Development of Ultra-High Resolution Model Using Cut Cells

Hiroe Yamazaki and Takehiko Satomura
Kyoto University, Kyoto, Japan

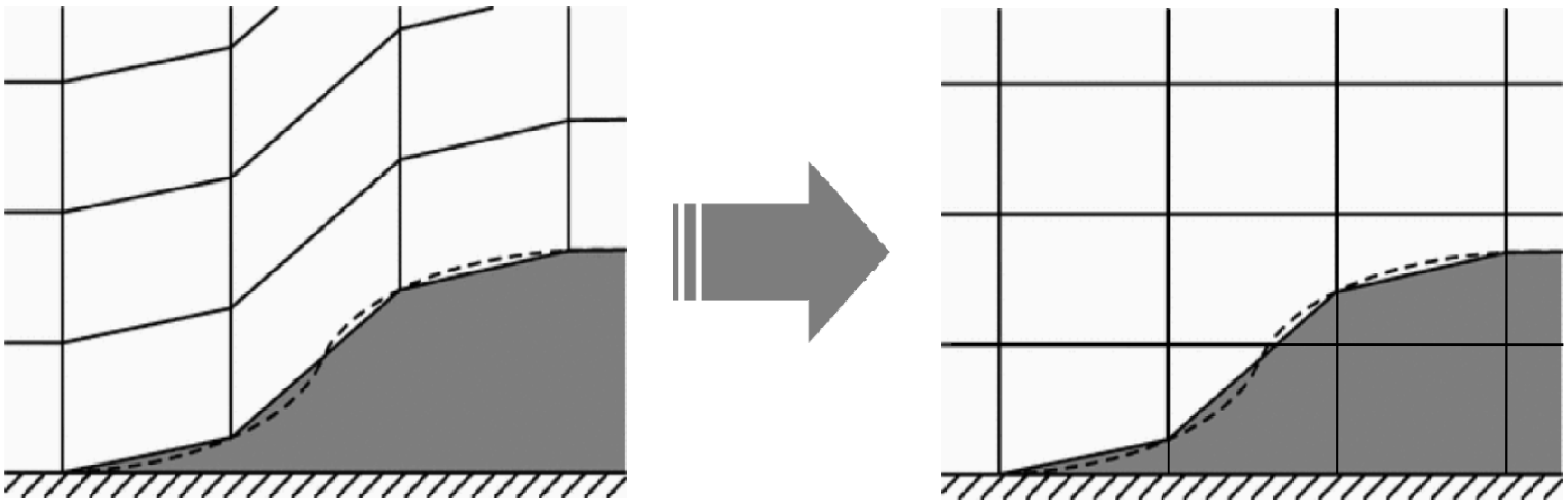
May 16, 2011 @ Bad Orb

This study explores the development of
a new atmospheric model for ultra-high resolution
simulations using Δx = O(10) m.

- One hurdle to ultra-high resolution simulations is handling of steep slopes in mountainous areas.
- Conventional terrain-following vertical coordinates introduce significant errors around steep slopes.

<u>Cartesian coordinates are an attractive choice for next-generation atmospheric models.</u>

We are developing a new atmospheric model using
a cut cell representation of topography
as an alternative to the terrain-following approach.



terrain-following approach   cut cell approach

Nonhydrostatic atmospheric cut cell model “Sayaca-2D”

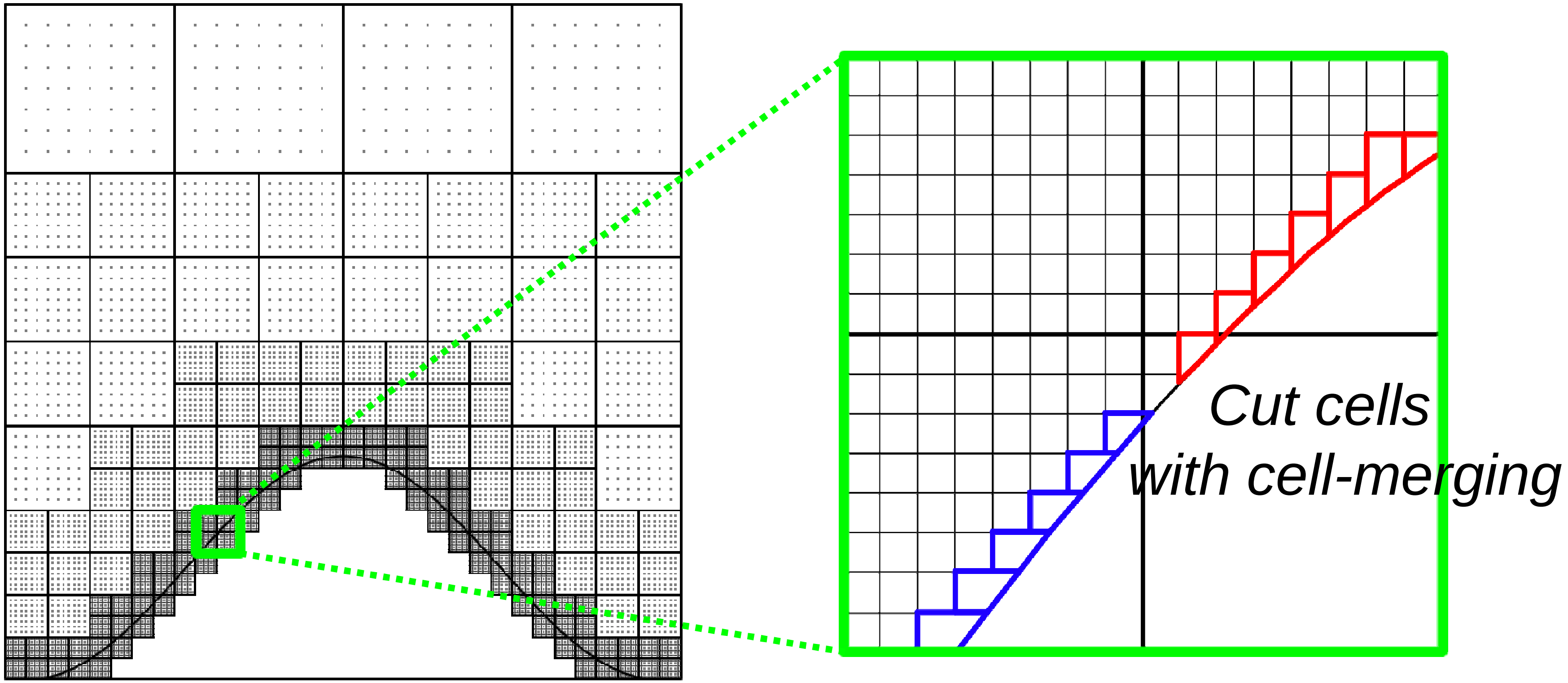
■ Dynamics

Dimension	2-D
Governing equations	Fully compressible (Satomura & Akiba 2003)
Variable arrangement	Semi-staggered (Yamazaki & Satomura 2010)
Spatial discretization	Finite Volume Method
Time integration	Leap frog with Asselin filter (All explicit)
Topography	Cut cell method with cell-merging
Numerical smoothing	4th order artificial diffusion

■ Physics

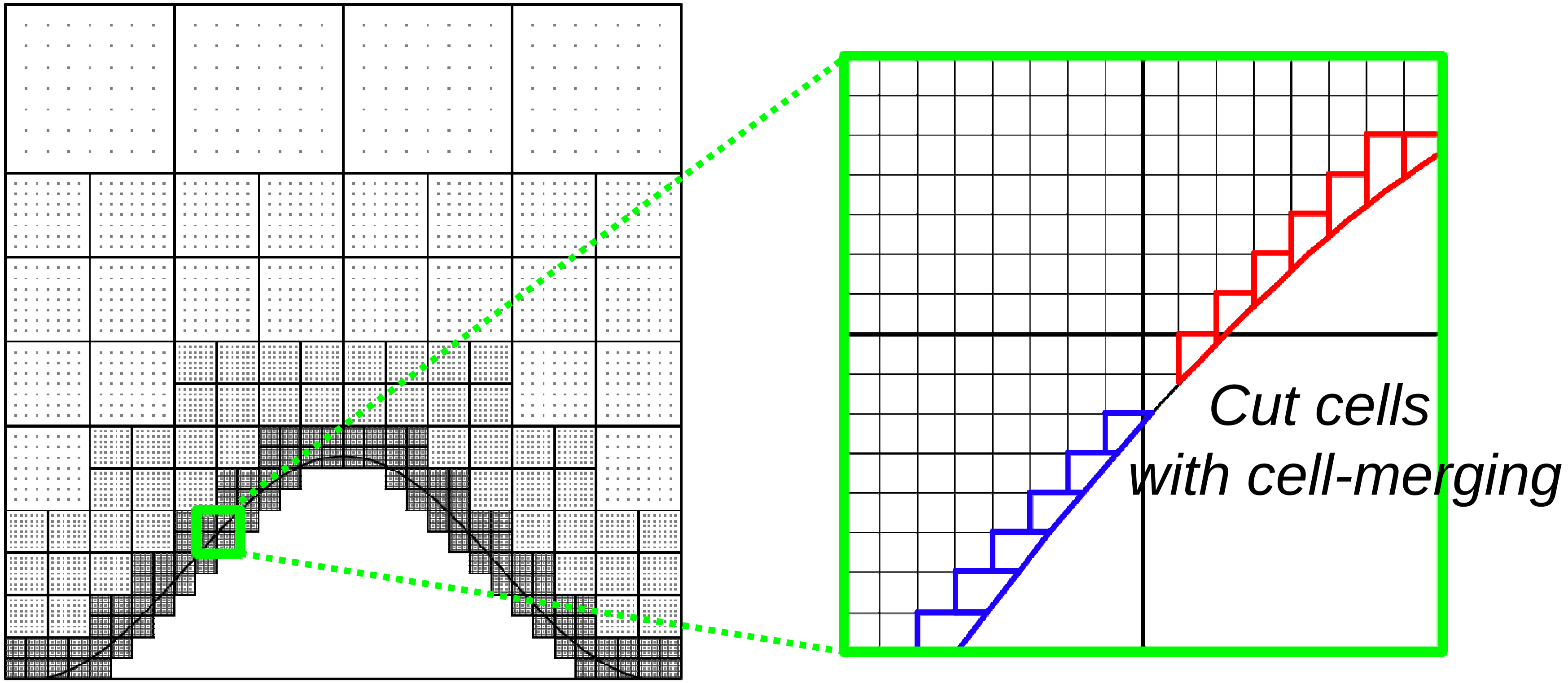
Subgrid turbulence parameterization	1.5 order（Klemp and Wilhelmson 1974）

Block-structured Cartesian mesh refinement



- A flow field is divided into a number of blocks, and each block has the same number of cells.
- A locally refined grid at an arbitrary boundary is obtained by refining the size of blocks near the boundary.

Block-structured Cartesian mesh refinement



- A uniform Cartesian mesh in each block allows the direct use of any existing Cartesian grid code.
- Using the same number of cells in each block makes the load balance of each block equivalent (e.g. Nakahashi 2003; Jablonowski et al. 2006, 2009).

Key techniques in applying block-structured mesh refinement to atmospheric models

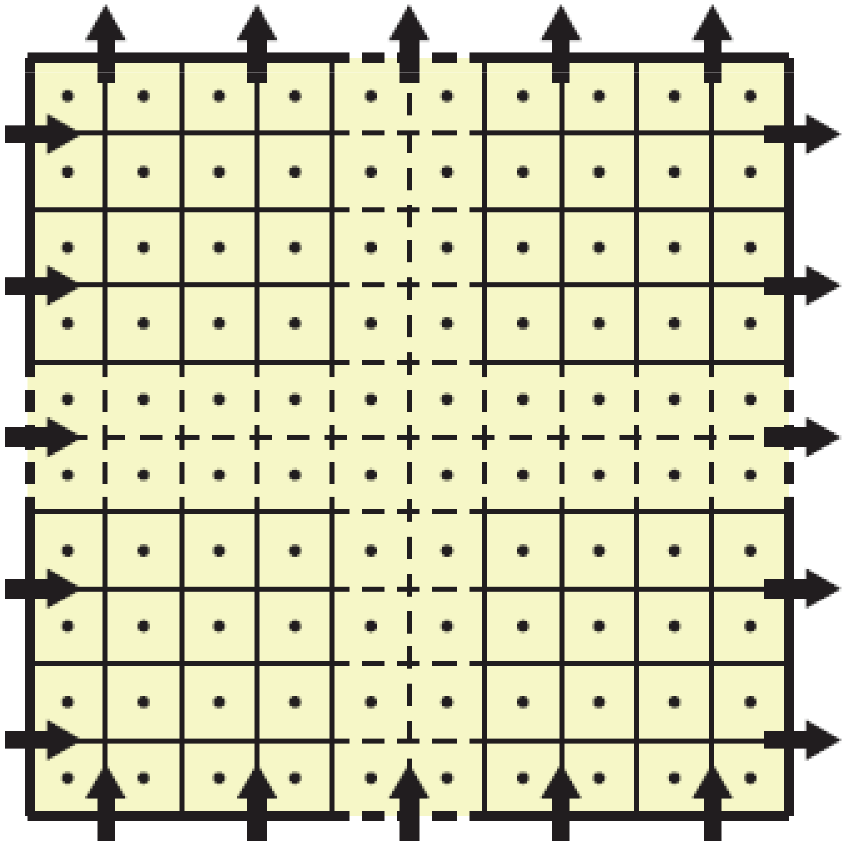
(1) Subcycling time integration

Use larger time steps at coarse grids in order to minimize the overhead in time integration.

(2) Flux-matching at fine-coarse grid interfaces

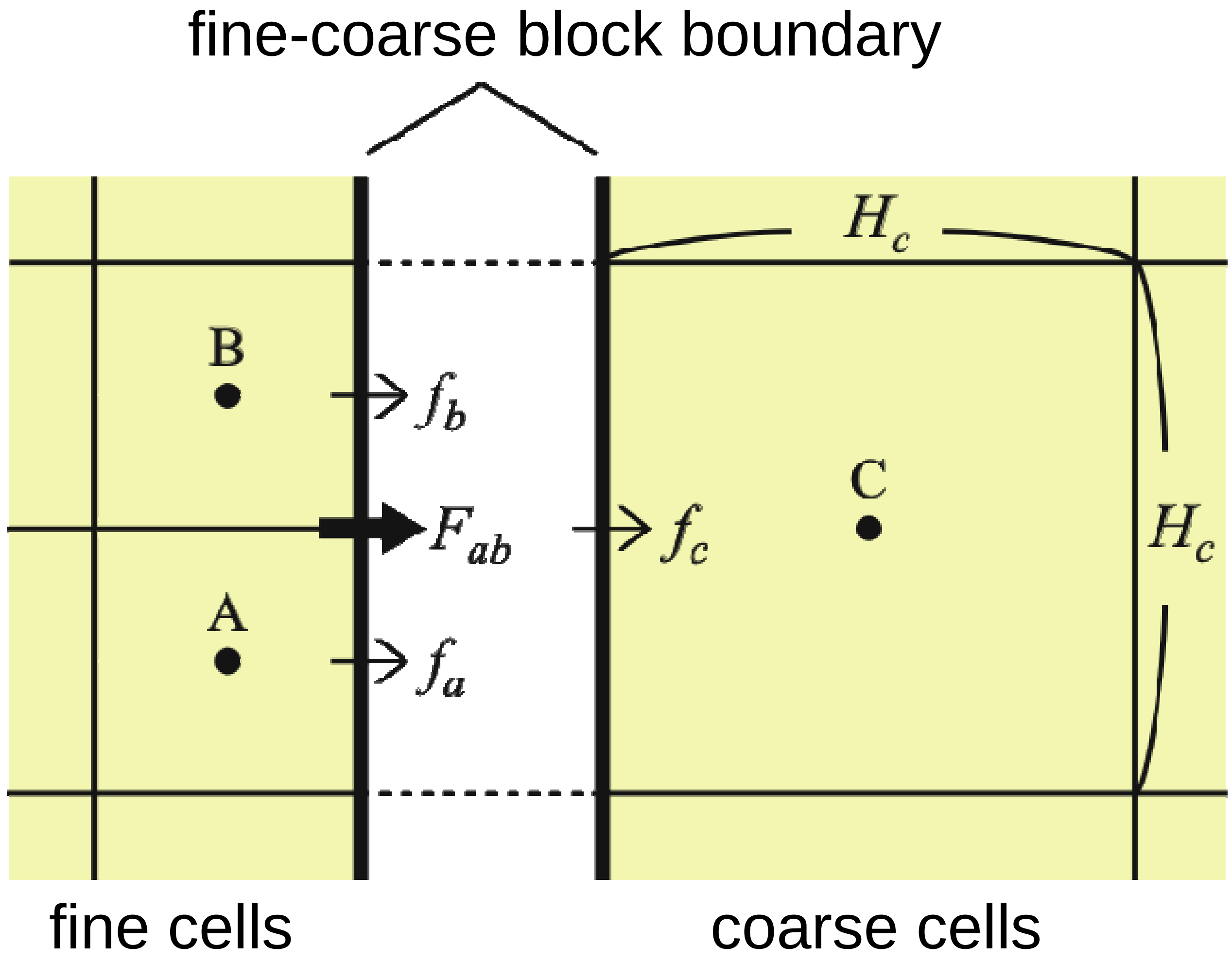
Correct the coarse grid flux to match the accumulated fine grid fluxes in order to ensure mass conservation (Berger and Colella, 1989).

We use a subcycled time step with introducing
a simple flux-matching algorithm
suitable for block-structured Cartesian mesh.



Define '*block boundary flux*' on each block side.

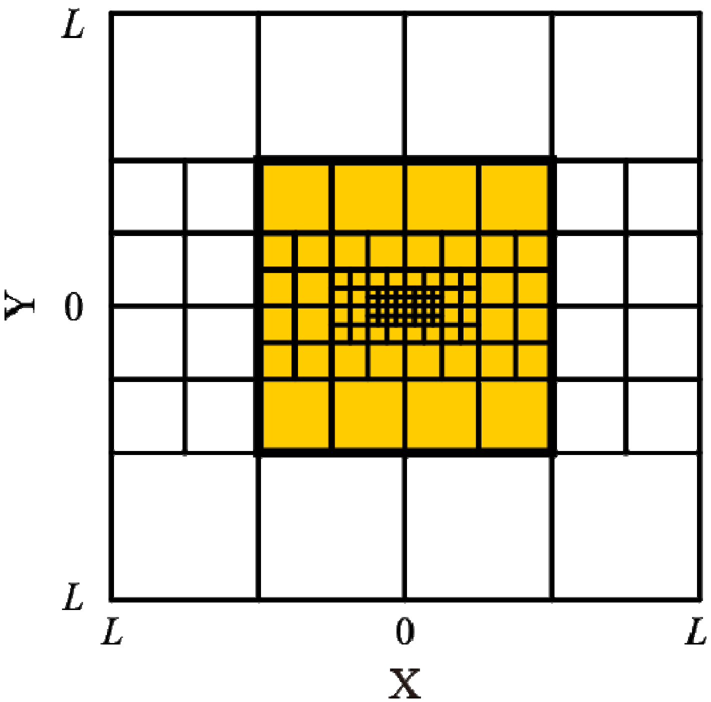
Flux matching algorithm



$$F_{ab}^{n\Delta t} = \frac{1}{4}\left(f_a^{n\Delta t} + f_b^{n\Delta t} + f_a^{(n+1/2)\Delta t} + f_b^{(n+1/2)\Delta t}\right)$$

Results

A heat diffusion problem was solved to confirm the conservation property and parallel efficiency.



112 blocks with 64x64 cells in each block.

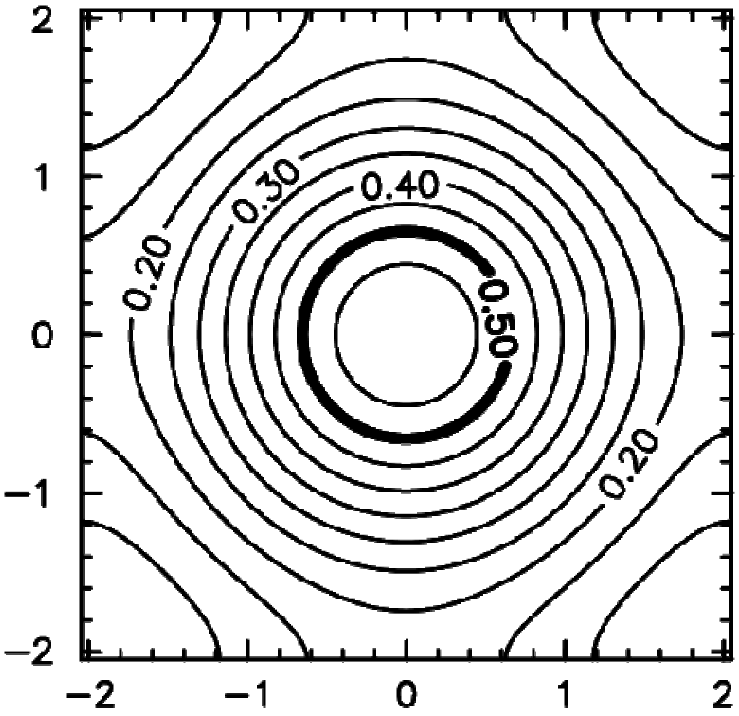
$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \quad \text{for} \quad -L \leq x, y \leq L$$

$$\frac{\partial T}{\partial x} = 0 \quad \text{at} \quad x = \pm L, \qquad \frac{\partial T}{\partial y} = 0 \quad \text{at} \quad y = \pm L$$

$$T(x, y, 0) = \begin{cases} 1 & \text{in the heated region} \\ 0 & \text{in other regions} \end{cases}$$

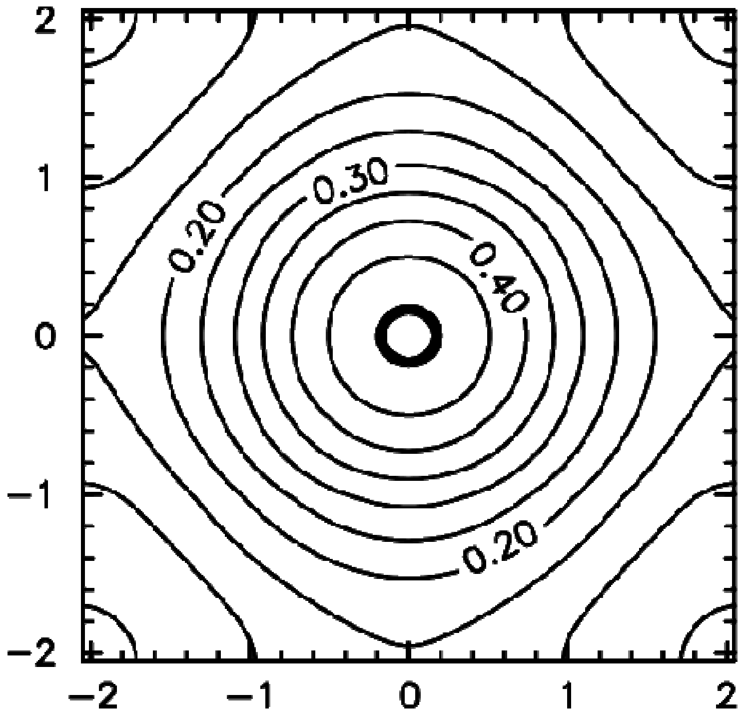
Simulated temperature fields

T (t = 100 hours)



