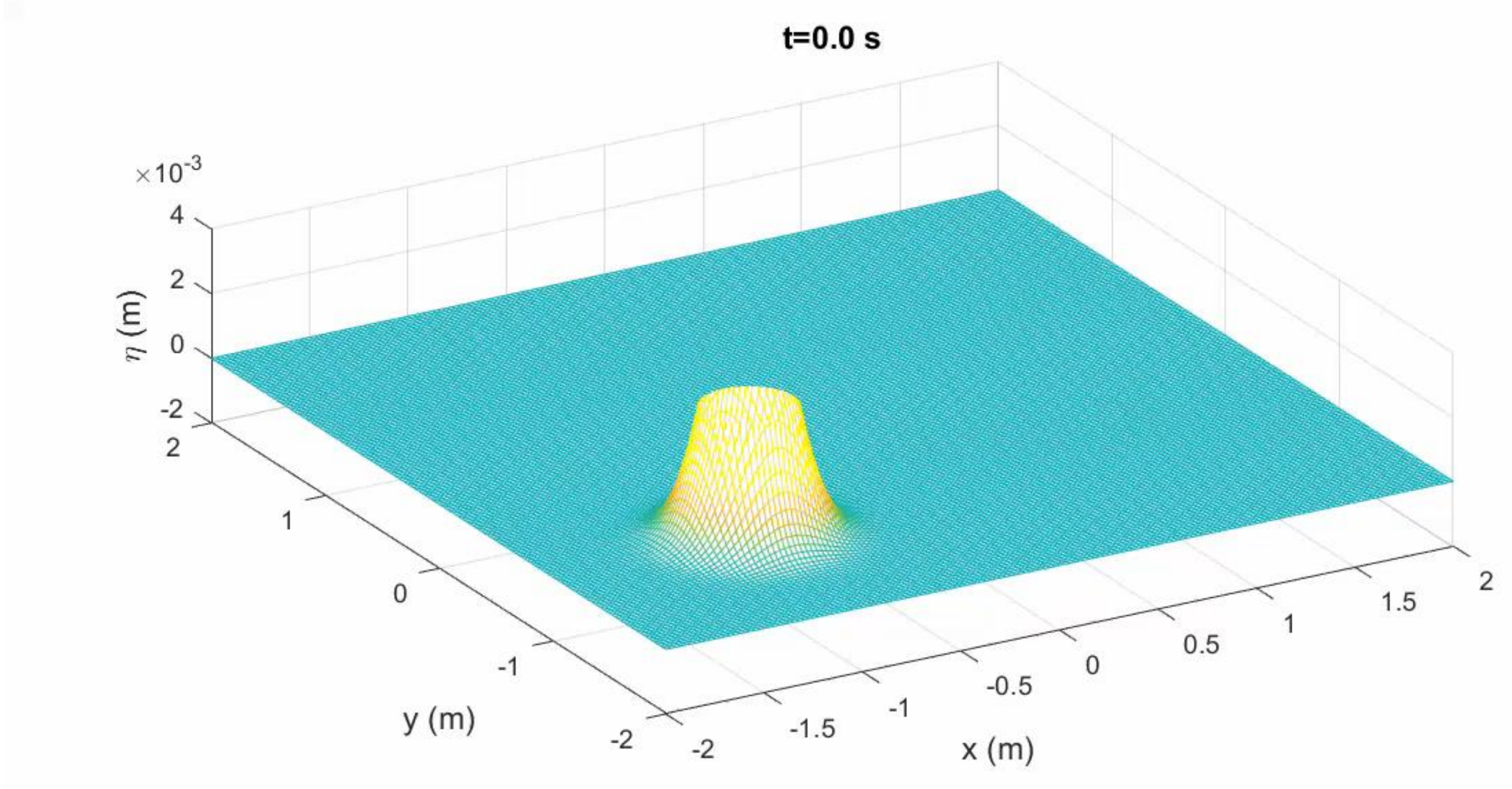
$$V_{j,i=2} = V_{j,i=4}$$
$$V_{j,i=1} = V_{j,i=5}$$

$$U_{j,i=3} = 0$$

$$U_{j,i=2} = \frac{\Delta x}{h_{i=3}}\left(\frac{\partial \eta}{\partial t} + \frac{\partial Vh}{\partial y}\right)_{i=3}$$

$$U_{j,i=1} = \frac{2\Delta x}{h_{i=3}}\left(\frac{\partial \eta}{\partial t} + \frac{\partial Vh}{\partial y}\right)_{i=3}$$

Wall BC – example



numerical unstable, too many assumptions, just use mirror BC

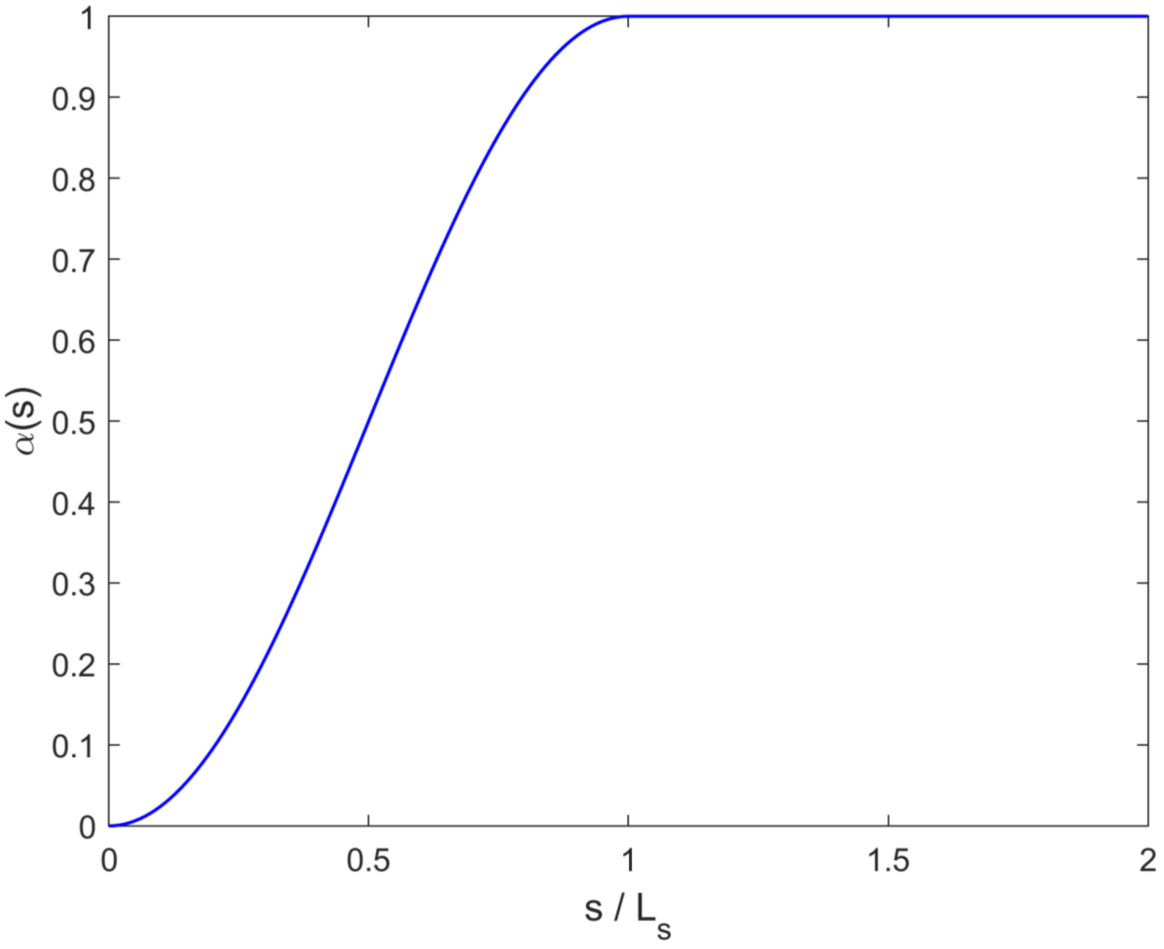
Sponge layer – Tonelli & Petti (2009)