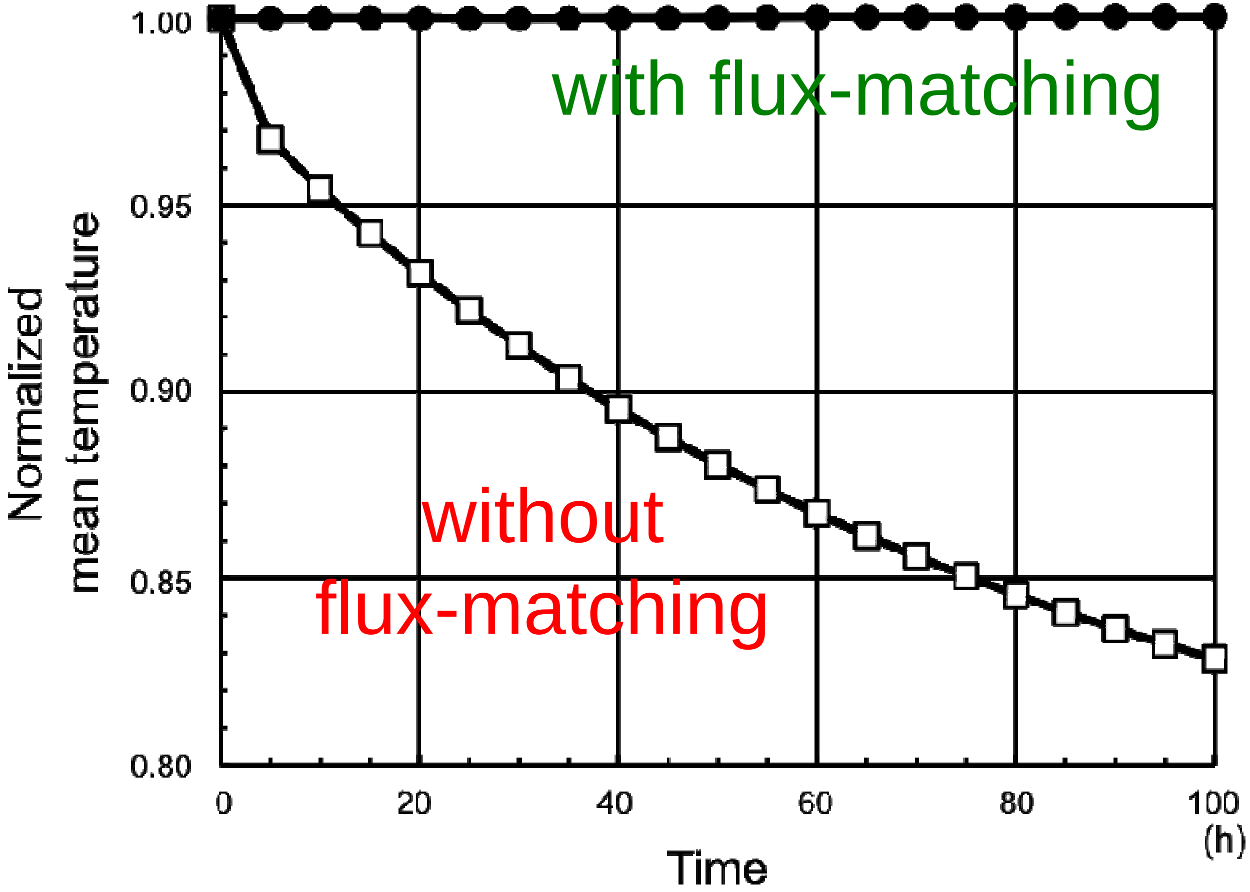
with flux-matching

T (t = 100 hours)



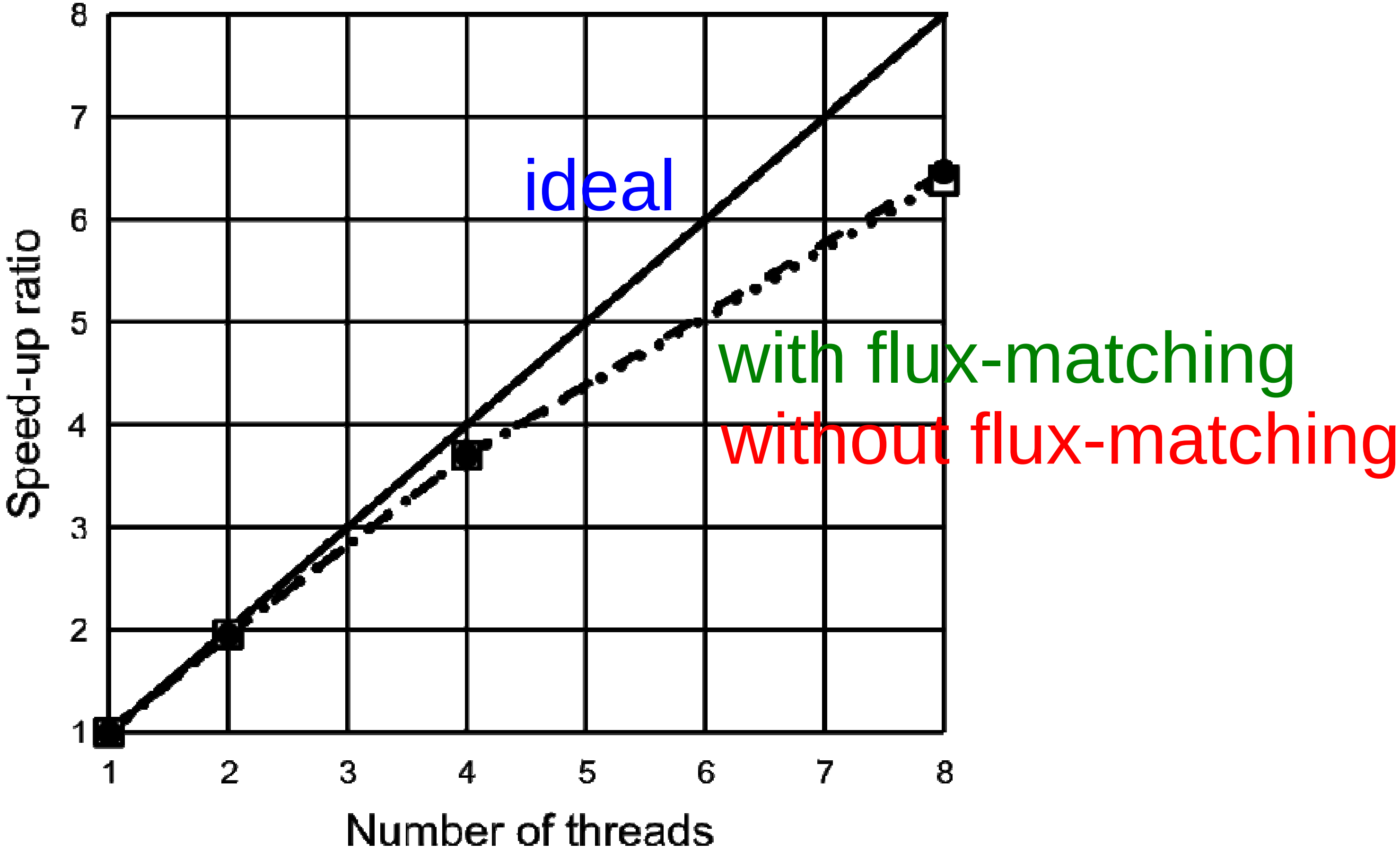
without flux-matching

Change of the mean temperature



Our flux-matching algorithm shows an excellent conservation property.

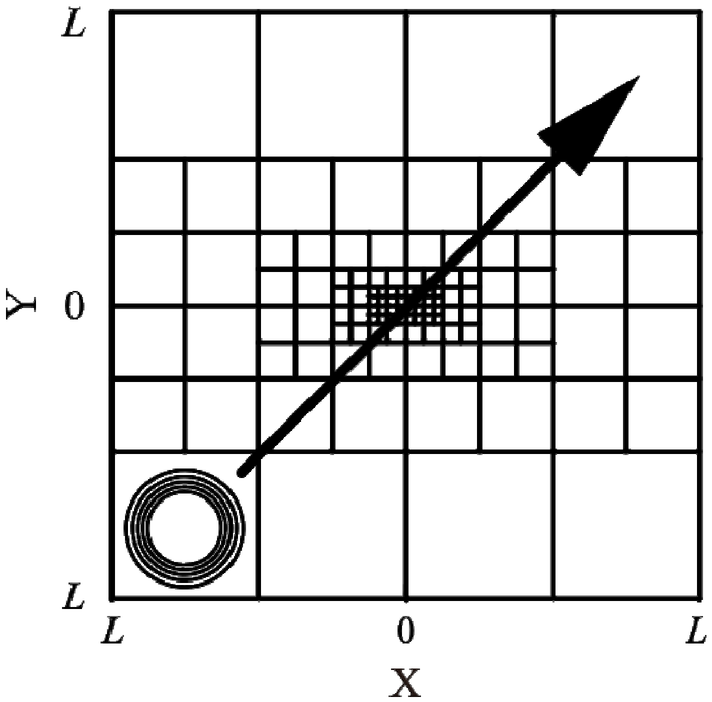
Comparison of speed-up ratio



The flux-matching algorithm keeps the high parallel efficiency.

Advection problem

A simple advection problem was solved to examine the accuracy of the method.



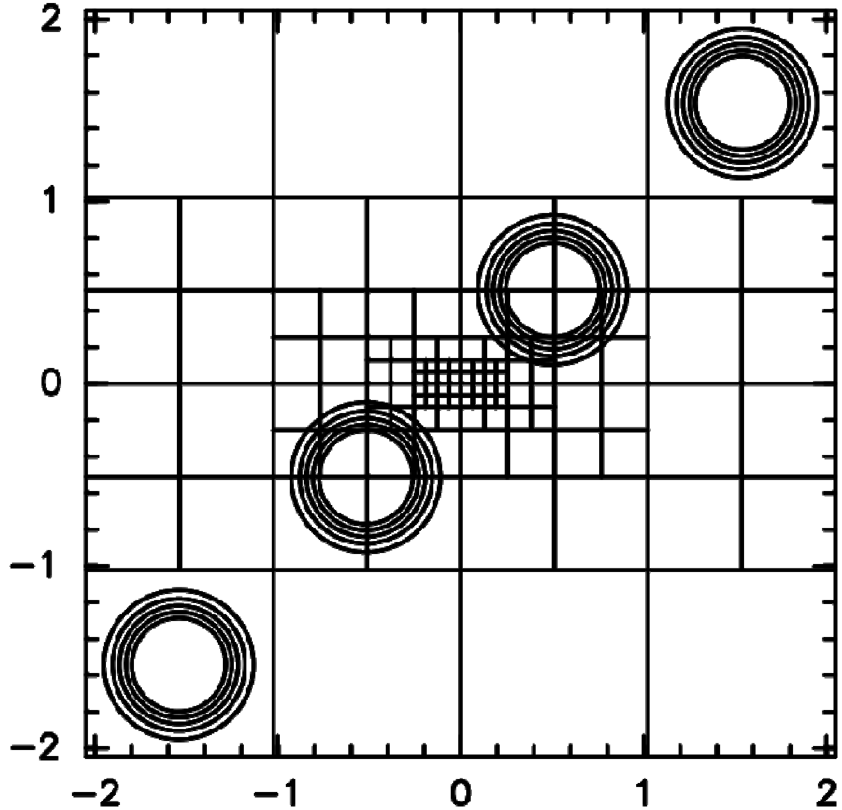
$$\frac{\partial \phi}{\partial t} = -c_x \frac{\partial \phi}{\partial x} - c_y \frac{\partial \phi}{\partial y} \quad \text{for} \quad -L \le x, y \le L,$$

where $c_x = c_y = 1.$

A cosine bell is advected diagonally across the domain.

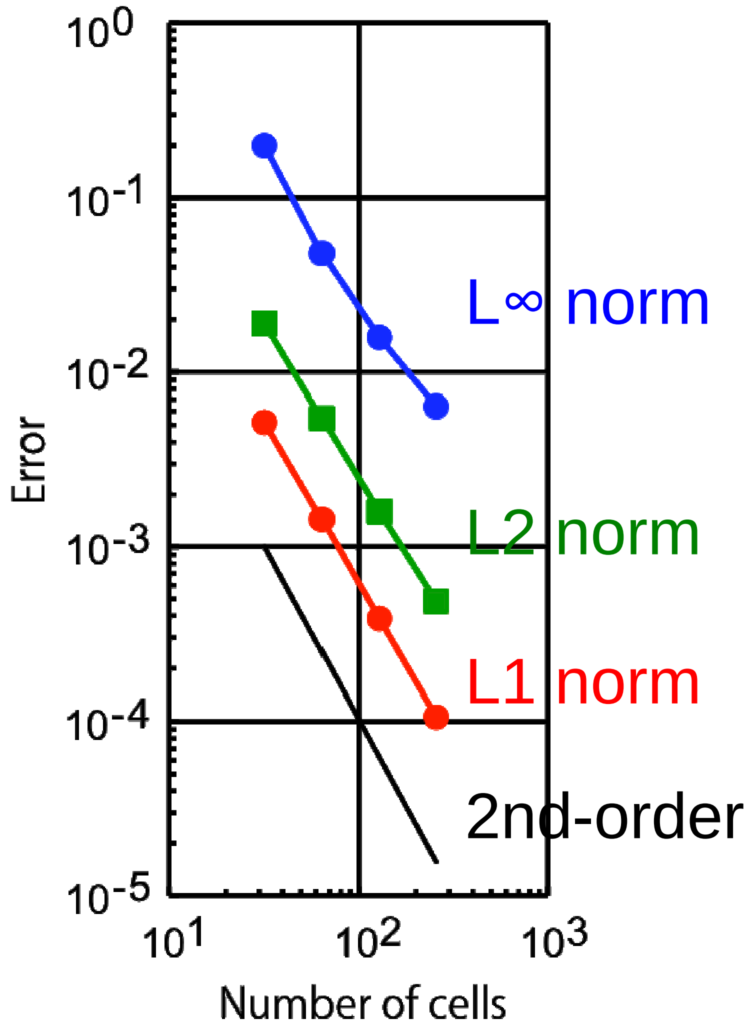
A block has 64x64 cells.

Snapshots of the cosine bell



No visible distortion is found as the cosine bell passes over various mesh resolutions.

Grid refinement study



The method has globally 2nd-order accuracy.

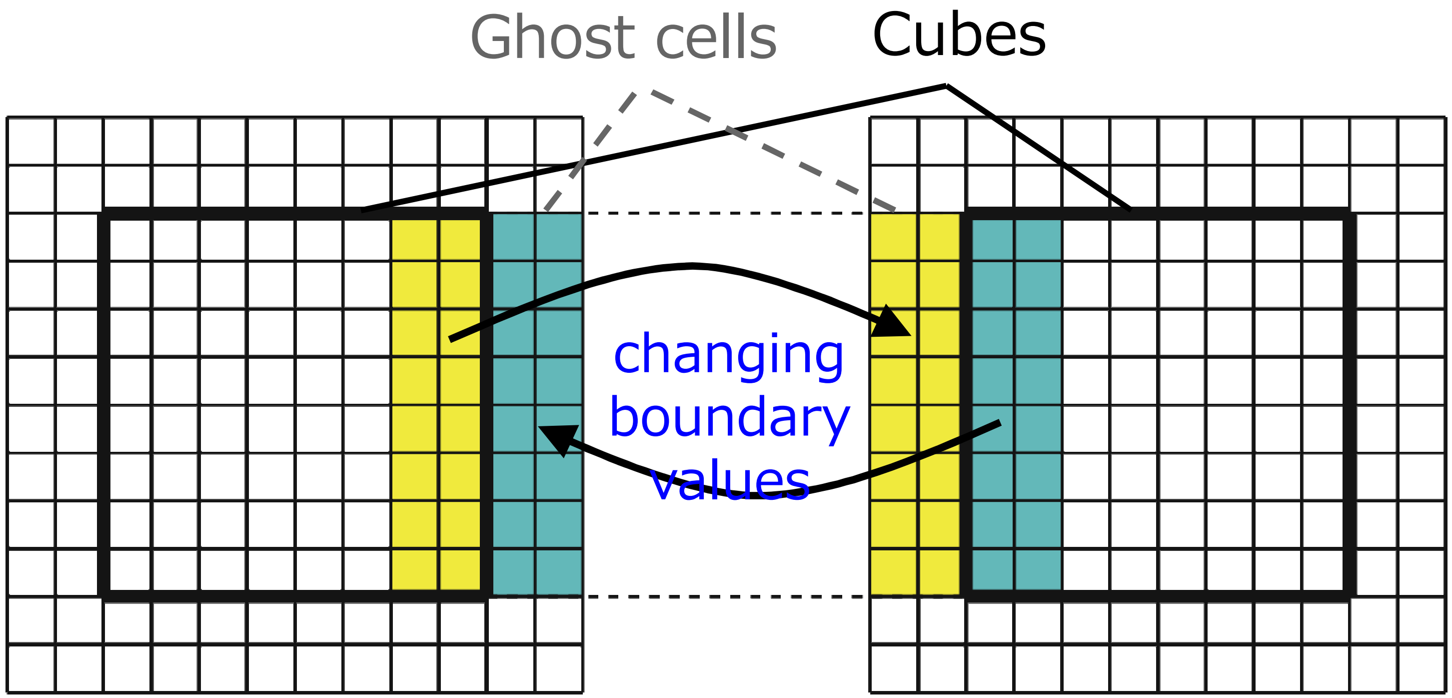
Summary

We are now introducing a block-structured mesh refinement to our cut cell model “Sayaca-2D”.

- Using the same number of cells in each block makes it easy to parallelize the model code.
- A simple flux-matching algorithm ensures mass conservation in subcycling time integration.
- Results of simple diffusion and advection problems showed sufficient accuracy and high computational efficiency of the method.

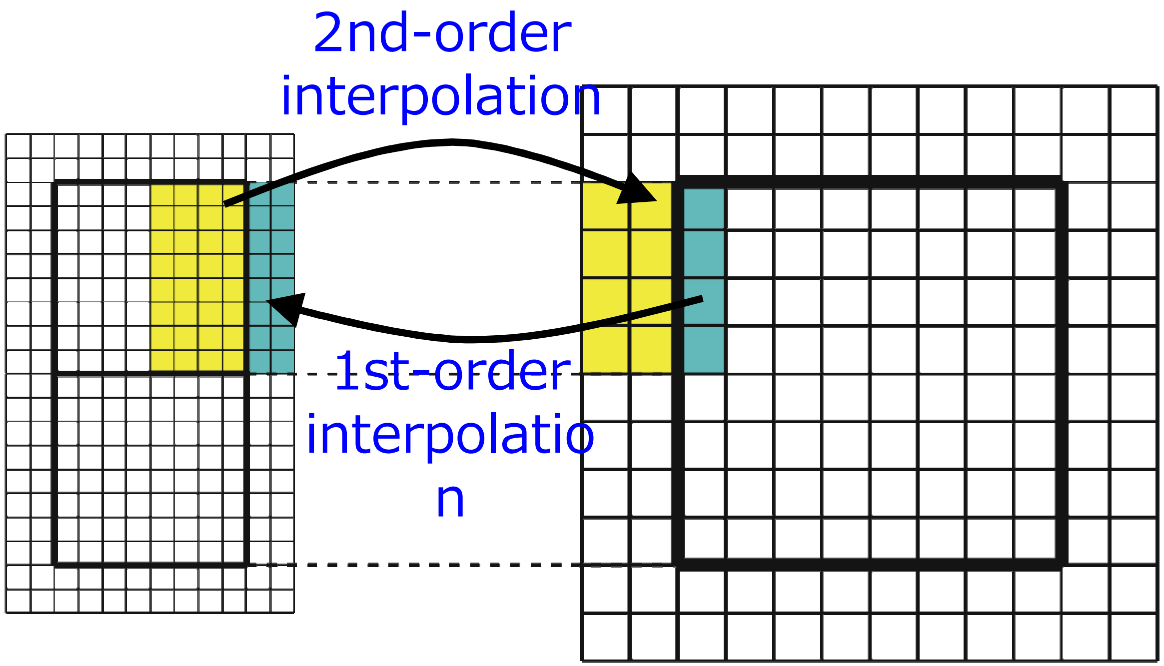
Data exchange

Between the same-size blocks

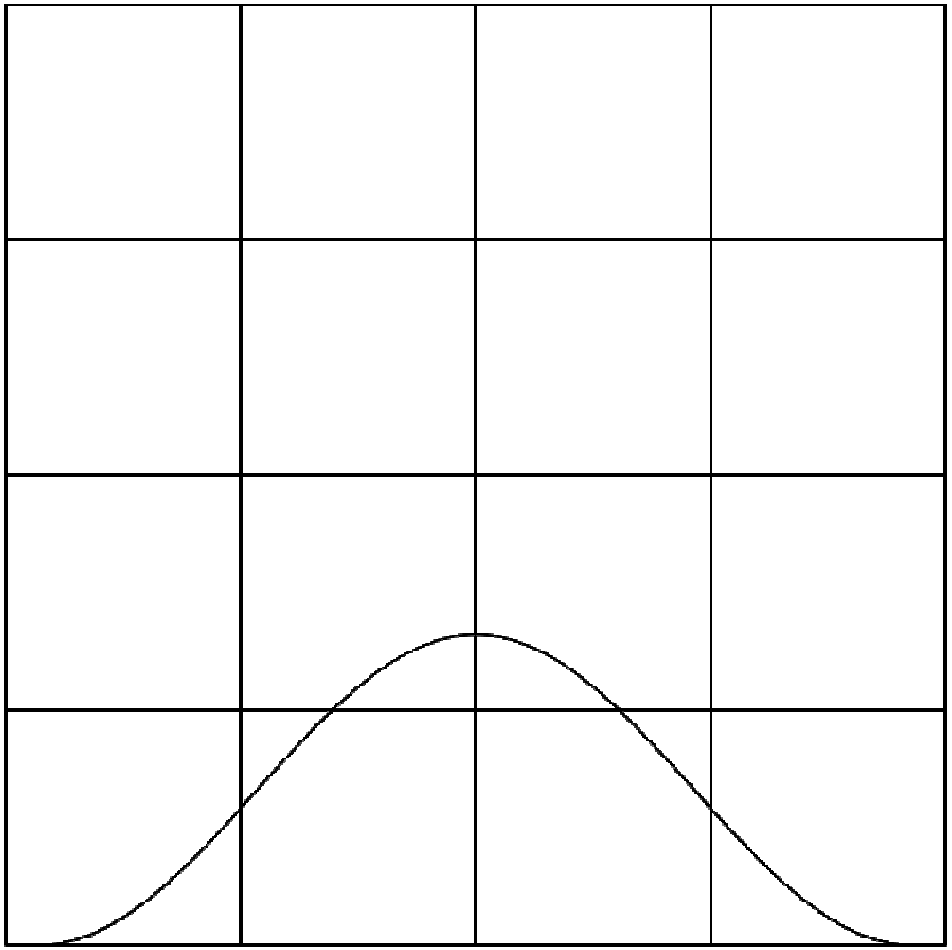


Data exchange

Between different sizes of blocks

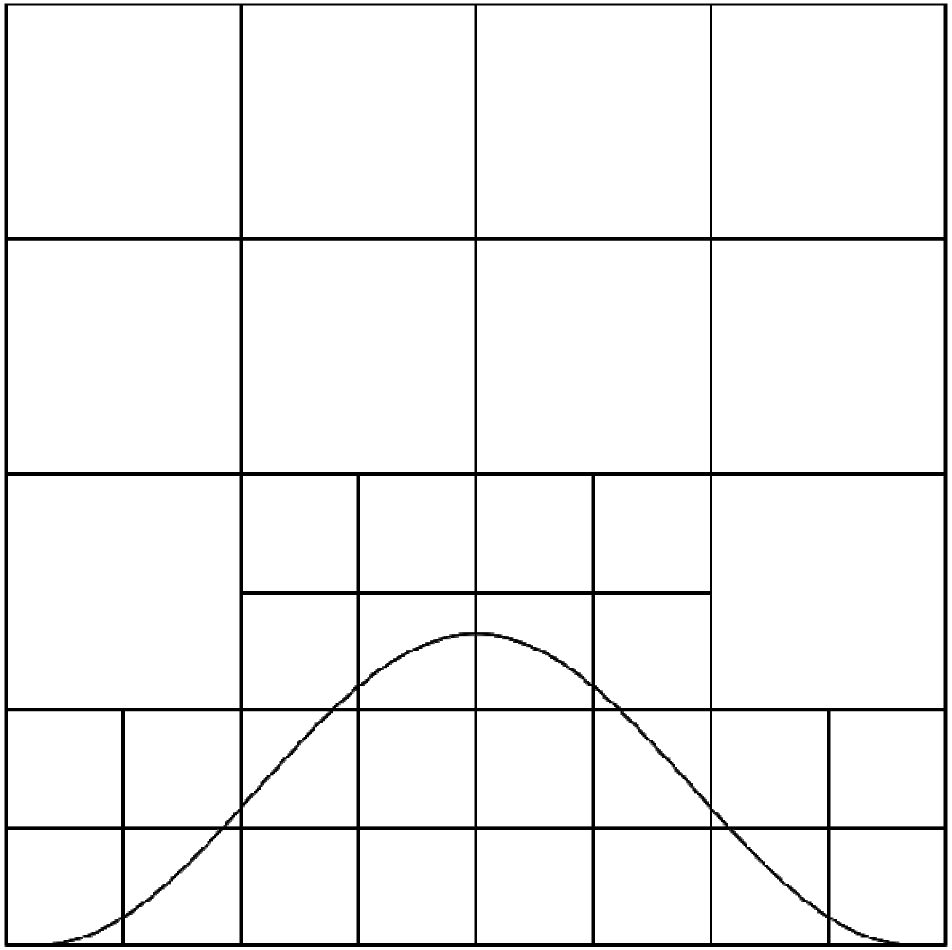


Mesh generation



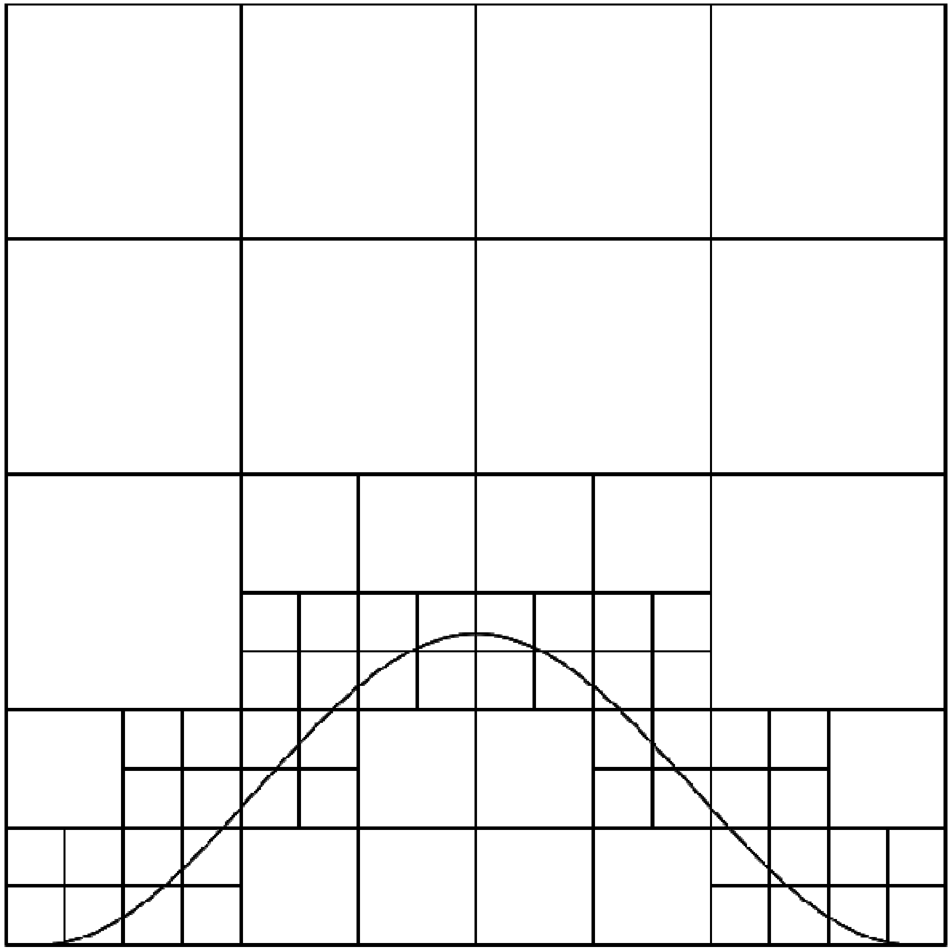
1. Divide the domain into a coarse Cartesian mesh.