in each iteration, multiply the results by the decay coefficient

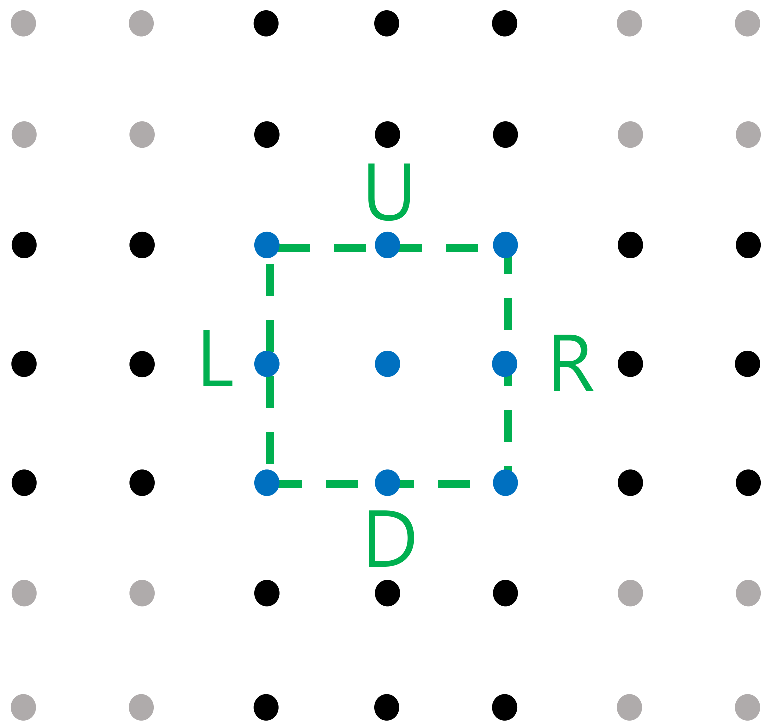
$$\alpha(s) = \begin{cases} 0.5 - 0.5\cos\left(\frac{s\pi}{L_s}\right), & s \le L_s \\ 1, & s > L_s \end{cases}$$

s: distance to the boundary

L_s: sponge layer length



Sponge layer



$\eta_{j,i}$ decayed by all boundaries

$$\eta_{j,i} = \eta_{j,i}\, \alpha_L \alpha_R \alpha_D \alpha_U$$

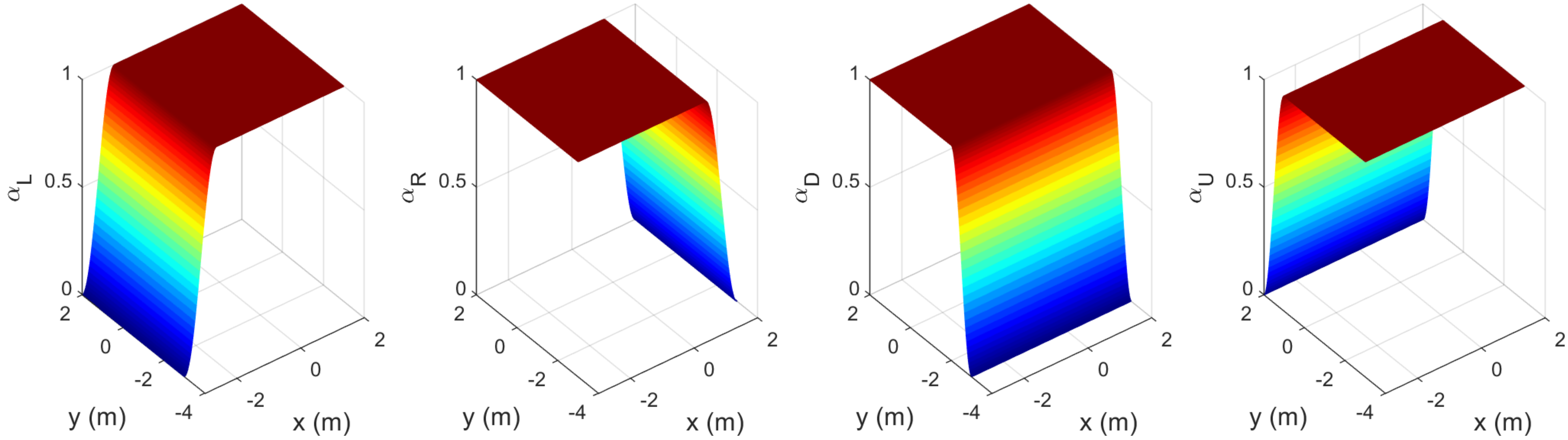
$U_{j,i}$ decayed by L and R boundaries

$$U_{j,i} = U_{j,i}\, \alpha_L \alpha_R$$

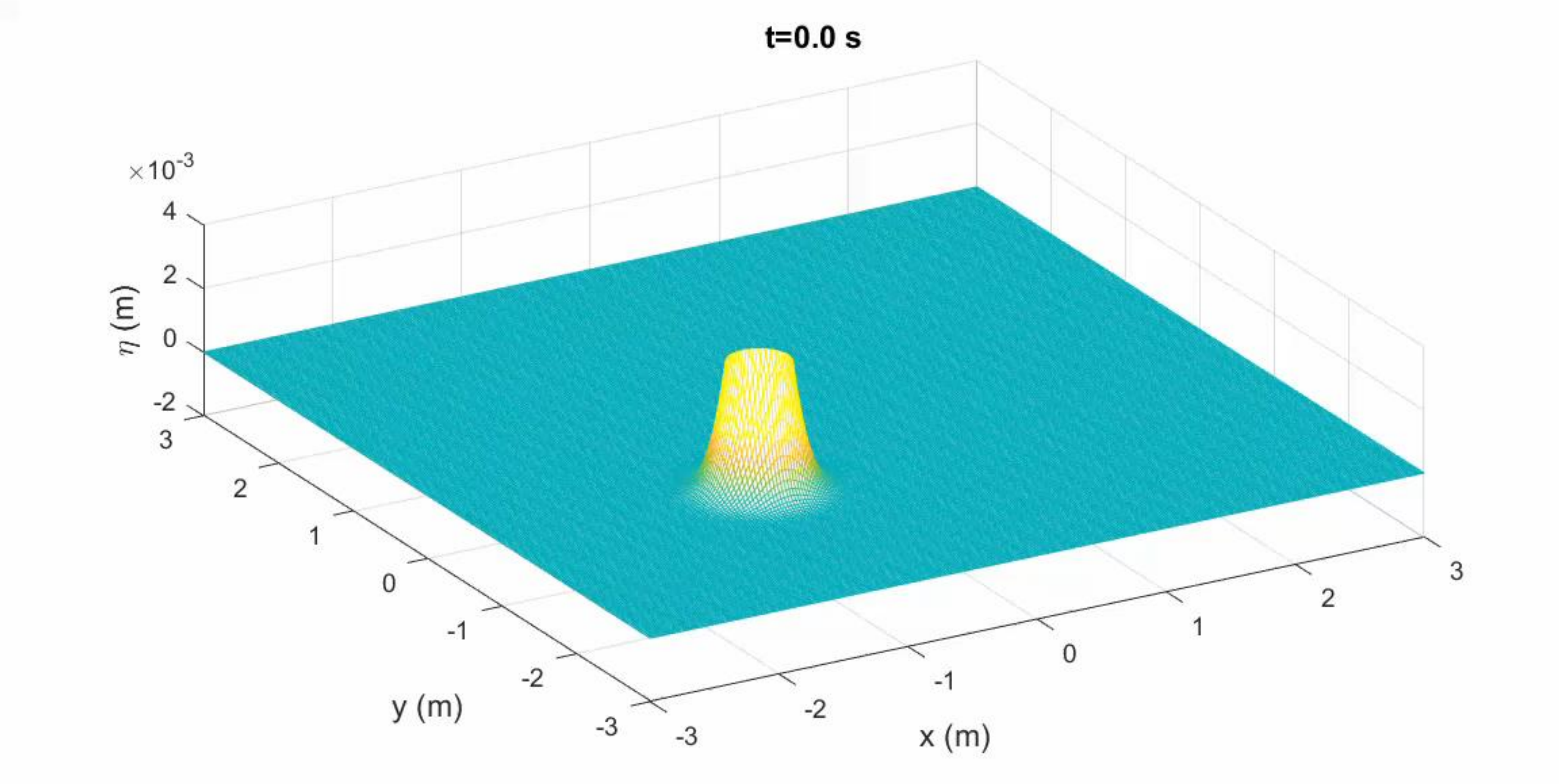
$V_{j,i}$ decayed by D and U boundaries

$$V_{j,i} = V_{j,i}\, \alpha_D \alpha_U$$

Decay coefficient examples



Sponge layer – example



seems to perform a little bit better than the simple open BC

Storage in 2DH

1. 2DH data are 3D – (x, y, t)

2. storage management is very important in 2DH

3. for quick computation on a laptop, try to not exceed
   nx * ny =100 * 100 = 10000 points

4. can reduce domain size if the problem is symmetric

Time stepping & saving

1. CFL says $\Delta t \leq C_{CFL} \frac{\min(\Delta x, \Delta y)}{\sqrt{g h_{max}}}$
2. but you can use a smaller Δt
3. Δt does not have to be the same in every time step
4. change Δt to achieve the desired save time

```
t=0:0.1:1;   %specify save time
ti_save=2;   %index for the save time
... ... (other code) ... ...
while t_now < t(end)   %time-marching iteration

        dt=Ct*min([dx dy])/sqrt(g*h_max);   %calculate standard time step

        if (t_now+dt)>t(ti_save)   %adjust time step if necessary
                dt=t(ti_save)-t_now;
                ti_save=ti_save+1;
        end
        ... ... (other code) ... ...
end
```