Mesh generation



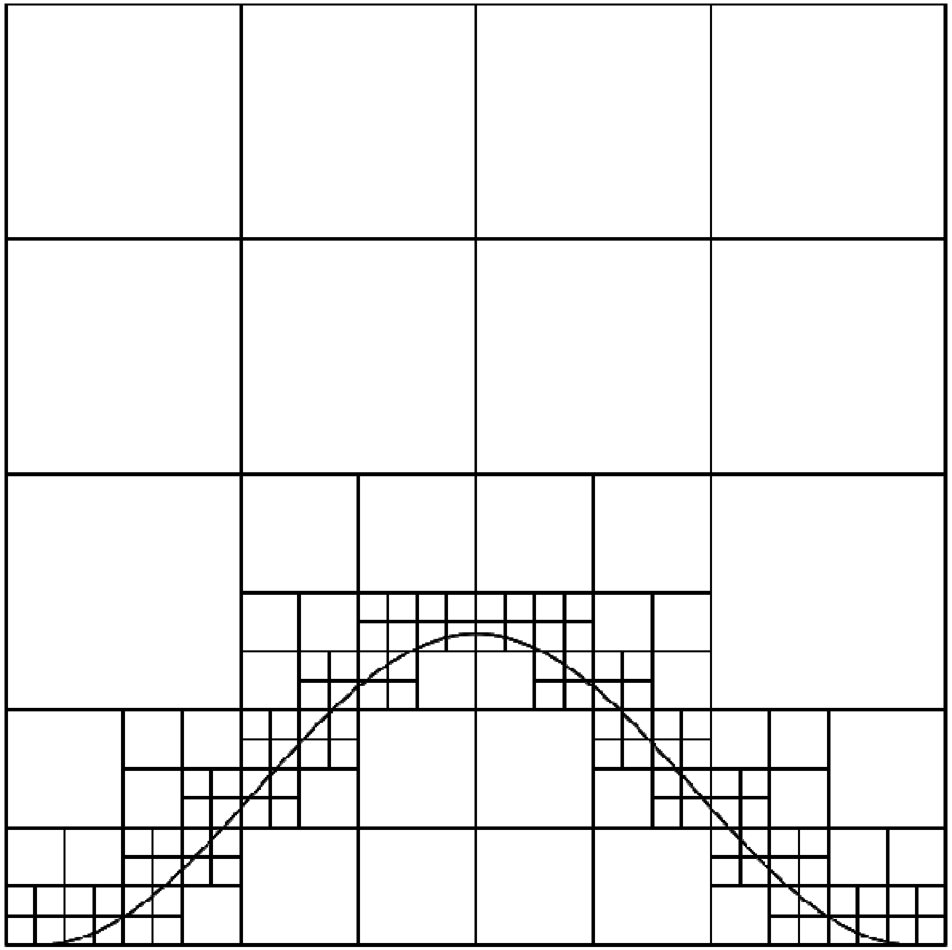
2. Divide blocks that cross the topography into 4 blocks.

Mesh generation



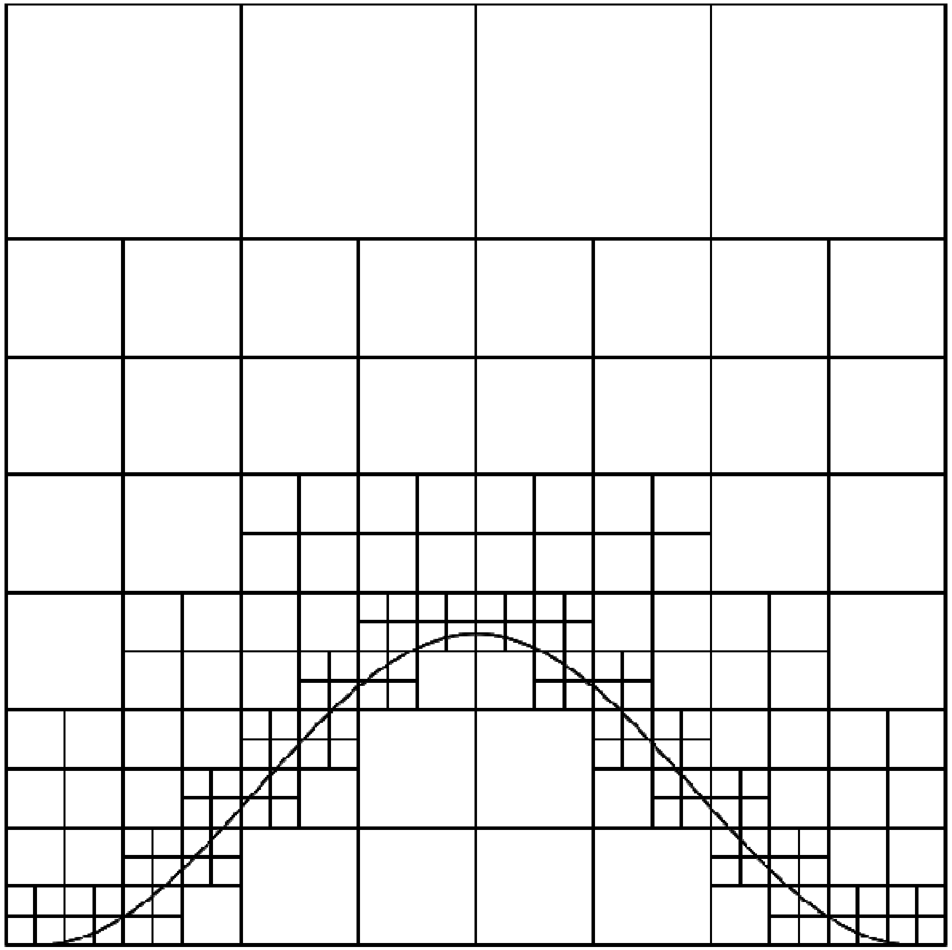
2. Divide blocks that cross the topography into 4 blocks.

Mesh generation



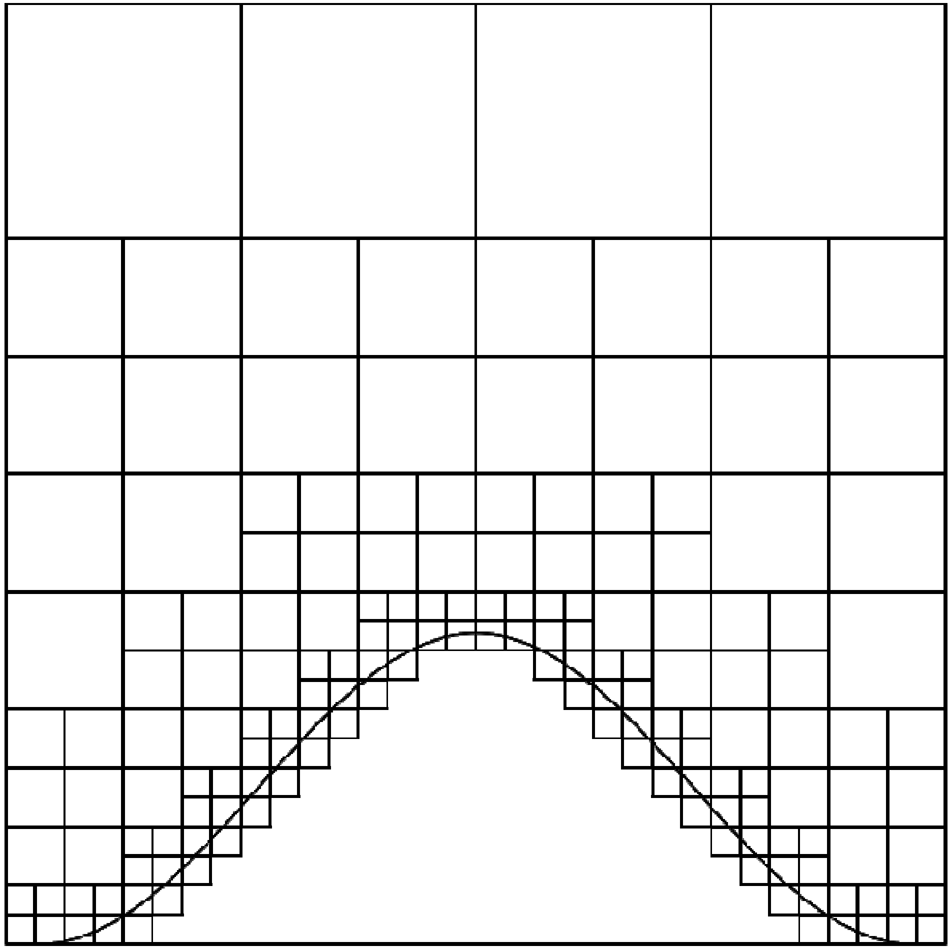
2. Divide blocks that cross the topography into 4 blocks.

Mesh generation



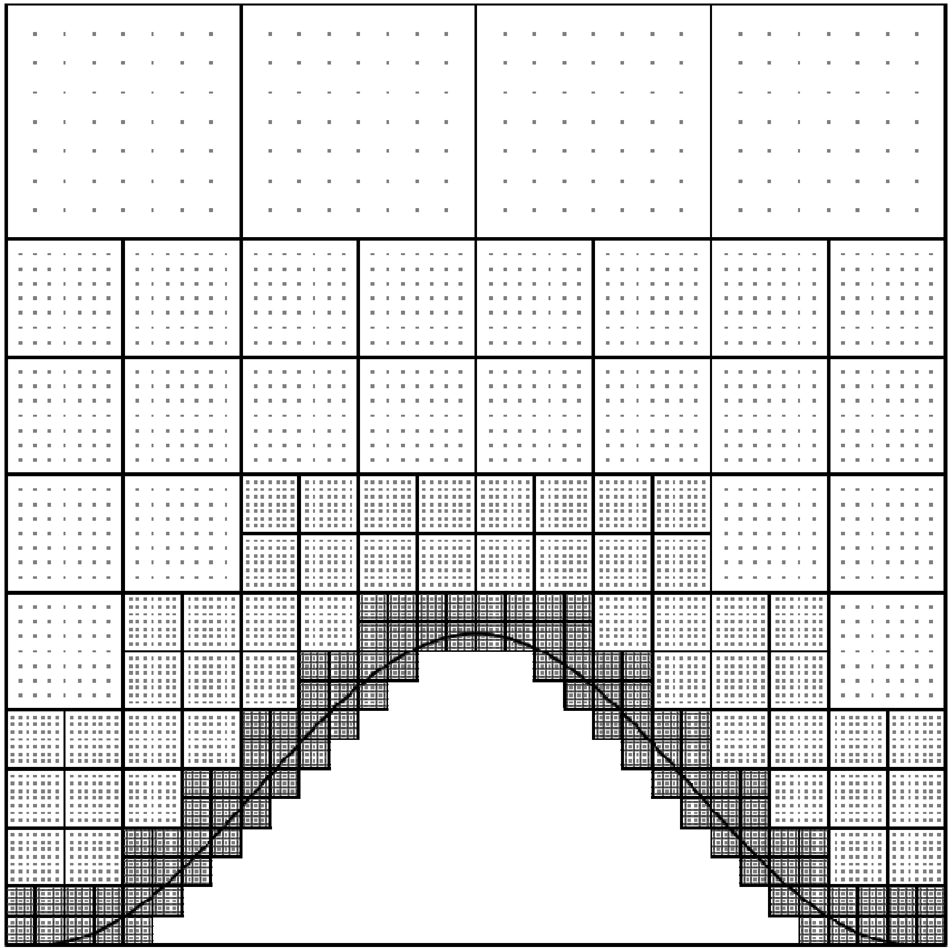
3. Adjust the size differences among blocks to guarantee a uniform 2:1 mesh resolution.

Mesh generation



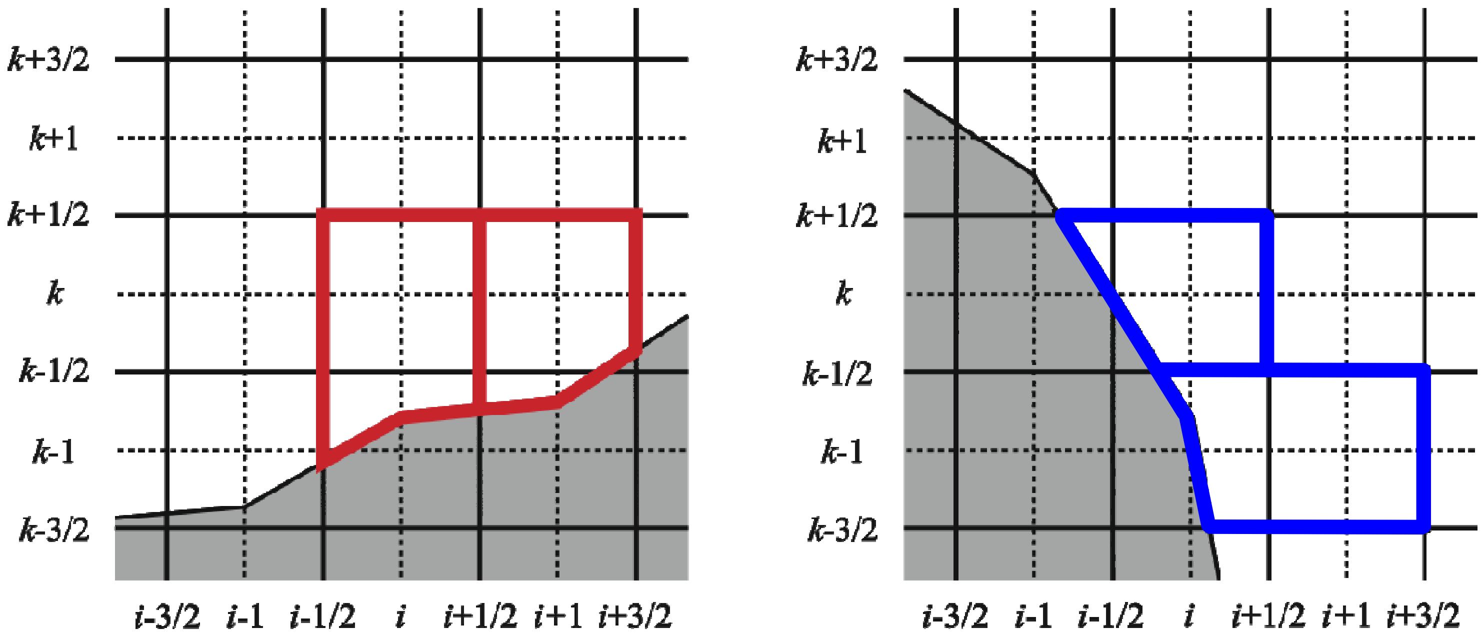
4. Remove the blocks that locate completely inside of the topography.

Mesh generation



5. Build a Cartesian mesh of equal spacing and equal number of cells in each cube.

Combining rules



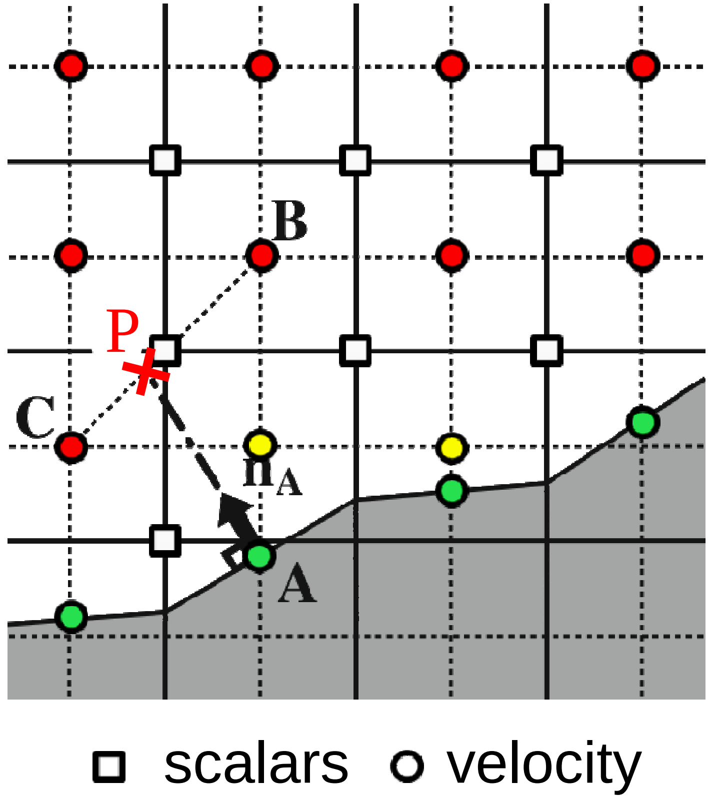
- The following cells are combined with a neighboring cell.
  - Cut cells whose volume is smaller than $\Delta x \Delta z / 2$.
  - Cut cells whose center is underground.

- The direction of cell combination is determined by mean slope angle:

$$\alpha_{i,k} = \tan^{-1}\left(\frac{h_{i+1/2} - h_{i-1/2}}{\Delta x}\right).$$

- If $|\alpha| \le \tan^{-1}(\Delta z / \Delta x)$, small cut cells are combined with each upper cell; else, they are combined with either each right or left cell in the fluid.

Direct evaluation of the velocity components on the boundary



With the non-slip boundary condition

- All velocity values on the boundary are set to zero.

With the free-slip boundary condition

- The component of the velocity that is tangential to the surface is preserved.

- For example, the velocity at the boundary point A is extrapolated as

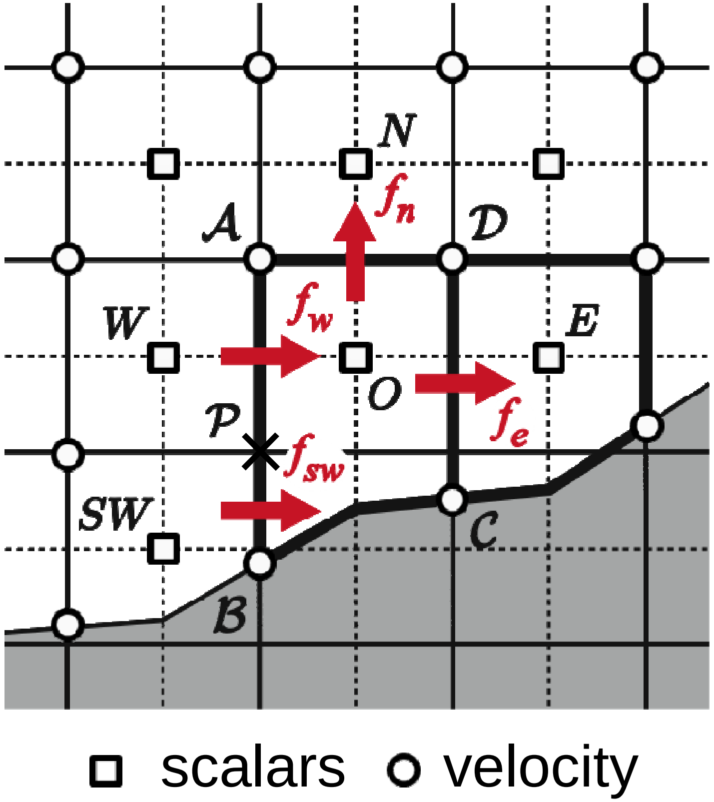
$$\mathbf{v}_A = \mathbf{v}_P - (\mathbf{v}_P \cdot \mathbf{n}_A)\mathbf{n}_A$$

where $\mathbf{n}_A$ indicates the unit normal direction to the surface, and $\mathbf{v}_P$ is the mean velocity above the normal direction calculated as

$$\mathbf{v}_P = \frac{\mathbf{v}_B \cdot \overline{\mathrm{PC}} + \mathbf{v}_C \cdot \overline{\mathrm{PB}}}{\overline{\mathrm{BC}}}$$

Flux calculations on combined cells

- The cell O enclosed by the area $ABCD$ exchanges flux with the cells, N, E, W, and SW.
- Zero normal flow is assumed at the faces on the immersed boundary.
- The normal velocity of these fluxes is obtained by linear interpolation between the values on the ends of each face.



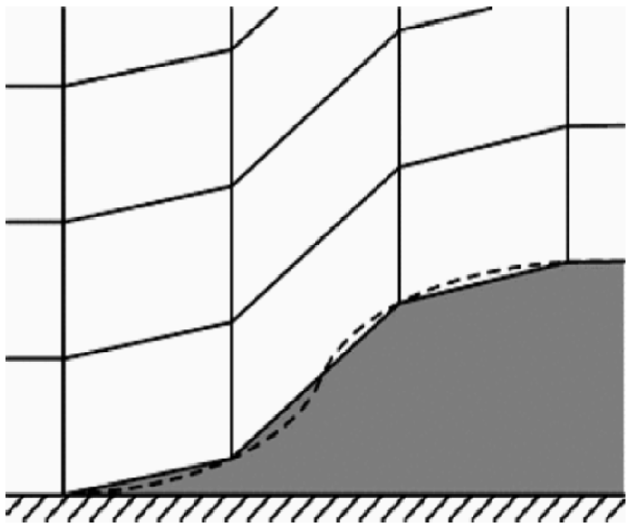
- The scalar quantity in these fluxes are approximated by the simple average of cell center values of the cells exchanging fluxes for computational simplification.

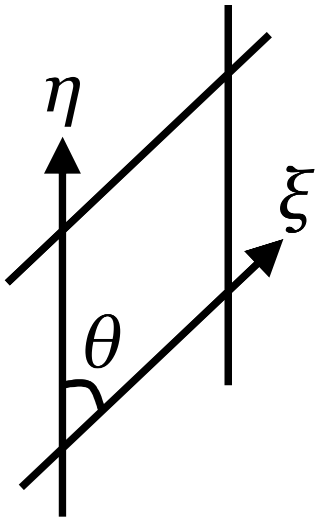
$$ex.)\quad \int_{AB} f\,dz \approx f_w \cdot \overline{AP} + f_{sw} \cdot \overline{PB} = \frac{u_A + u_P}{2} \cdot \frac{\phi_O + \phi_W}{2} \cdot \overline{AP} + \frac{u_P + u_B}{2} \cdot \frac{\phi_O + \phi_{SW}}{2} \cdot \overline{PB}$$

$$\text{where}\quad u_P = \frac{u_A \cdot \overline{PB} + u_B \cdot \overline{PA}}{\overline{AB}}.$$

Truncation errors in terrain-following models

- In terrain-following models, equations are discretized on a grid that conforms to the lower boundary.
- Horizontal gradient computation on a terrain-following grid essentially subject to truncation errors due to nonorthogonal coordinates.





- The steeper the slope, the larger the errors T (Thompson et al. 1985).

$$T \approx \frac{1}{2}\left\{ -x_{\xi\xi} f_{xx} + \left(y_{\eta\eta} f_{yy} - x_{\xi\xi} f_{xy} \right) \underline{\underline{\cot\theta}} \right\}$$

$T \propto \cot\theta$

- This errors will be serious in high-resolution simulations where steep slopes may appear.

Quasi-flux form fully compressible equations
(Satomura & Akiba, 2003)

$$\frac{\partial(\rho u)}{\partial t}+\frac{\partial(\rho uu)}{\partial x}+\frac{\partial(\rho uw)}{\partial z}=-\frac{\partial p'}{\partial x}+DIF(\rho u)$$

$$\frac{\partial(\rho w)}{\partial t}+\frac{\partial(\rho wu)}{\partial x}+\frac{\partial(\rho ww)}{\partial z}=-\frac{\partial p'}{\partial z}-\rho' g+DIF(\rho w)$$

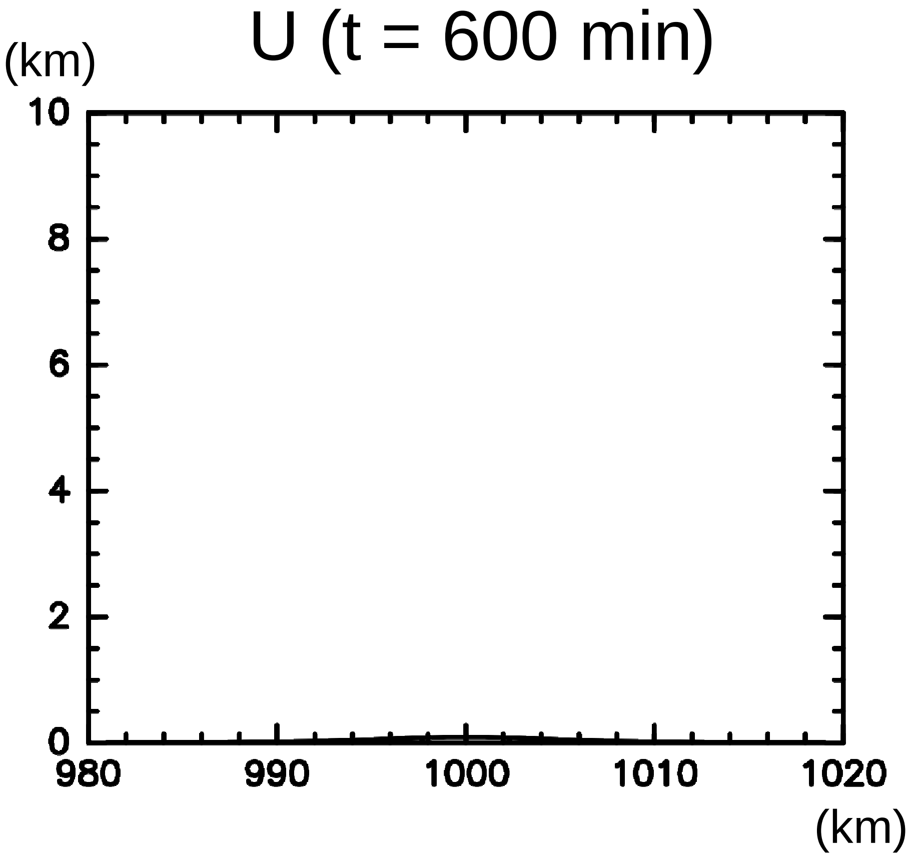
$$\frac{\partial p'}{\partial t}=-\frac{R_d\pi}{1-R_d/C_p}\left\{\frac{\partial(\rho\theta u)}{\partial x}+\frac{\partial(\rho\theta w)}{\partial z}+DIF(\rho\theta)\right\}$$

$$\frac{\partial\rho'}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho w)}{\partial z}=0$$

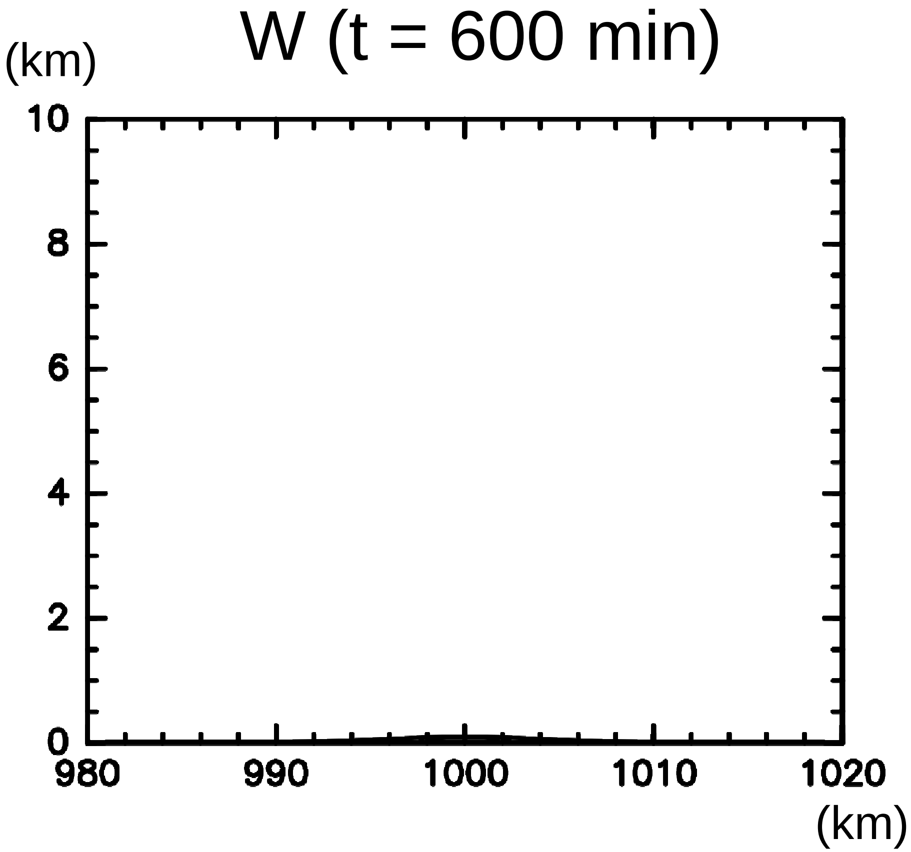
- This form does not suffer from the cancellation error caused by subtracting the hydrostatic variable ($\overline{p}$ or $\overline{\rho}$) from the nearly hydrostatic total variable (p or ρ).

static atmosphere test

(a = 5 km, h = 100 m, Δx = 1 km, Δz = 50 m)



(contour interval : 0.1 ms^{-1})

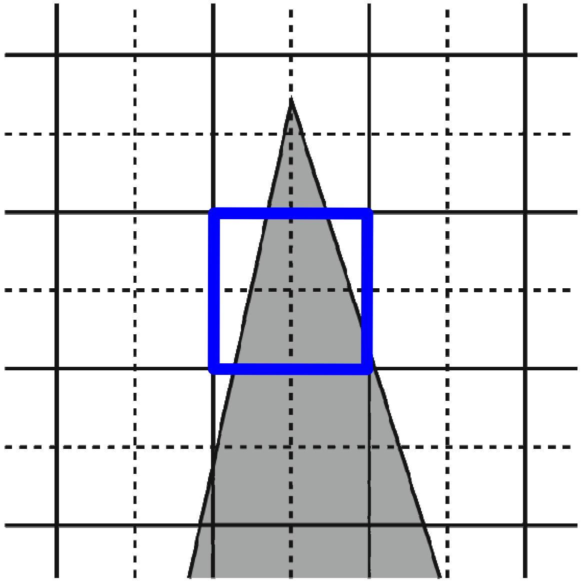


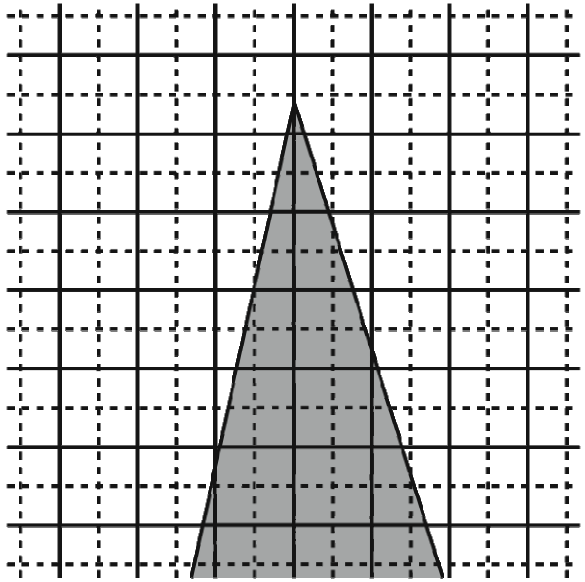
(contour interval : 0.05 ms^{-1})

The model fields remain still for 10 hours time integration.

Limits on topographic complexity

- ✕ A sharp-pointed mountain
- ✕ A steep v-shaped valley



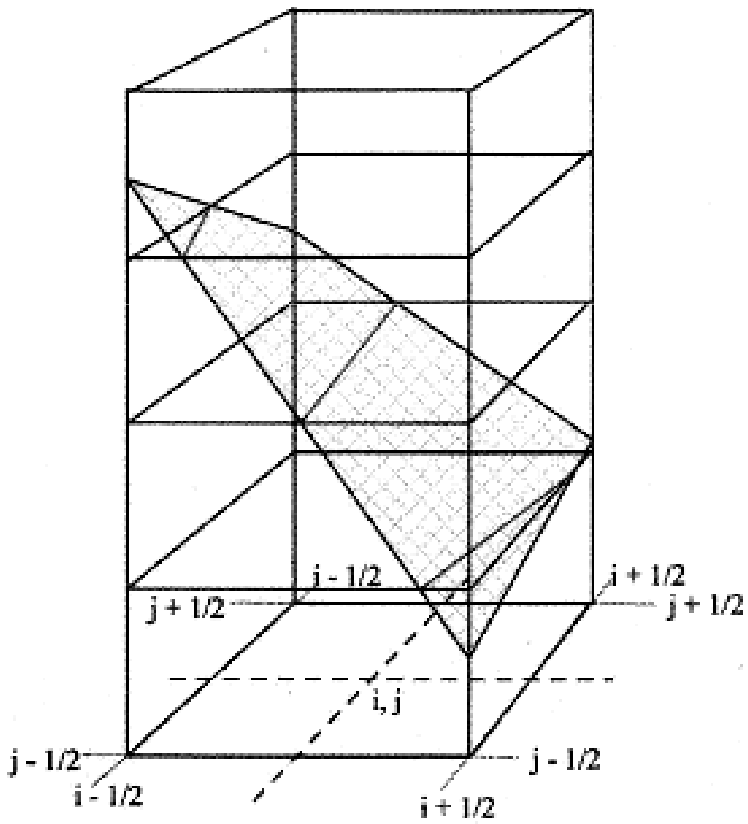


The fluid part of a cell is cut into two distinct pieces

Increase the resolution while keeping the shape of topography fixed

- In this study, we are not concerned with long thin topography such as a sharp-pointed mountain or a steep v-shaped valley where topography varies substantially on the grid scale.
- A way to handle long thin topography is to increase the resolution while keeping the shape of topography fixed.

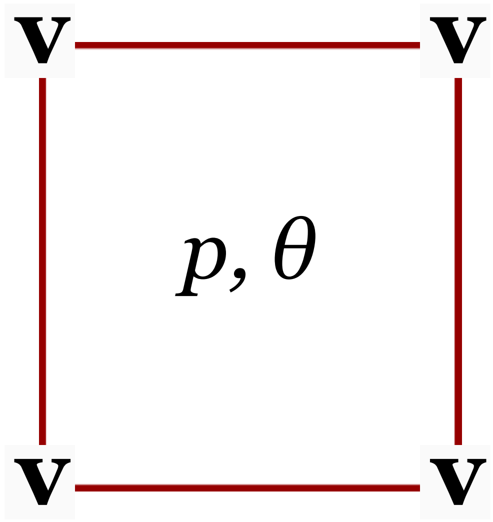
Three-dimensional modeling



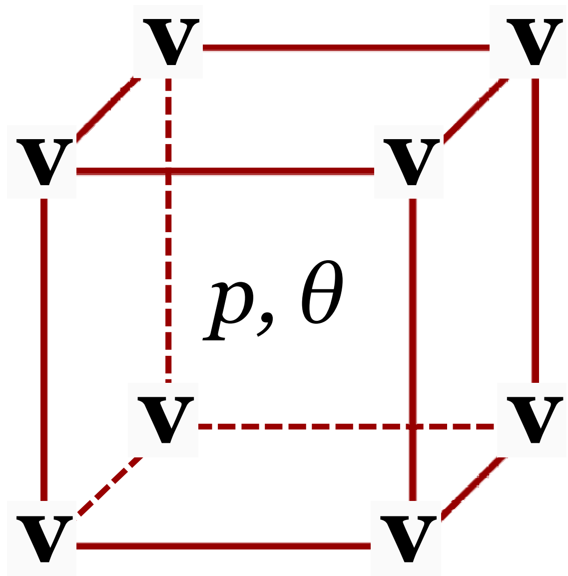
- The extension of our method to three dimensions should be straightforward.

- For establishing of a three-dimensional orographic surface, we will use a series of two-dimensional bilinear surface (Steppeler et al. 2006; Lock 2008).

A unique arrangement of variables



$$\mathbf{v} = (u, w)$$



$$\mathbf{v} = (u, v, w)$$

Contents

One of the greatest concerns in cut cell modeling is strategies for the small cell problem.

- Cell-merging
- Implicit
- *h*-box
- Wave-propagation
- Thin-wall
- Local reflection
- Flux redistribution

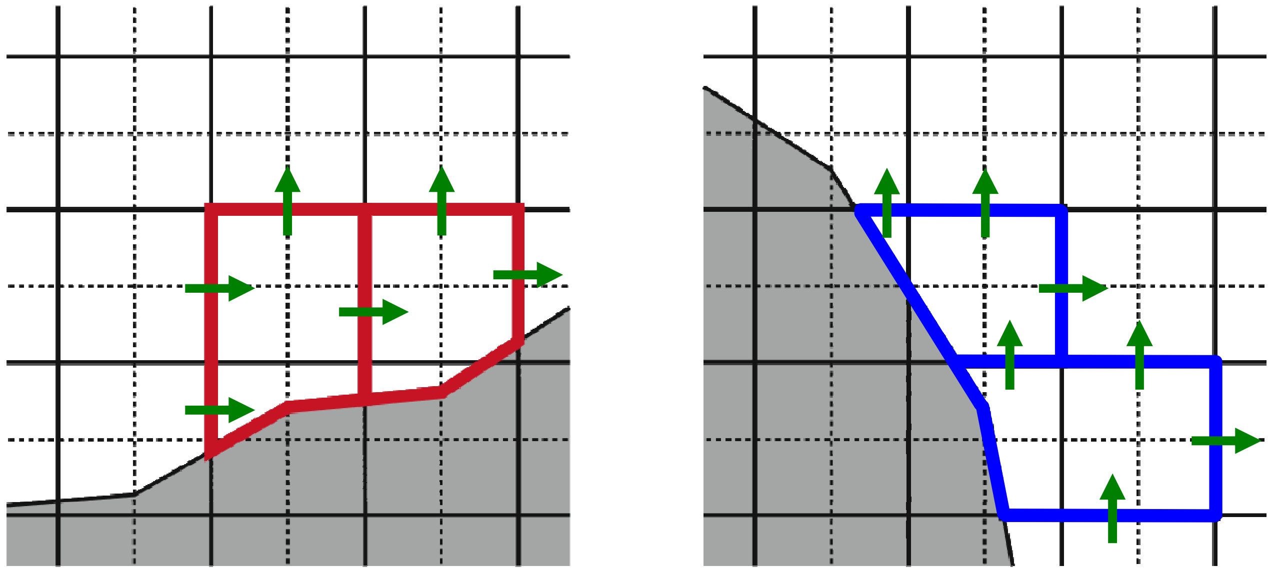
⋮

One of the greatest concerns in cut cell modeling is strategies for the small cell problem.

- Cell-merging
- Implicit
- *h*-box
- Wave-propagation
- Thin-wall
- Local reflection
- Flux redistribution
- ⋮

Cell-merging approach
(e.g., Ye et al. 1999; Yamazaki & Satomura 2008,2010)

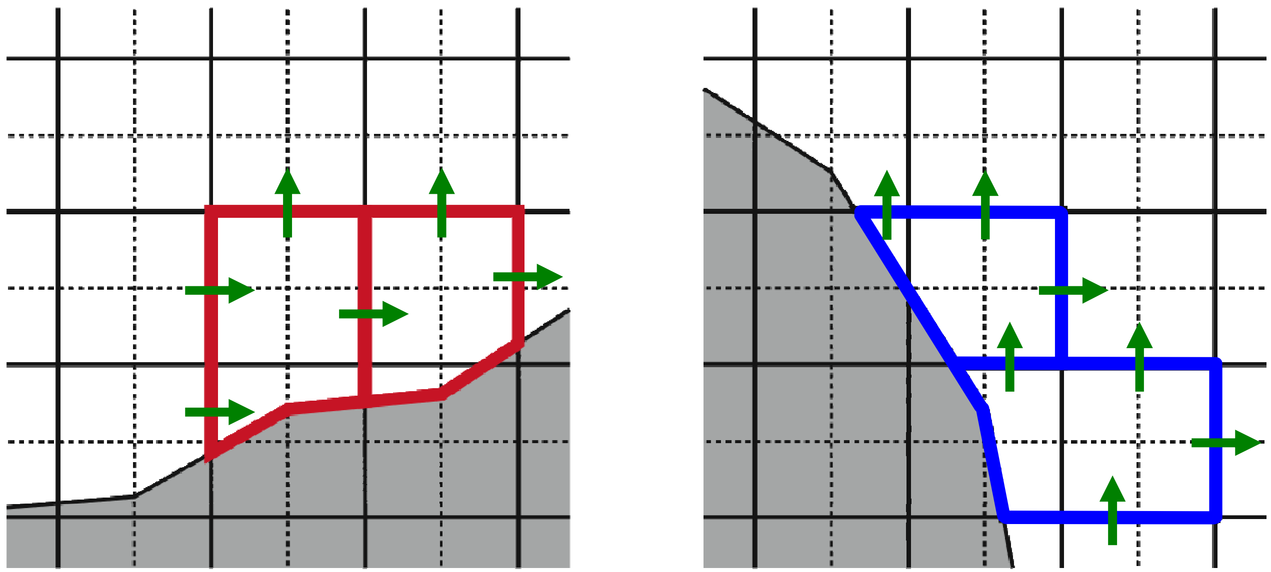
Small cut cells are merged with neighboring cells either vertically or horizontally.



- Naturally avoiding severe CFL restriction maintaining the rigid evaluation of cell volumes.
- Cell-merging generally entails a considerable increase in complexity.

Cell-merging approach
(e.g., Ye et al. 1999; Yamazaki & Satomura 2008,2010)

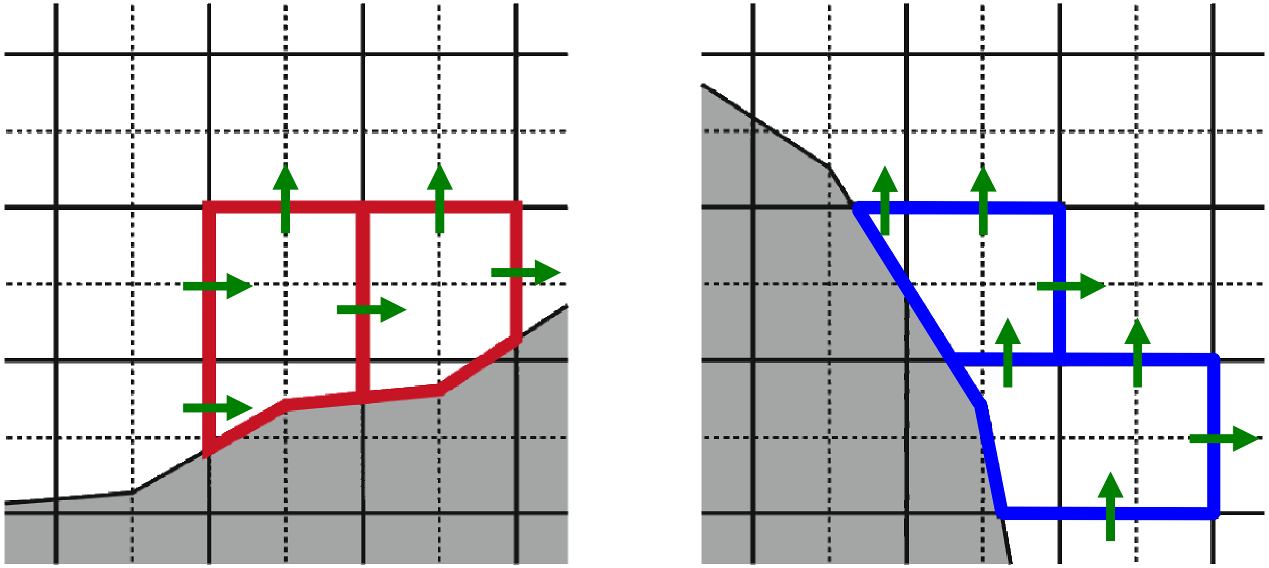
Small cut cells are merged with neighboring cells either vertically or horizontally.



- Naturally avoiding severe CFL restriction maintaining the rigid evaluation of cell volumes.
- Cell-merging generally entails a considerable increase in complexity.

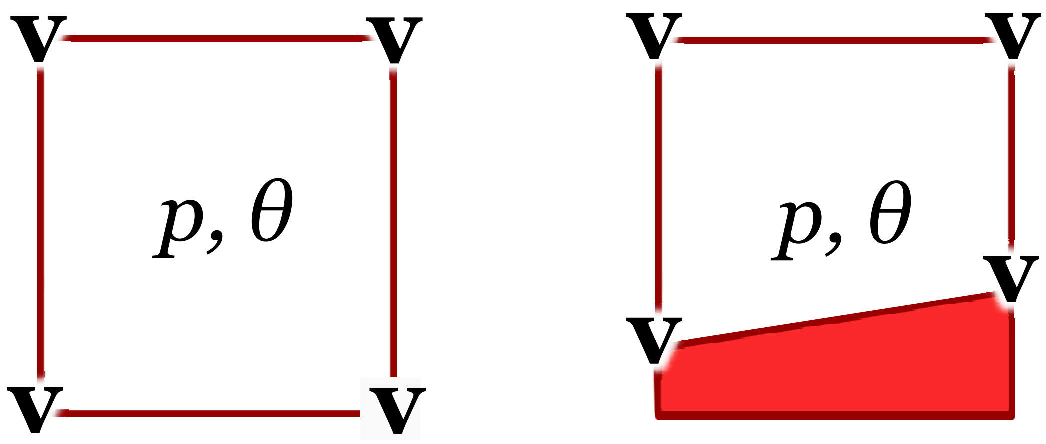
Cell-merging approach
(e.g., Ye et al. 1999; Yamazaki & Satomura 2008,2010)

Small cut cells are merged with neighboring cells either vertically or horizontally.



- Naturally avoiding severe CFL restriction maintaining the rigid evaluation of cell volumes.
- Cell-merging generally entails a considerable increase in complexity.

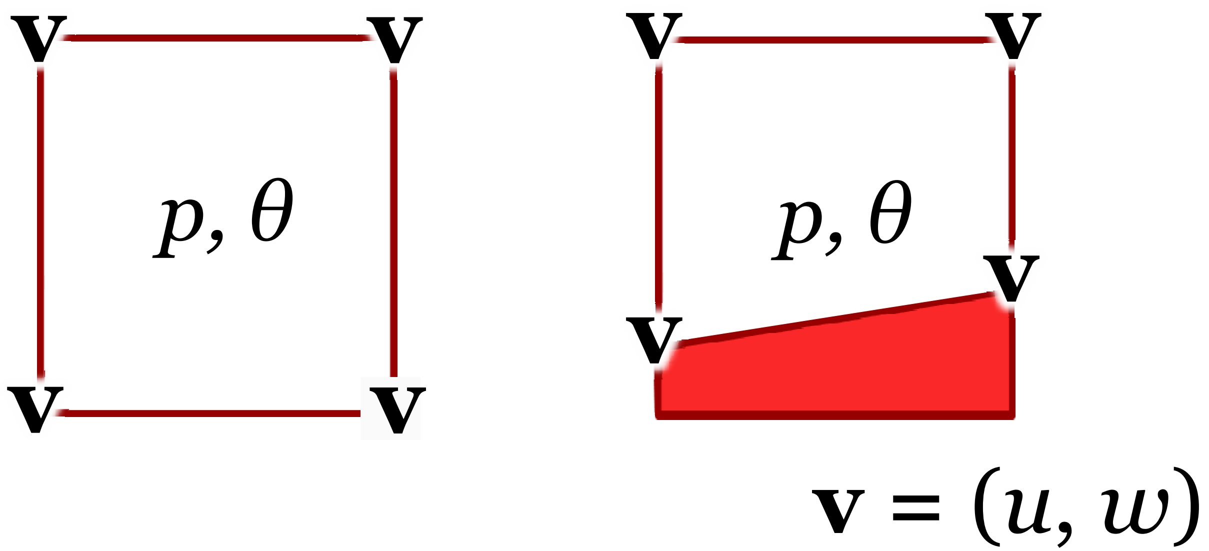
We propose a simple cut cell method with cell-merging using a novel arrangement of variables.



$$\mathbf{v} = (u, w)$$

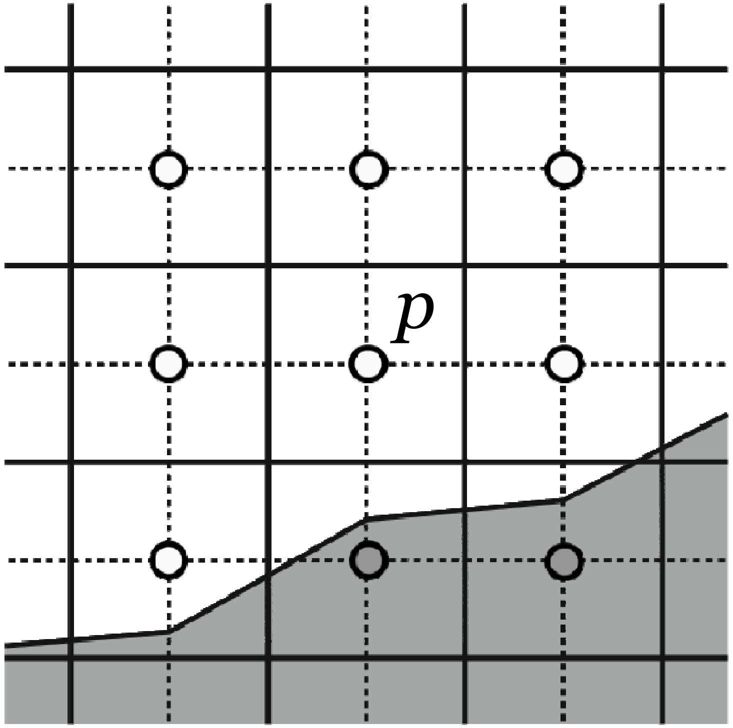
- Velocity components are co-located and arranged on the corners of the cells.
- This arrangement enables direct evaluation of the velocity on the boundary, thereby simplifying the calculation near the boundary.

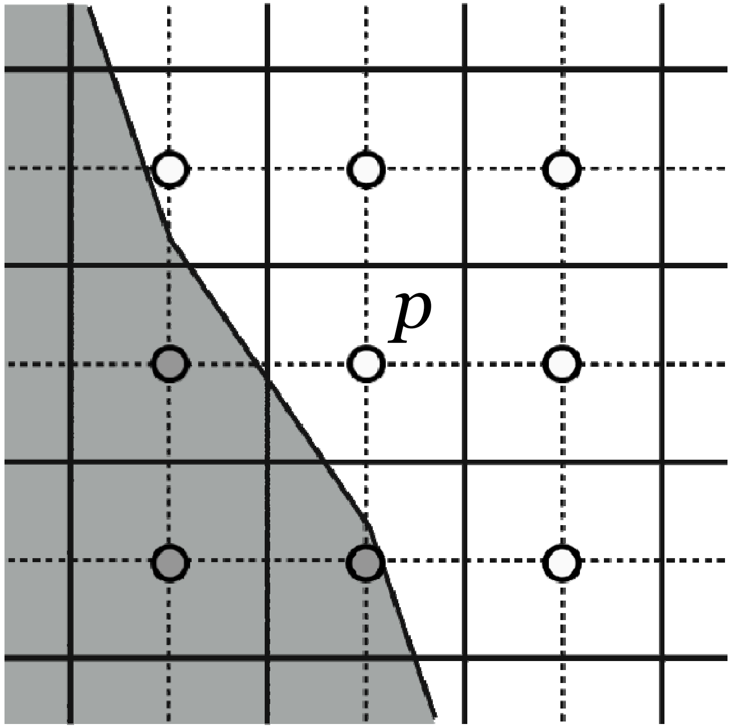
We propose a simple cut cell method with cell-merging using a novel arrangement of variables.



- Velocity components are co-located and arranged on the corners of the cells.
- This arrangement enables direct evaluation of the velocity on the boundary, thereby simplifying the calculation near the boundary.

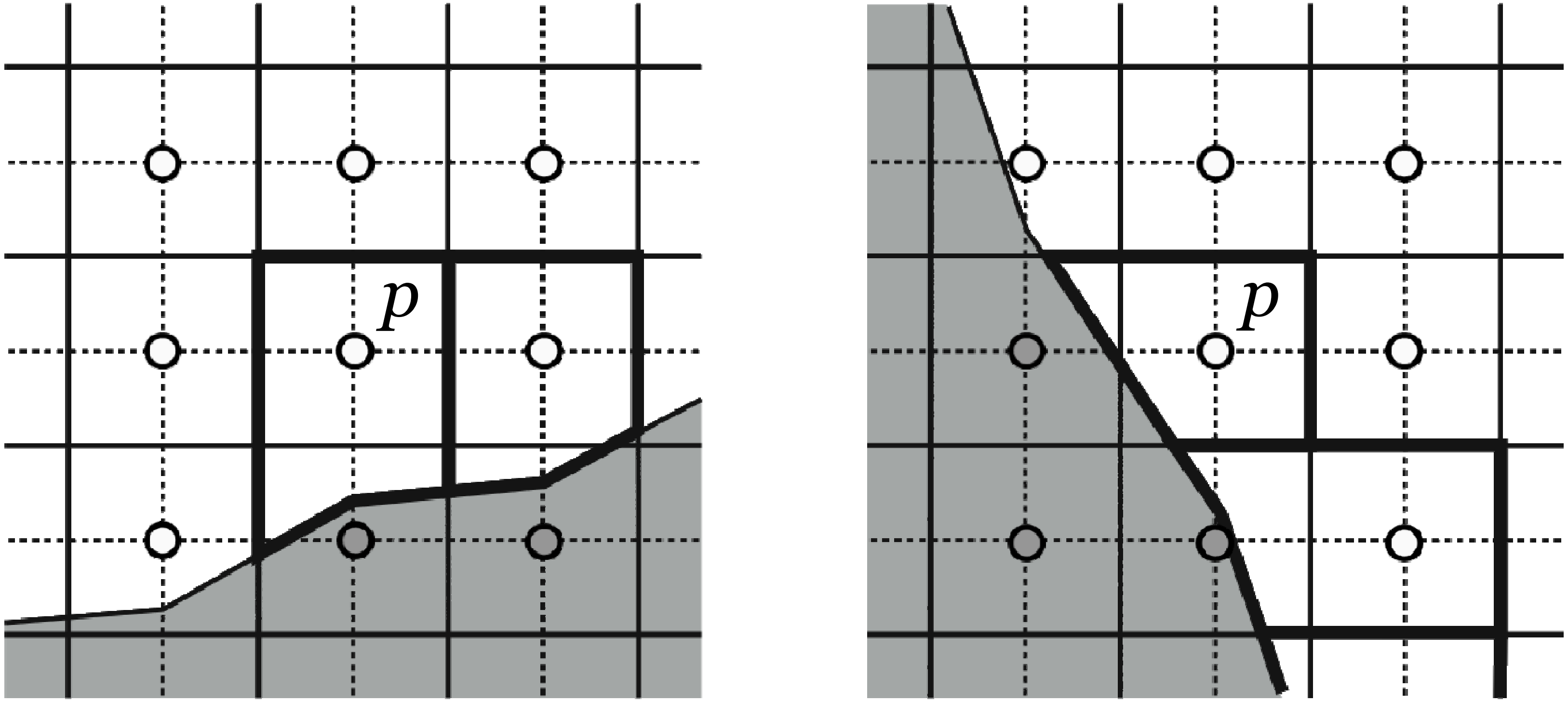
Our cut cell method
(Yamazaki and Satomura 2010)





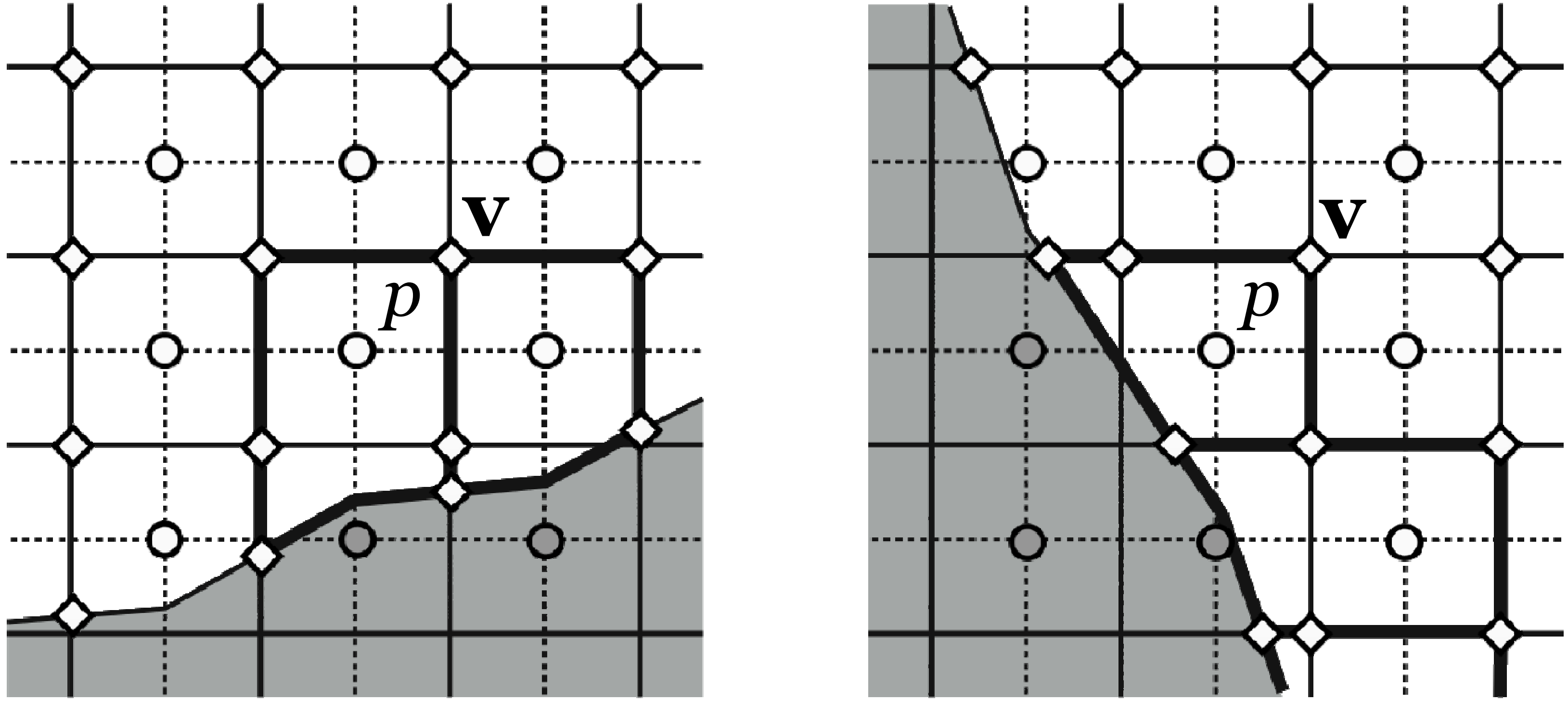
- Scalar variables are arranged on the cell centers.

Our cut cell method
(Yamazaki and Satomura 2010)



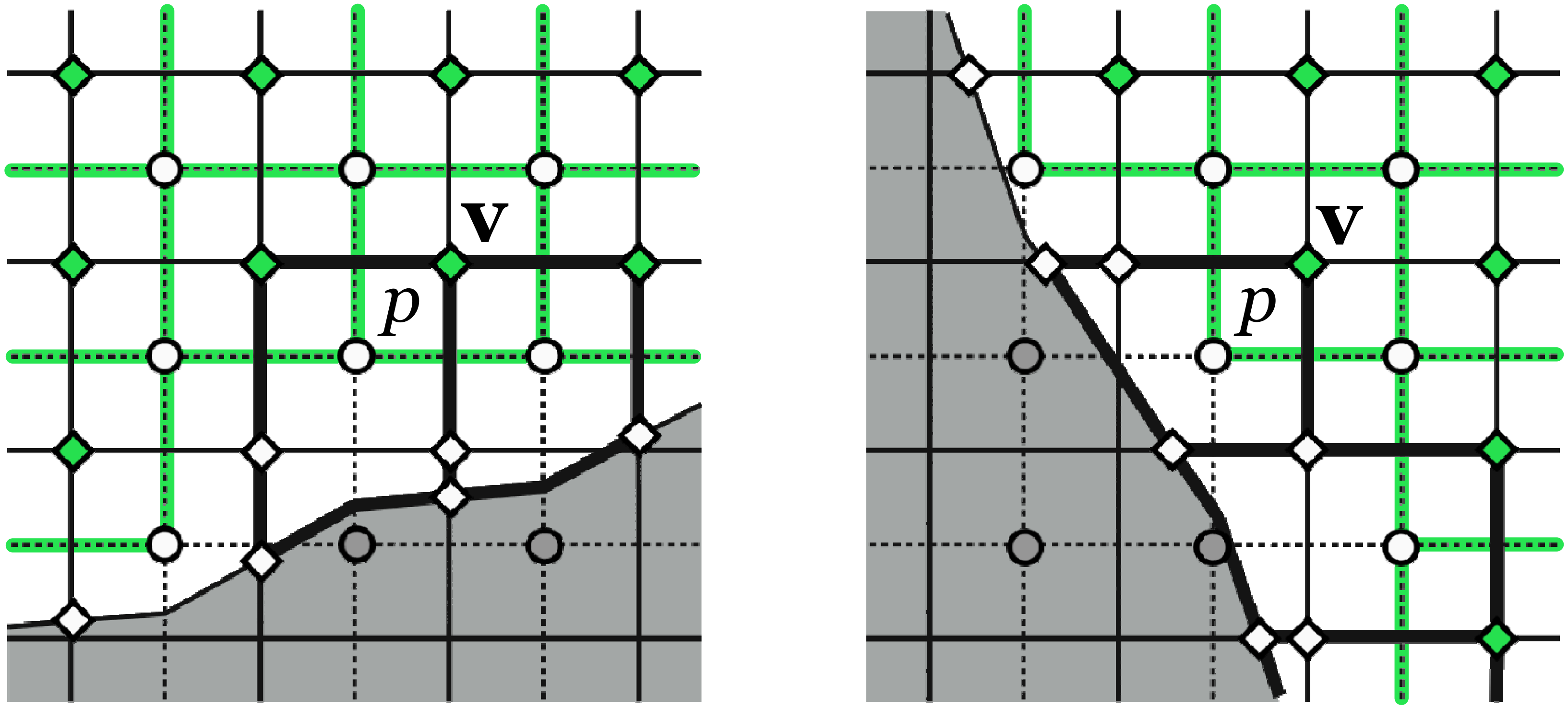
- Scalar variables are arranged on the cell centers.
- Cut cells whose center is underground are merged with a neighboring cell.

Our cut cell method
(Yamazaki and Satomura 2010)



- Scalar variables are arranged on the cell centers.
- Cut cells whose center is underground are merged with a neighboring cell.
- Velocity points are arranged on the cell corners.

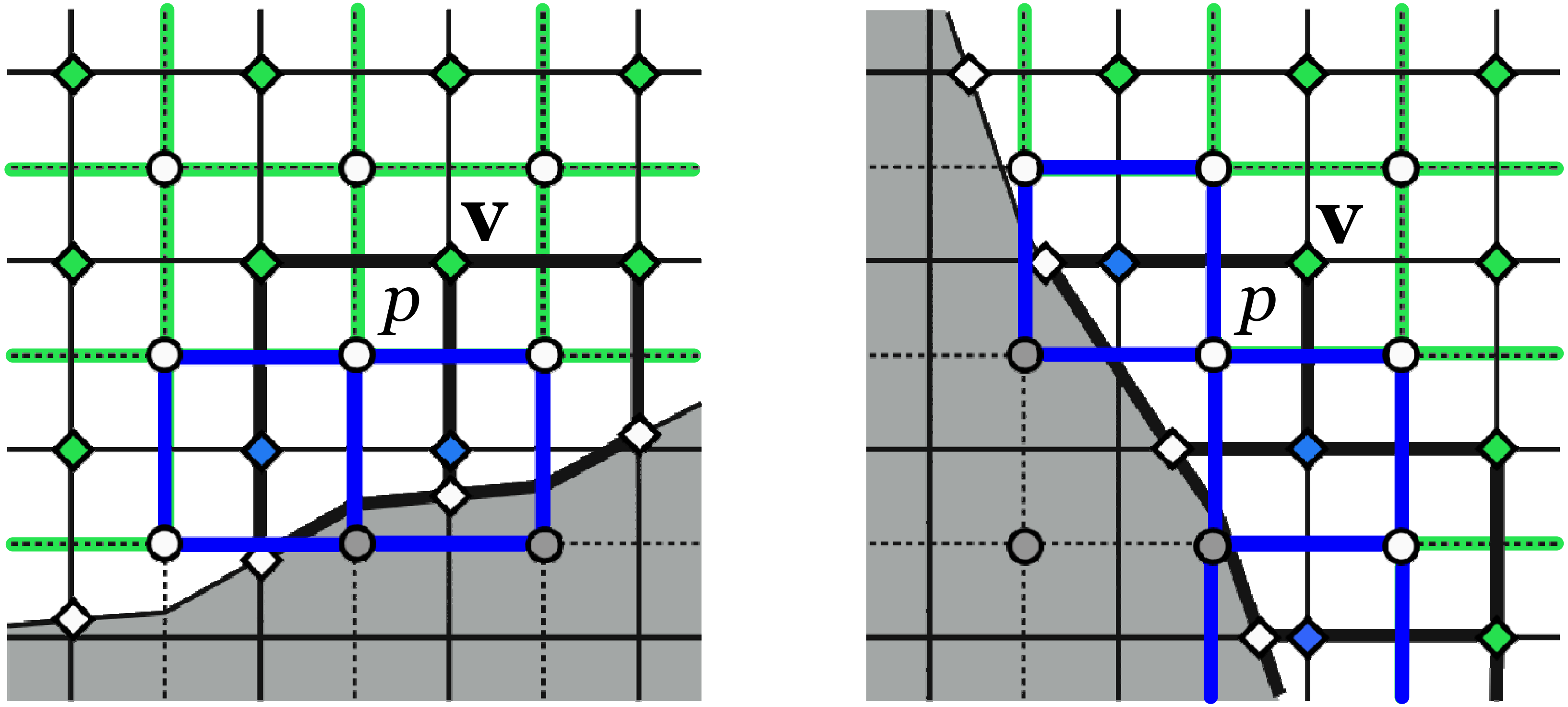
Our cut cell method
(Yamazaki and Satomura 2010)



- Velocity at cells which retain rectangular shapes is calculated in a standard way.

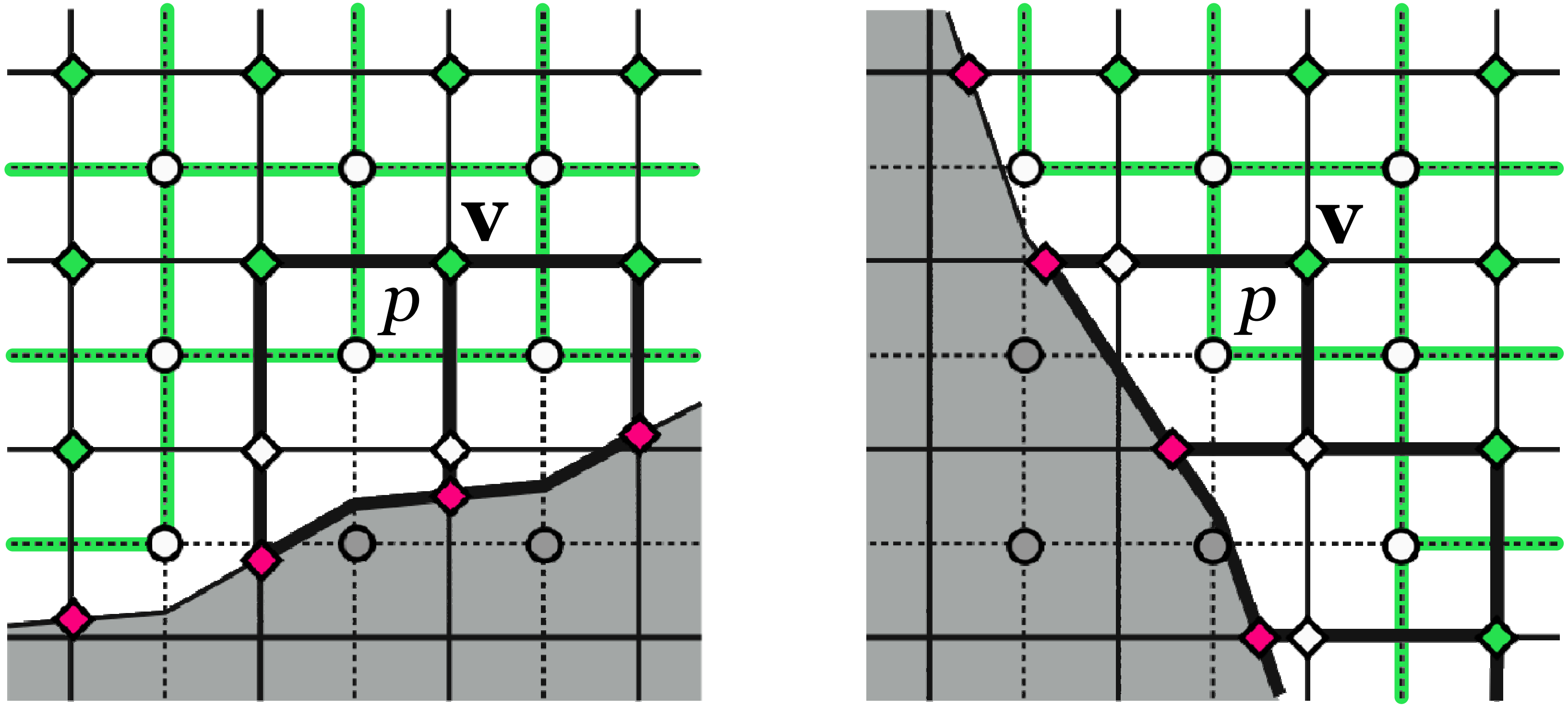
Our cut cell method

(Yamazaki and Satomura 2010)



- Velocity at cells which retain rectangular shapes is calculated in a standard way.
- Generally speaking, the calculation of velocity on remaining points near the boundary is complicated.

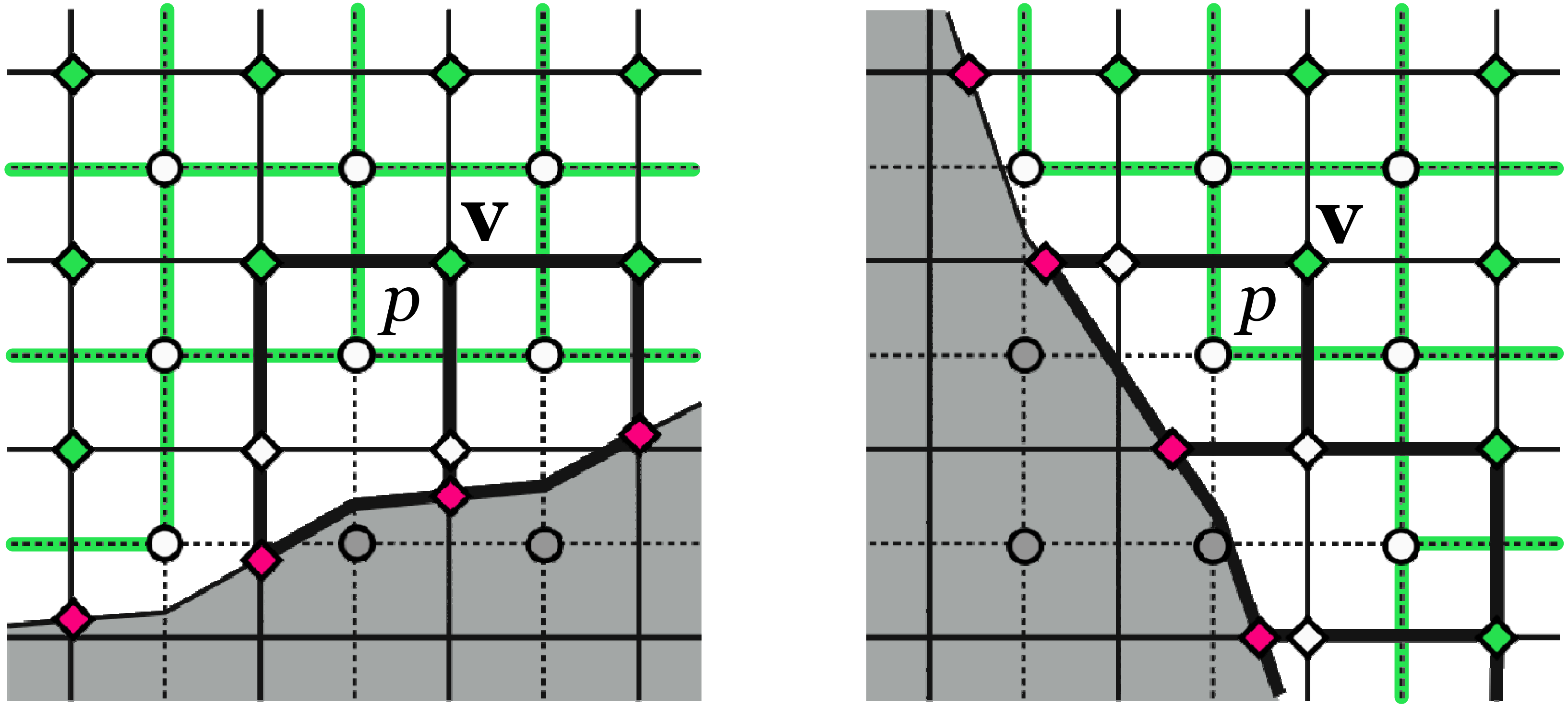
Our cut cell method
(Yamazaki and Satomura 2010)



- Velocity on the boundary is diagnostically calculated using the boundary conditions.

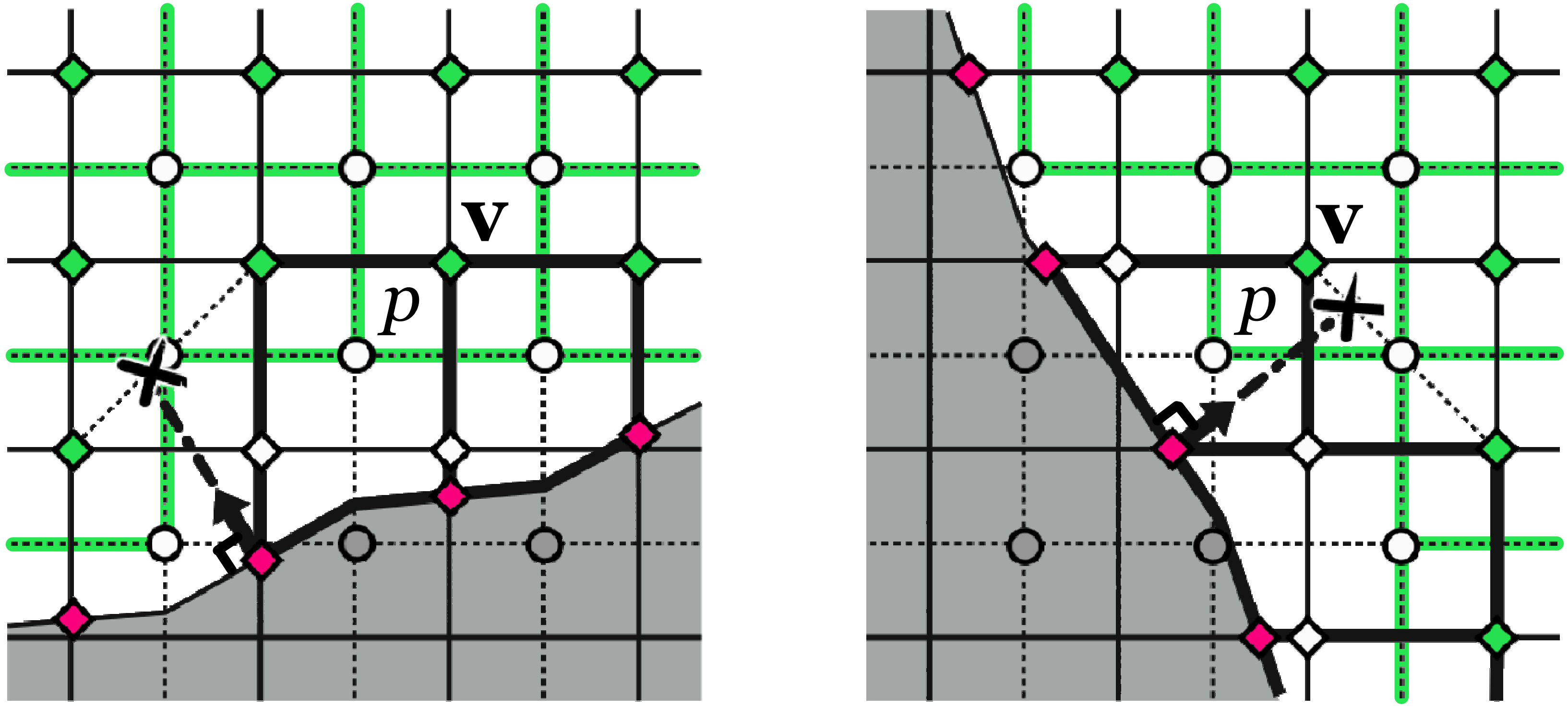
Our cut cell method

(Yamazaki and Satomura 2010)



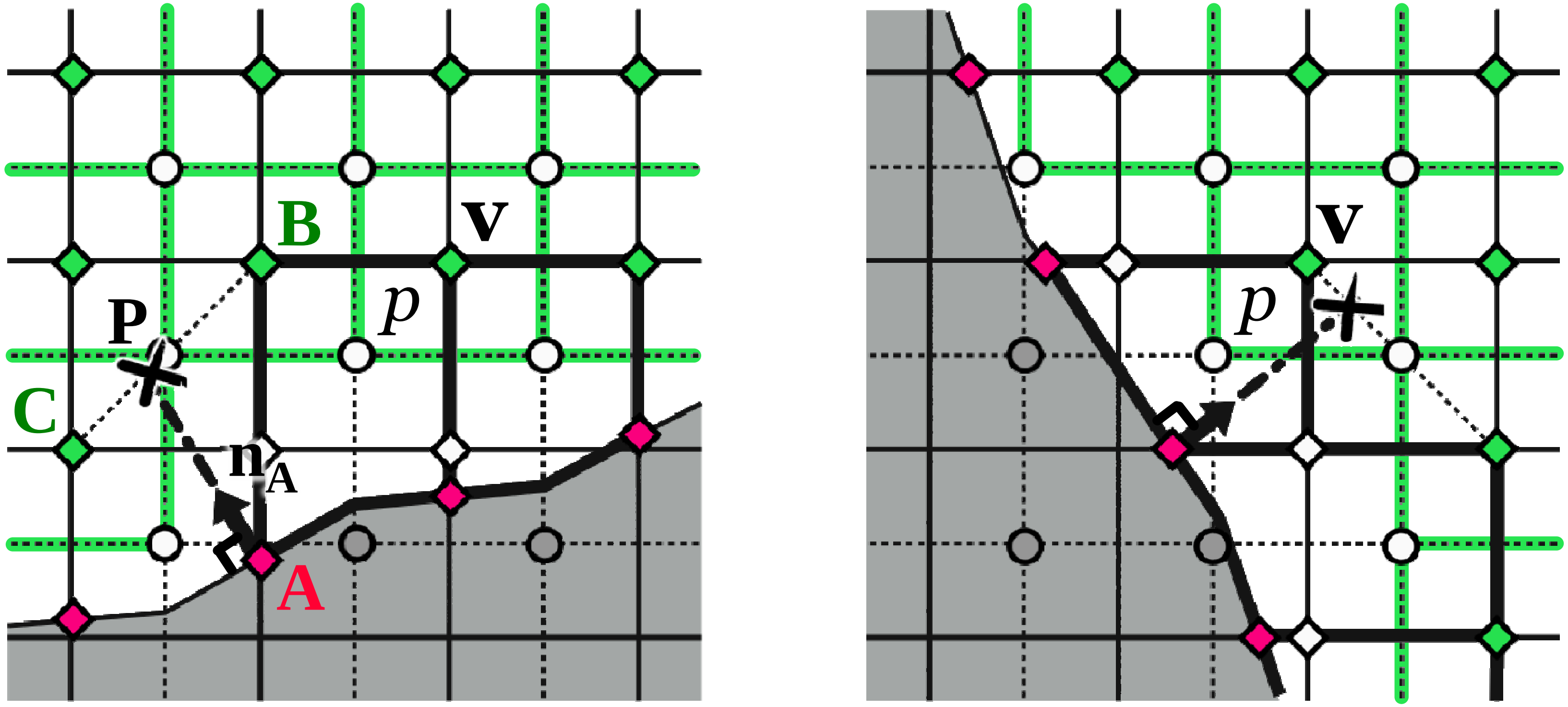
- Velocity on the boundary is diagnostically calculated using the boundary conditions.
  - non-slip – all boundary velocity is set to zero.

Our cut cell method
(Yamazaki and Satomura 2010)



- Velocity on the boundary is diagnostically calculated using the boundary conditions.
  - non-slip – all boundary velocity is set to zero.
  - free-slip – The component of the velocity that is tangential to the surface is preserved.

Our cut cell method
(Yamazaki and Satomura 2010)



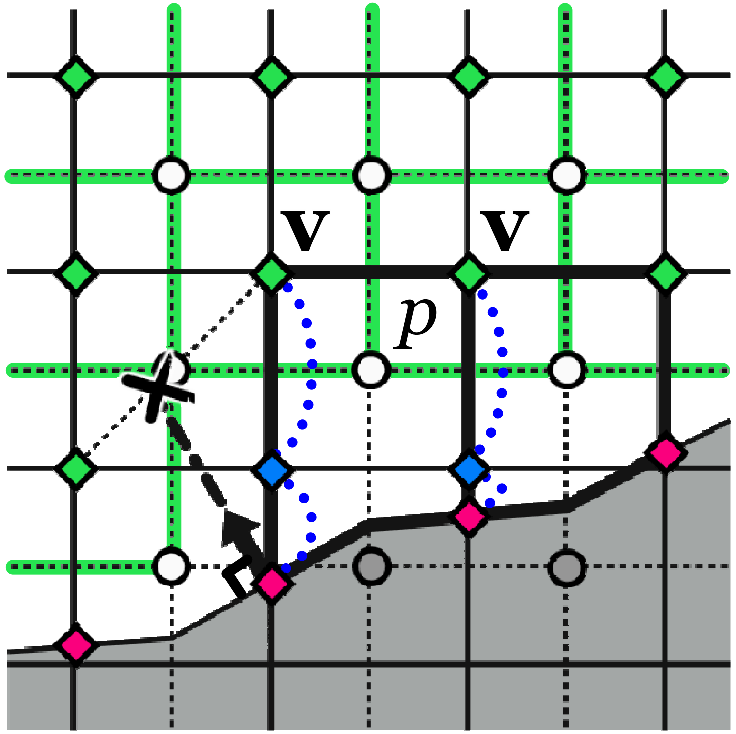
For example, $\mathbf{v}_A$ is calculated as

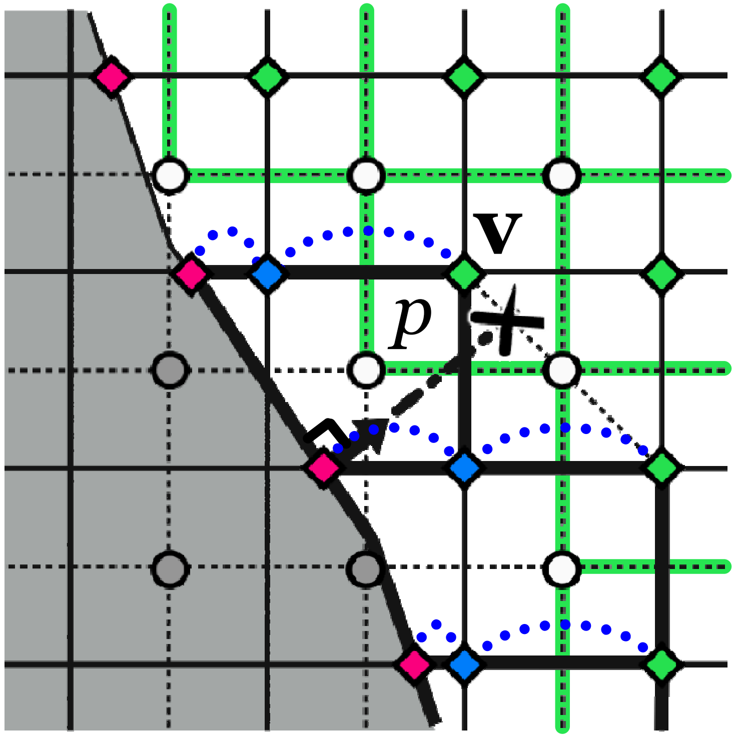
$$\mathbf{v}_A = \mathbf{v}_P - (\mathbf{v}_P \cdot \mathbf{n}_A)\mathbf{n}_A$$

where $\mathbf{v}_P$ is a velocity above the normal direction:

$$\mathbf{v}_P = (\mathbf{v}_B \cdot \overline{\mathrm{PC}} + \mathbf{v}_C \cdot \overline{\mathrm{PB}}) / \overline{\mathrm{BC}}$$

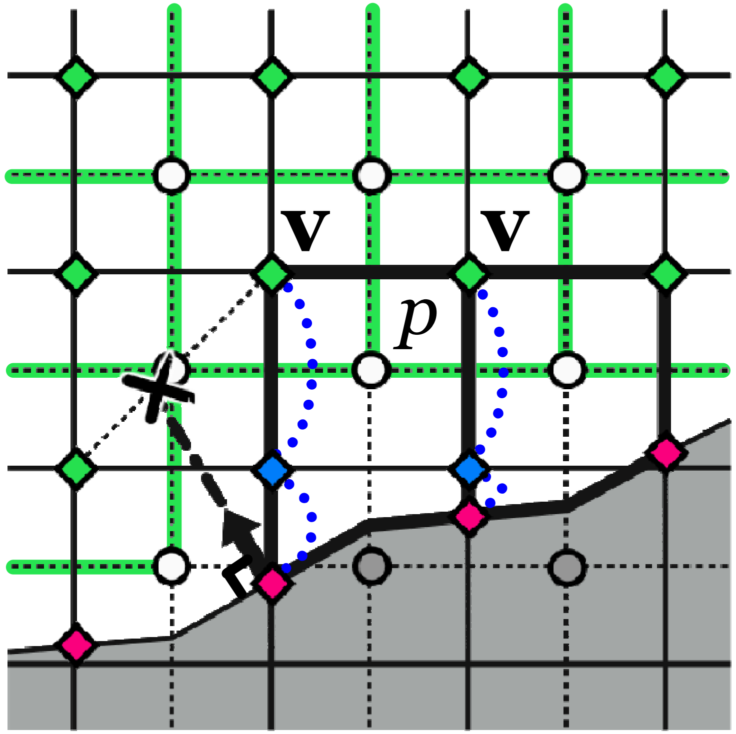
Our cut cell method
(Yamazaki and Satomura 2010)

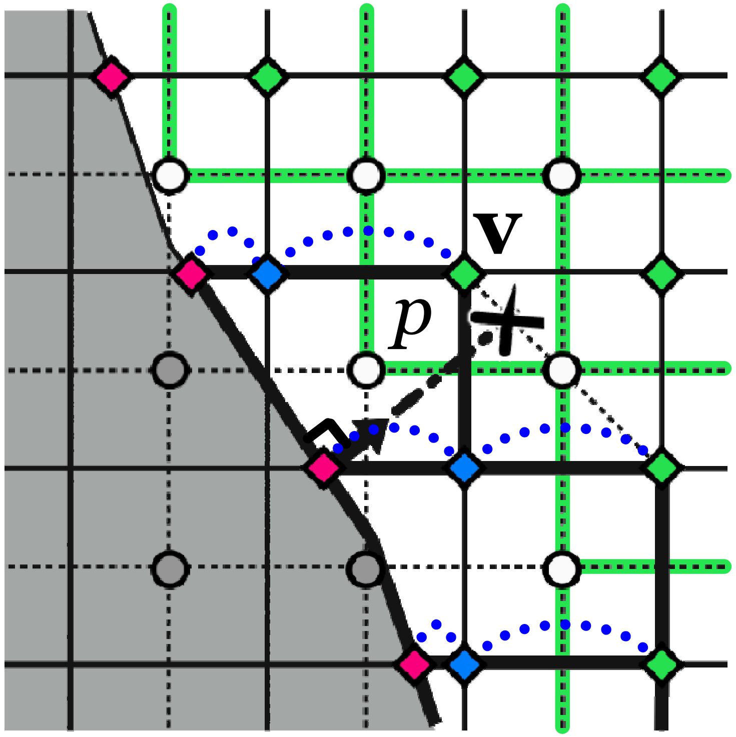




- The velocities on the remaining points are all on the merged faces, and are calculated by assuming a linear distribution of velocity on each merged face.

Our cut cell method
(Yamazaki and Satomura 2010)

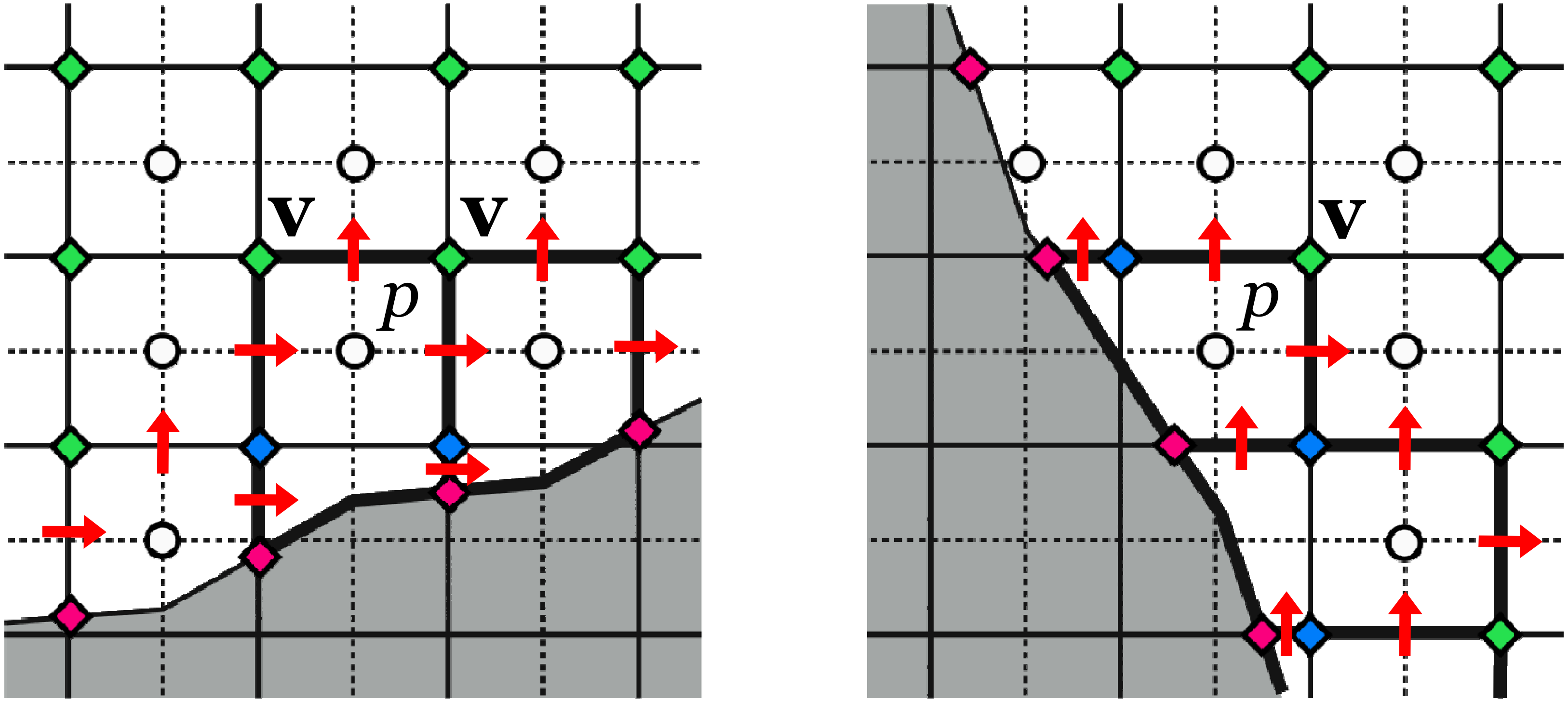




✦ With this 3-step calculation of velocity,

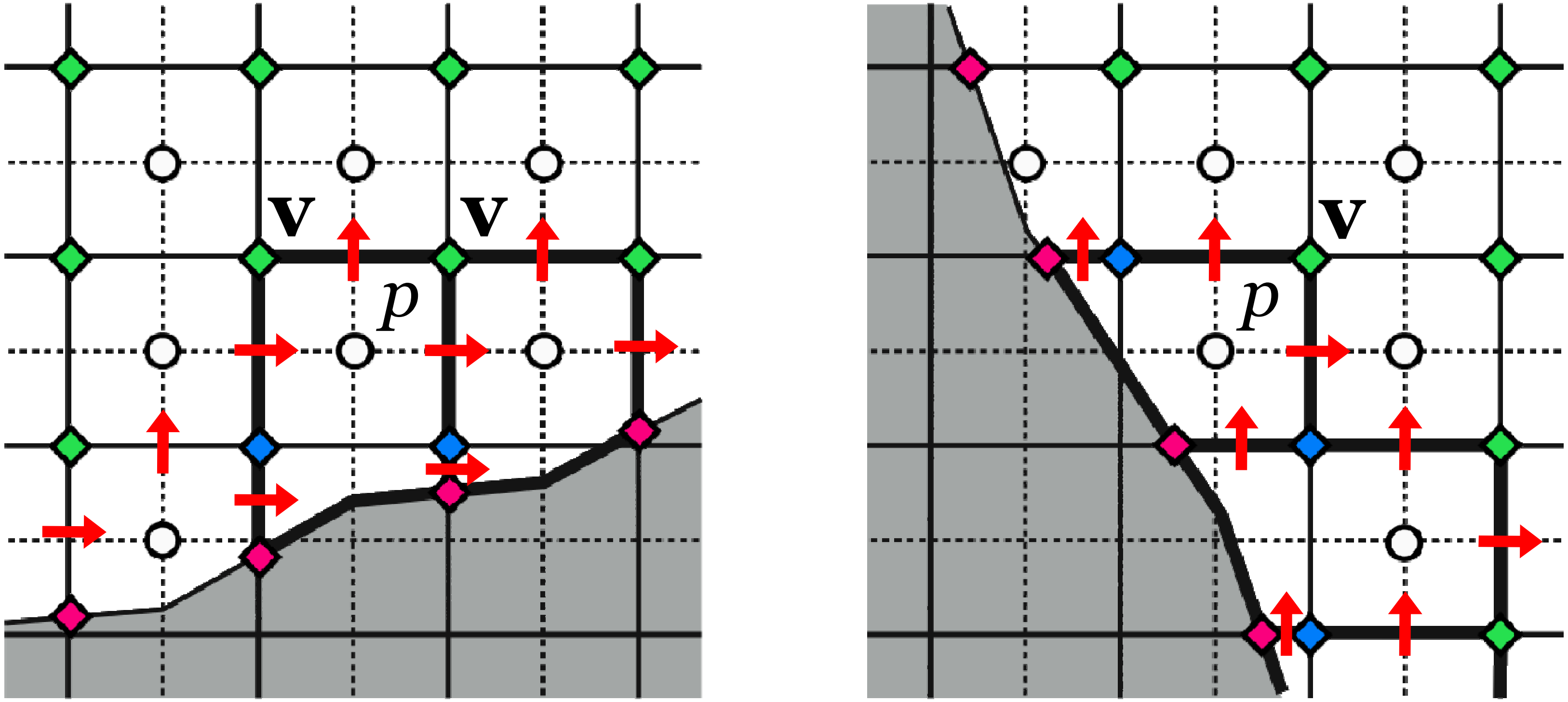
- all velocities are obtained without complex estimation of underground/surface pressure.
- no merging process for velocity cells is needed.

Our cut cell method
(Yamazaki and Satomura 2010)



- Velocity at face centers is obtained simply by two-point average of velocities on the ends of each face.
- Scalar quantity at face centers is calculated by first-order interpolation of the cell-center values.

Our cut cell method
(Yamazaki and Satomura 2010)



- These flux calculations
  - guarantee the mass-conservation.
  - maintain global second-order convergence.

Contents

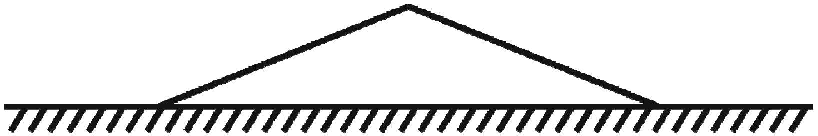
We developed a new nonhydrostatic atmospheric cut cell model code-named "Sayaca-2D".

■ Dynamics	
Dimension	2-D
Governing equations	Fully compressible (Satomura & Akiba 2003)
Variable arrangement	Semi-staggered (Yamazaki & Satomura 2010)
Spatial discretization	FVM (advection) / FDM (others)
Time integration	Leap frog with Asselin filter (All explicit)
Topography	Cut cell method with cell-merging
Numerical smoothing	4th order artificial diffusion
■ Physics	
Subgrid turbulence parameterization	1.5 order（Klemp and Wilhelmson 1974）

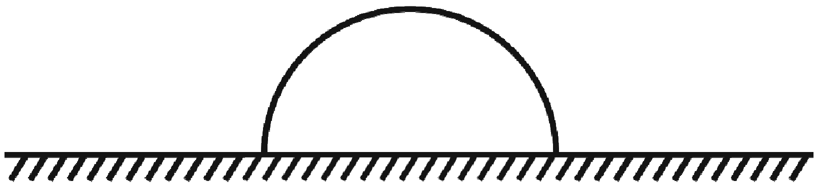
Numerical simulations of flow over slopes of various angles are performed using Sayaca-2D.

(1) a bell-shaped mountain

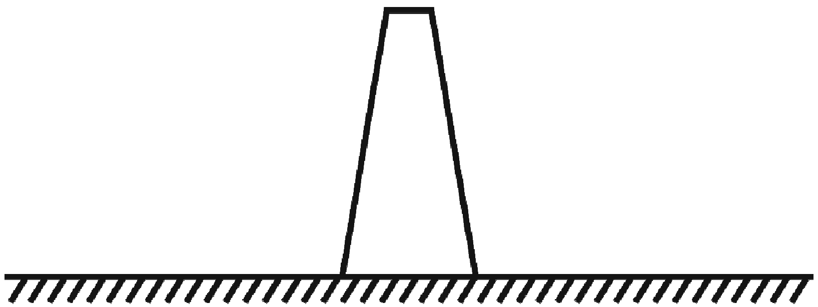
(2) a pyramidal mountain



(3) a semicircular mountain

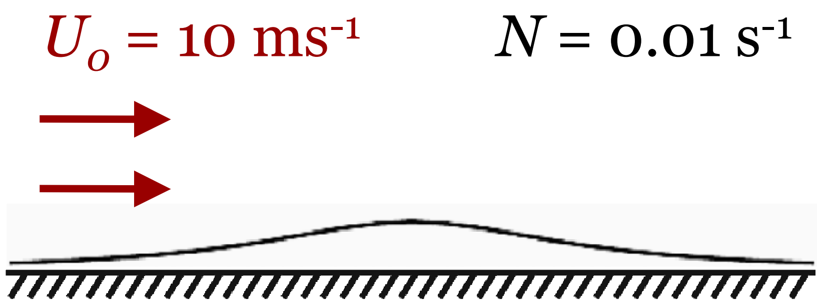


(4) a cliff



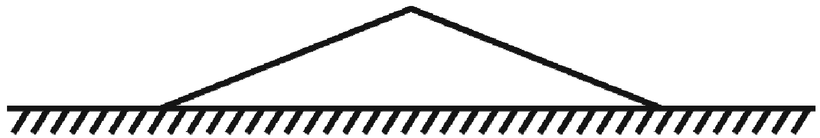
Numerical simulations of flow over slopes of various angles are performed using Sayaca-2D.

(1) a bell-shaped mountain

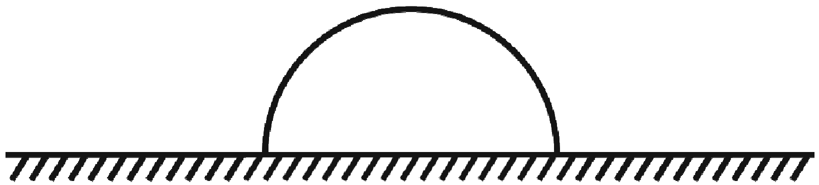


max. slope angle : 0.72°

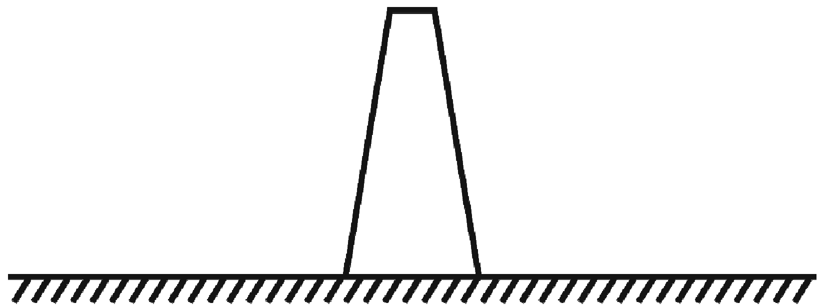
(2) a pyramidal mountain



(3) a semicircular mountain

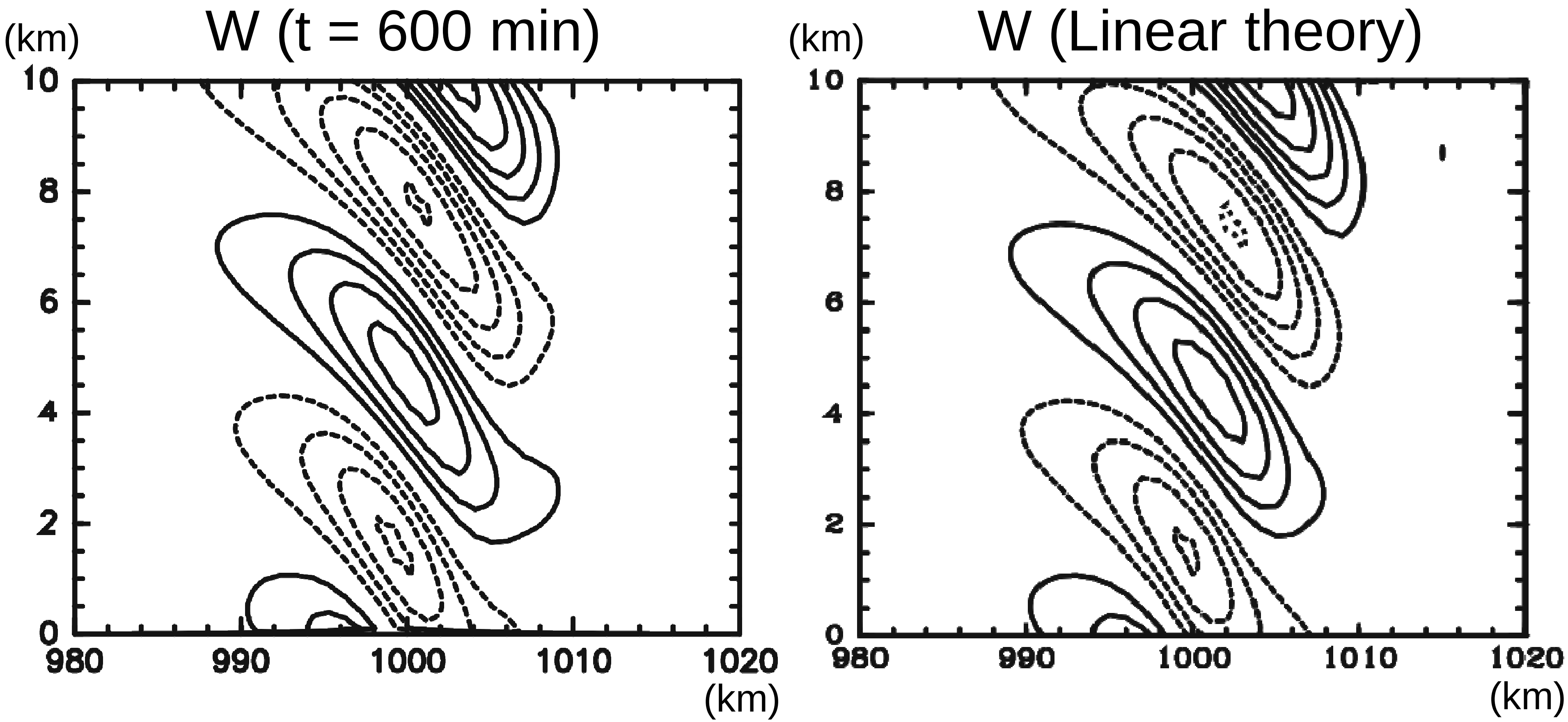


(4) a cliff



Flow over a bell-shaped mountain

(a = 5 km, h = 100 m, Δx = 1 km, Δz = 50 m)

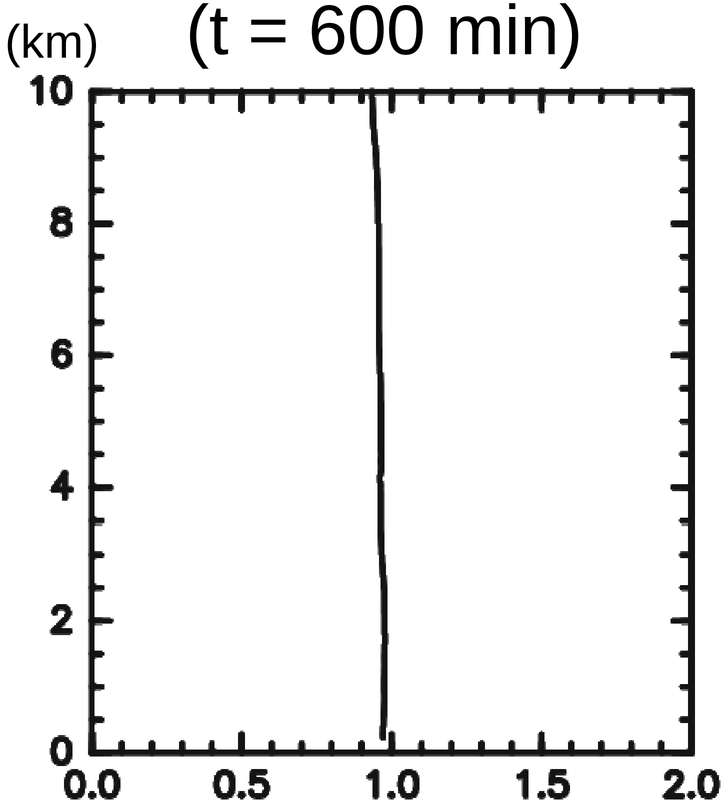


(contour interval : 0.05 ms^{-1})

Flow over a bell-shaped mountain

(a = 5 km, h = 100 m, Δx = 1 km, Δz = 50 m)

Normalized Flux
(t = 600 min)



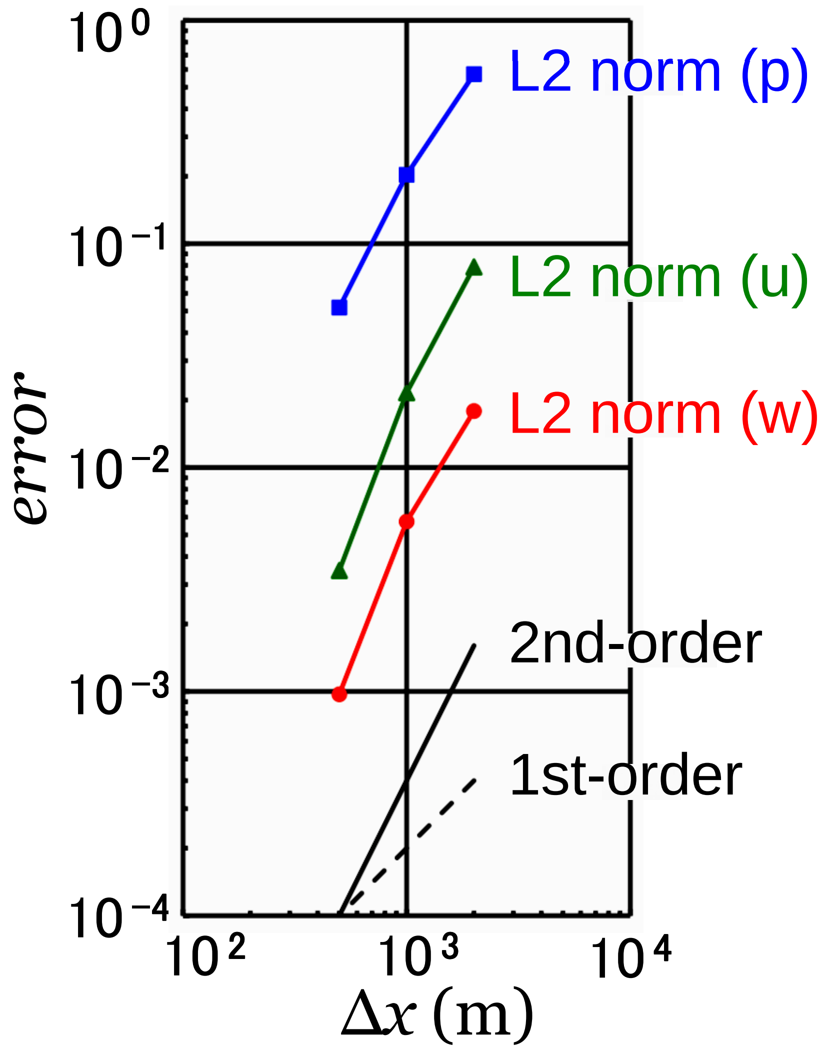
Sayaca-2D reproduces a sufficiently accurate linear mountain waves over the mountain.

Grid refinement study

We estimated the global accuracy of our method using the results with four different grid intervals.

- a = 10 km, h = 400 m
- Δx = 2 km, 1 km, 500 m, and 250 m are used. (Δz is fixed at 200 m.)
- The errors are calculated by assuming the solution with Δx = 250 m as the exact solution.

Sayaca-2D reproduces results of globally 2nd-order accurate.

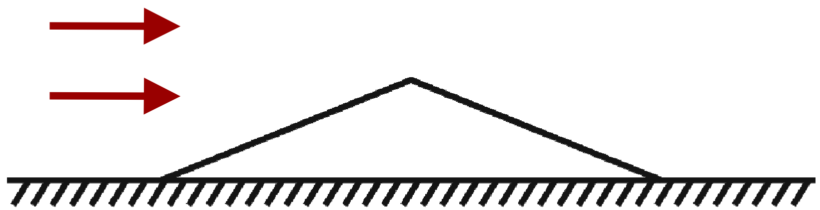


Numerical simulations of flow over slopes of various angles are performed using Sayaca-2D.

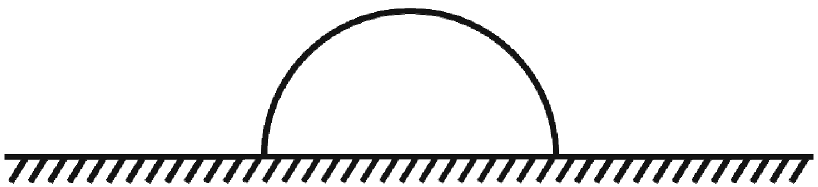
(1) a bell-shaped mountain

(2) a pyramidal mountain

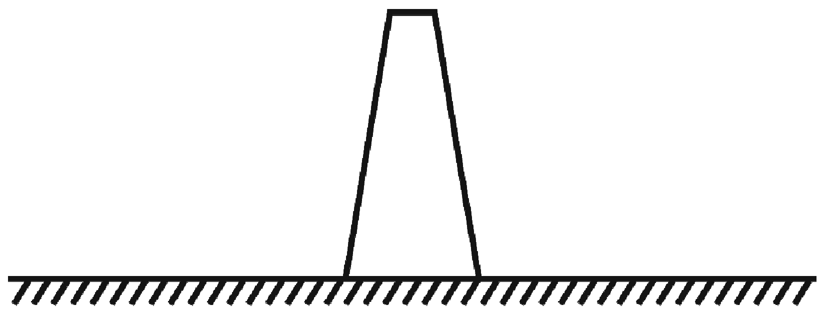
$U_o = 10\ \mathrm{ms}^{-1}$ $N = 0.01\ \mathrm{s}^{-1}$



(3) a semicircular mountain

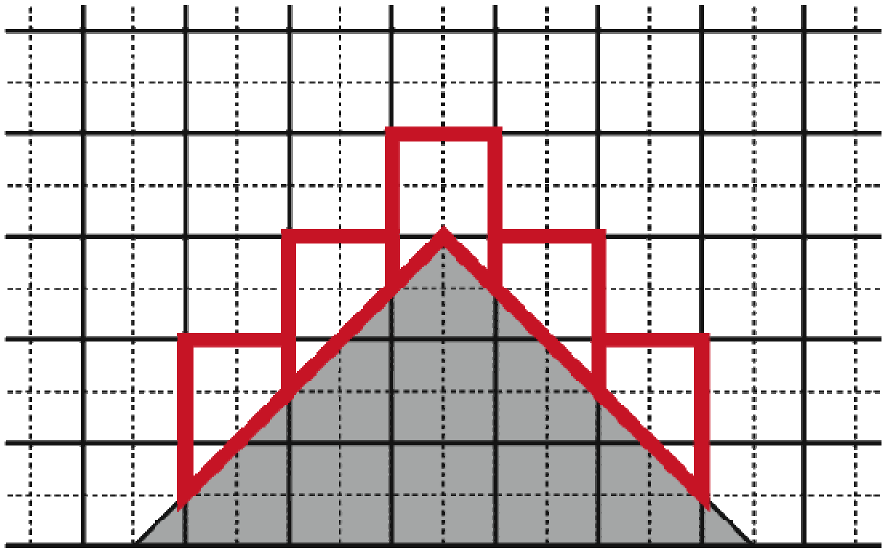


(4) a cliff

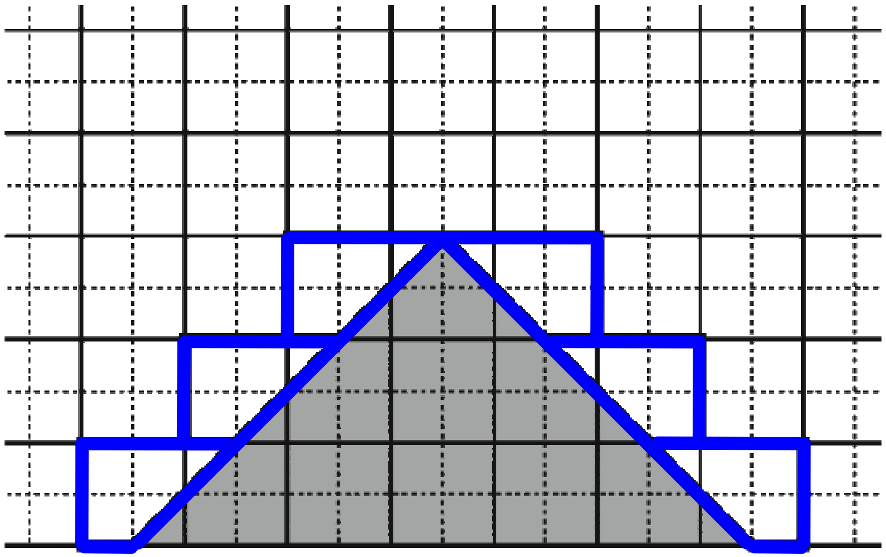


Flow over a pyramidal mountain

a) vertical combintation



b) horizontal combination

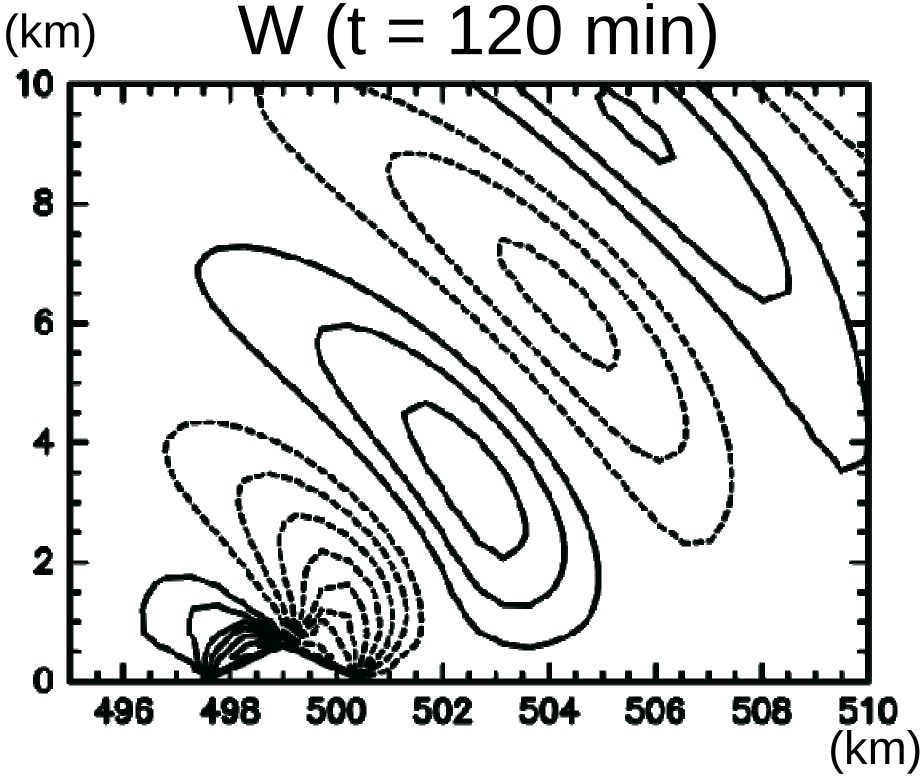


(Δx = 500 m, Δz = 200 m)

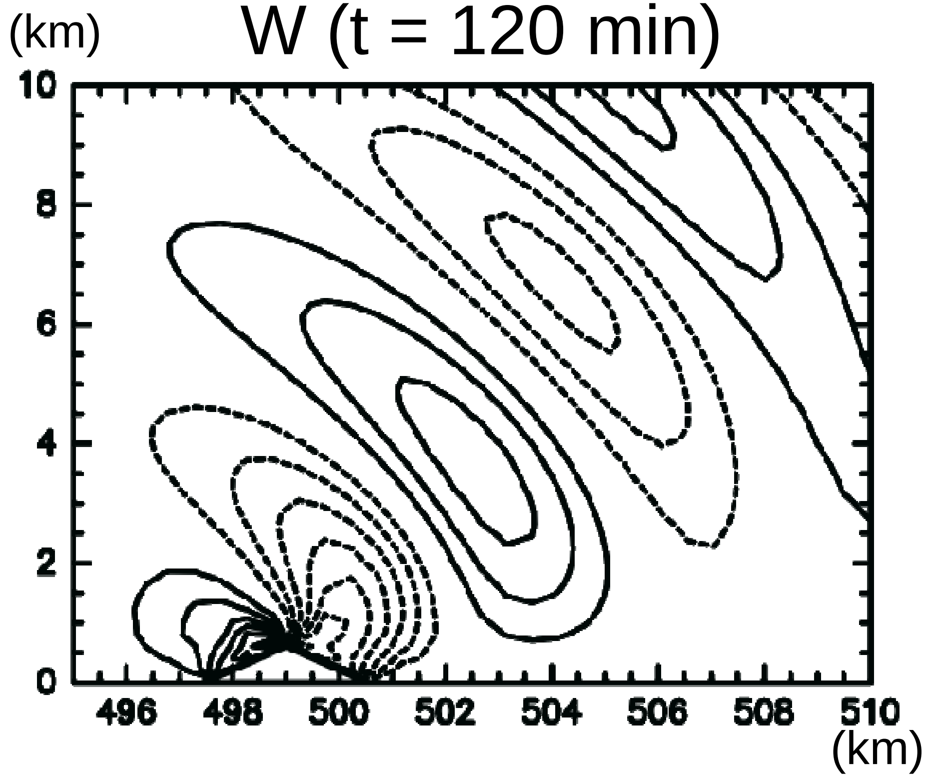
max. slope angle : 20.8 degrees

Flow over a pyramidal mountain

a) vertical combintation



b) horizontal combination



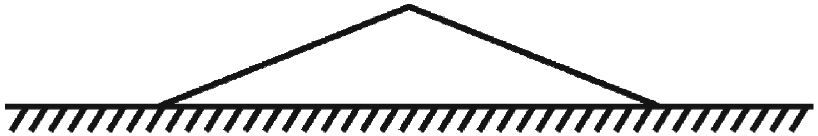
(contour interval : 0.5 ms^{-1})

Two combinations produce consistent results.

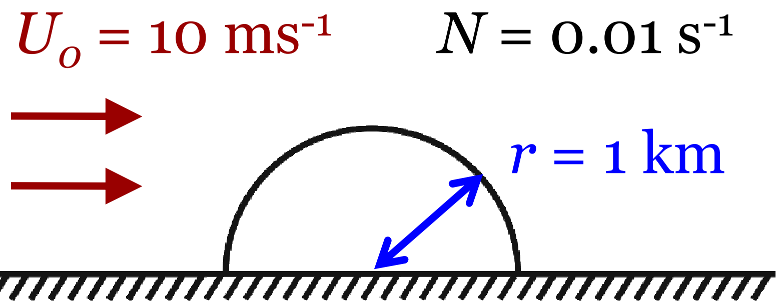
Numerical simulations of flow over slopes of various angles are performed using Sayaca-2D.

(1) a bell-shaped mountain

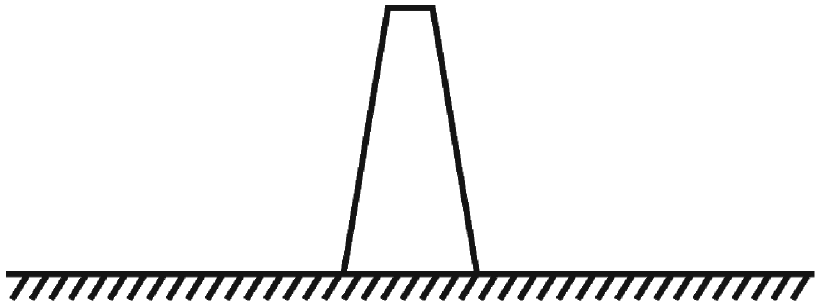
(2) a pyramidal mountain



(3) a semicircular mountain



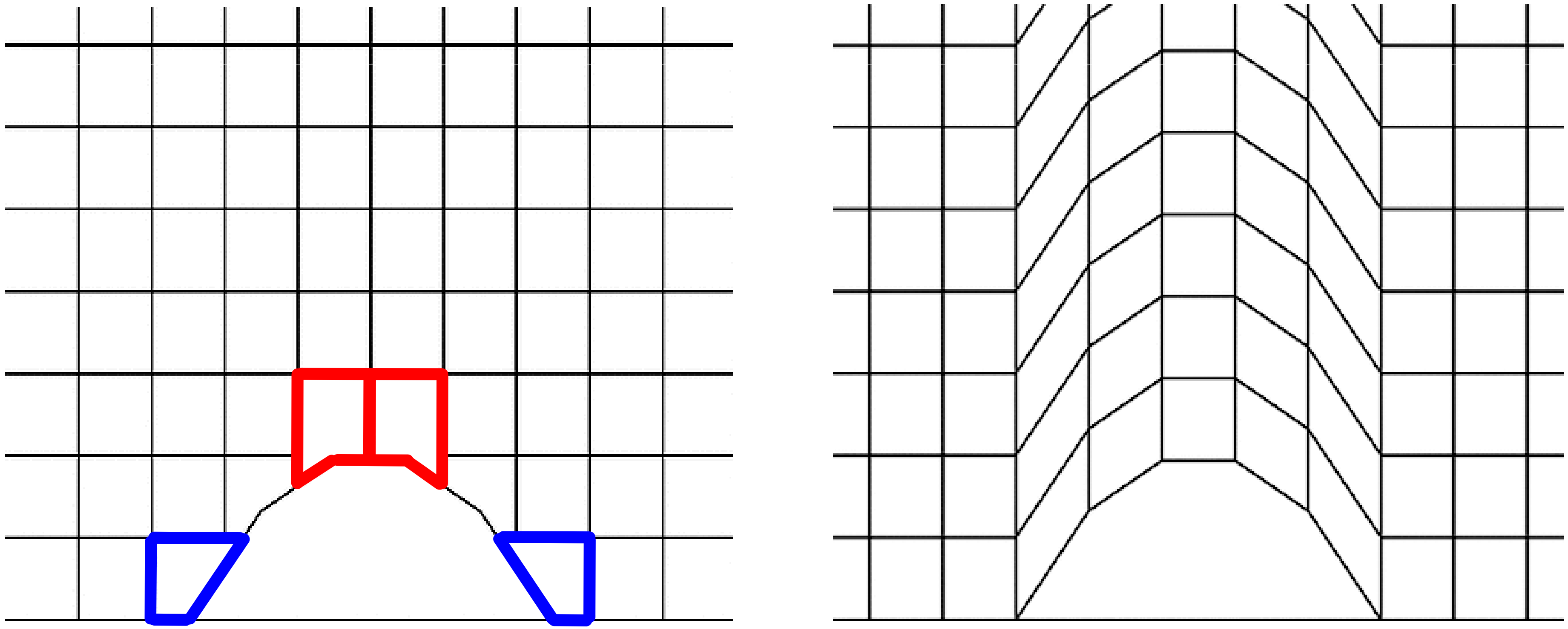
(4) a cliff



Flow over a semicircular mountain

a) Sayaca-2D

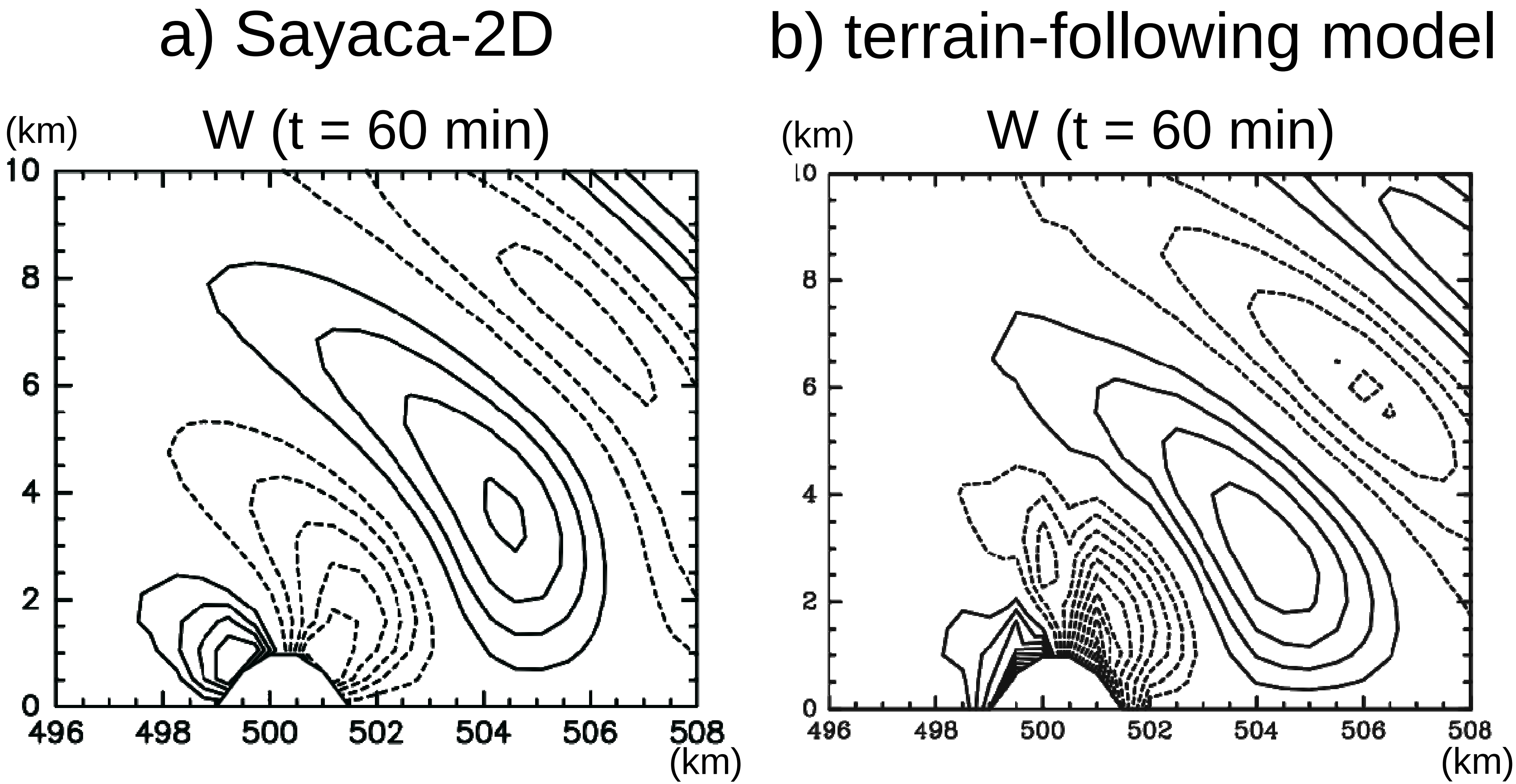
b) terrain-following model (Satomura 1989)



($\Delta x = \Delta z = 500$ m)

maximum slope angle : 52.9 degrees

Flow over a semicircular mountain



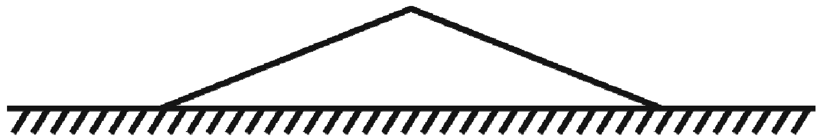
(contour interval : 1 ms^{-1})

Sayaca-2D reproduces an accurate flow over the semicircular mountain.

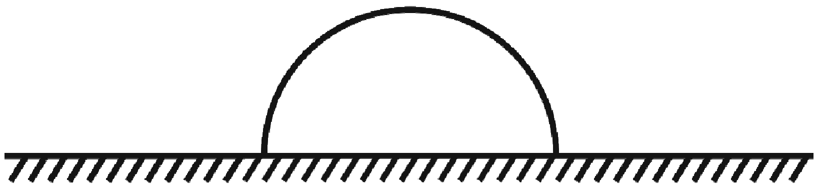
Numerical simulations of flow over slopes of various angles are performed using Sayaca-2D.

(1) a bell-shaped mountain

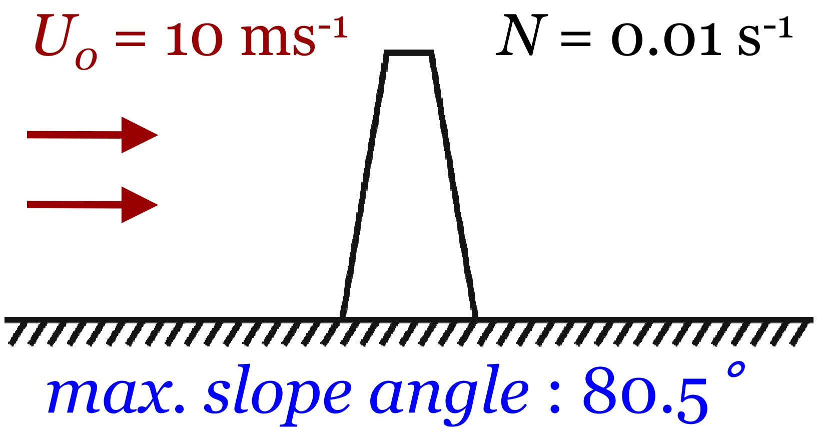
(2) a pyramidal mountain



(3) a semicircular mountain

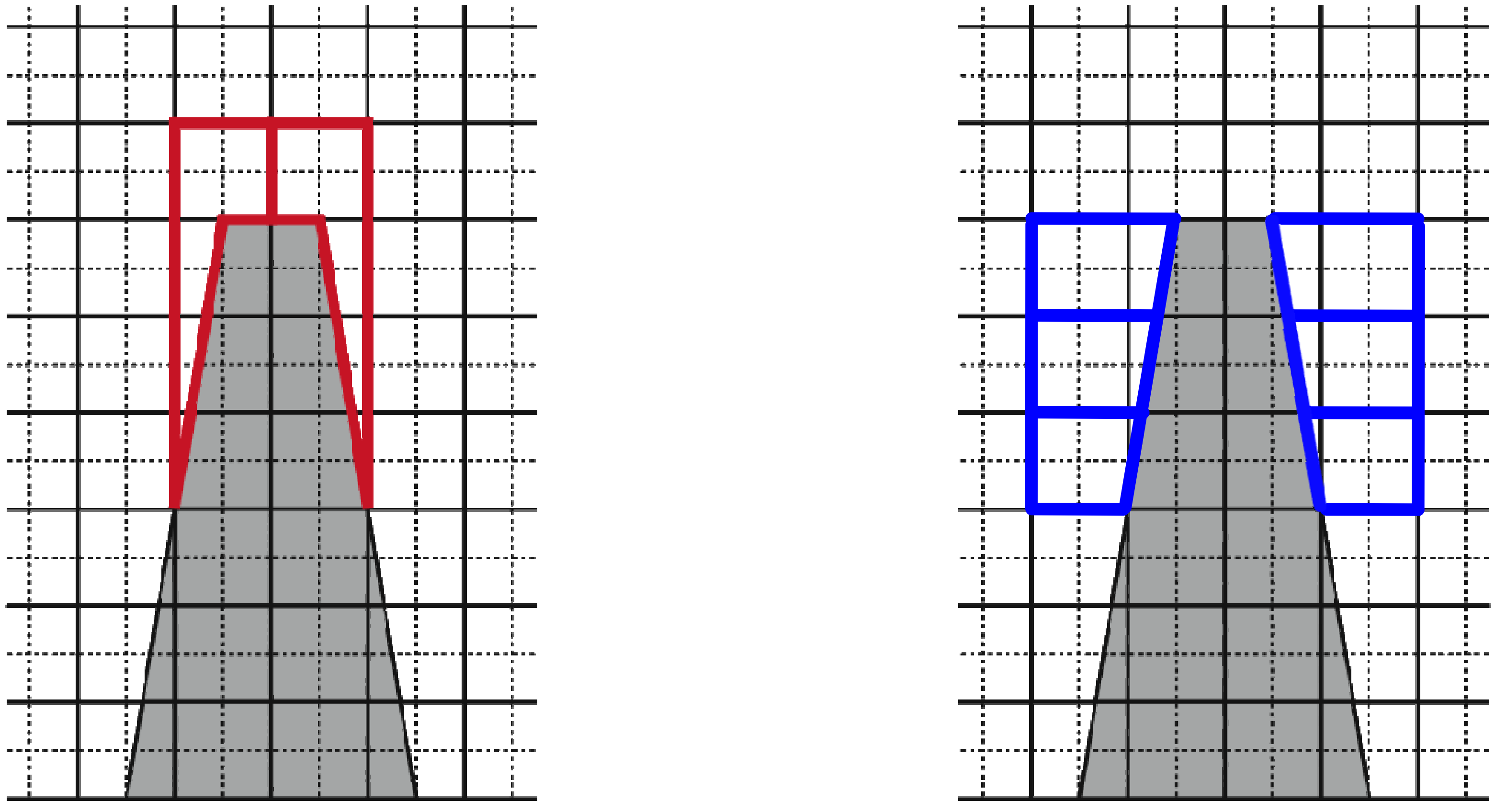


(4) a cliff



Flow over a semicircular mountain

a) vertical combintation   b) horizontal combination

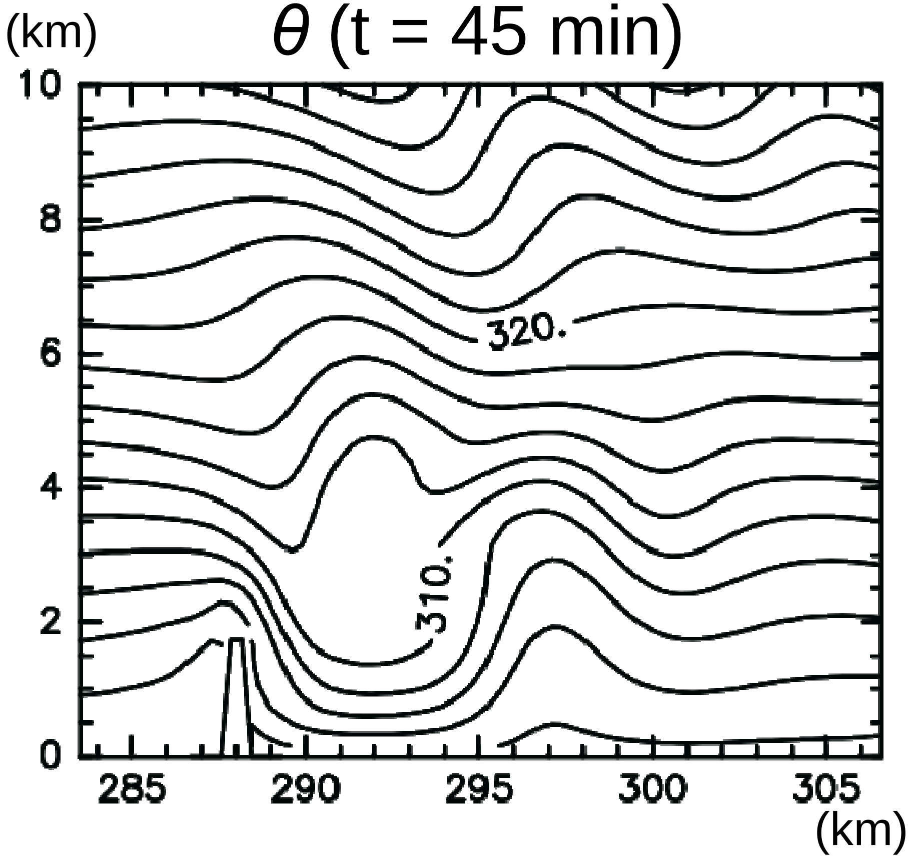


($\Delta x = \Delta z = 288.3$ m)

Flow over a semicircular mountain

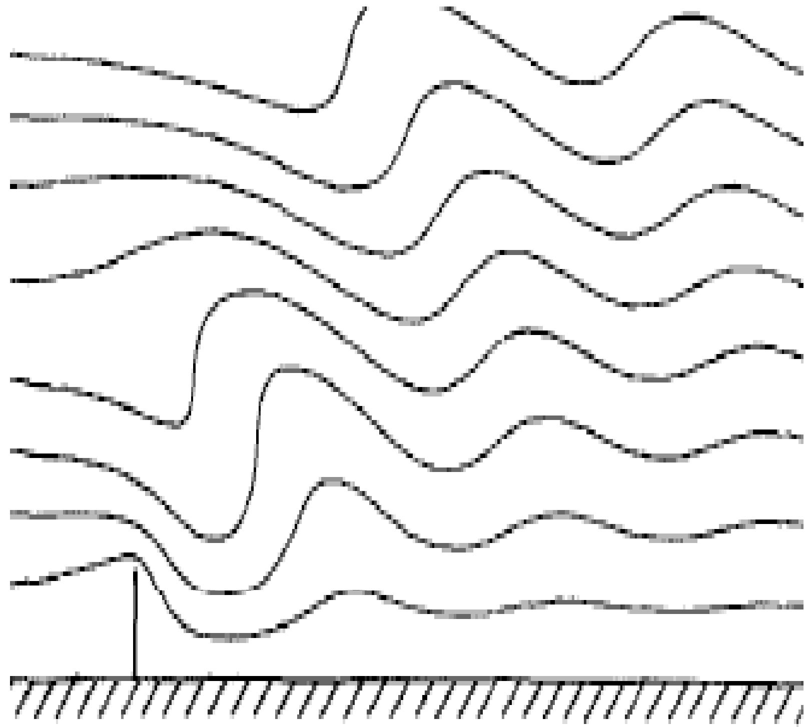
a) vertical combintation

θ (t = 45 min)



analytical solution

Streamline



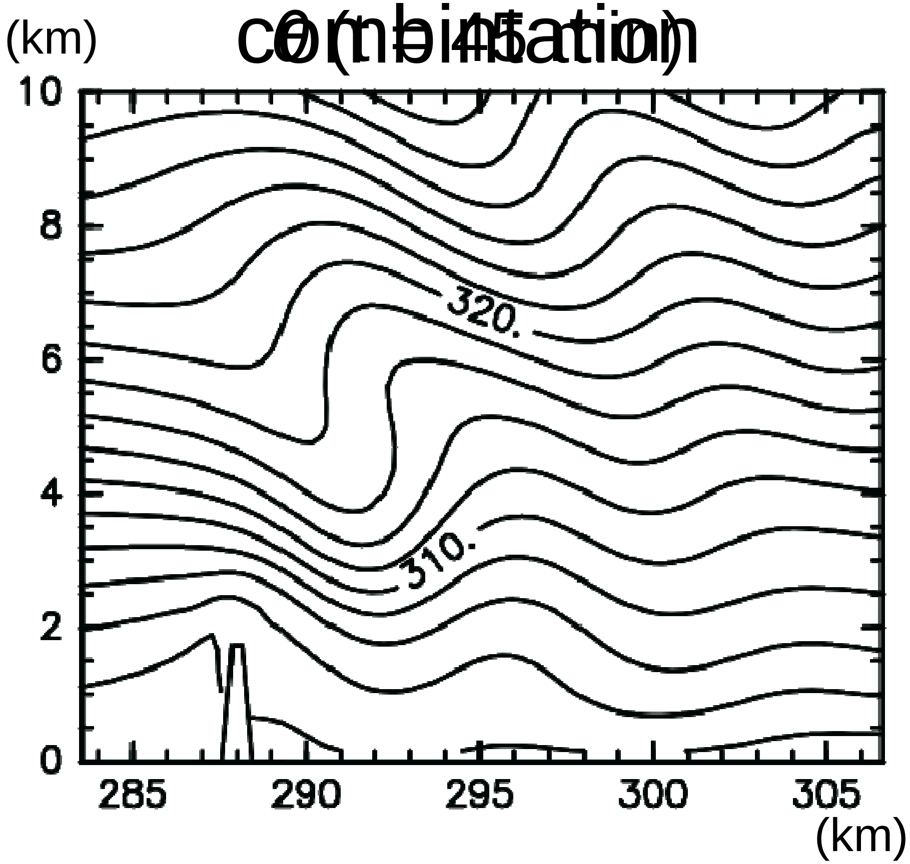
(Huppert&Miles 1969)

(contour interval : 2 K)

Flow over a semicircular mountain

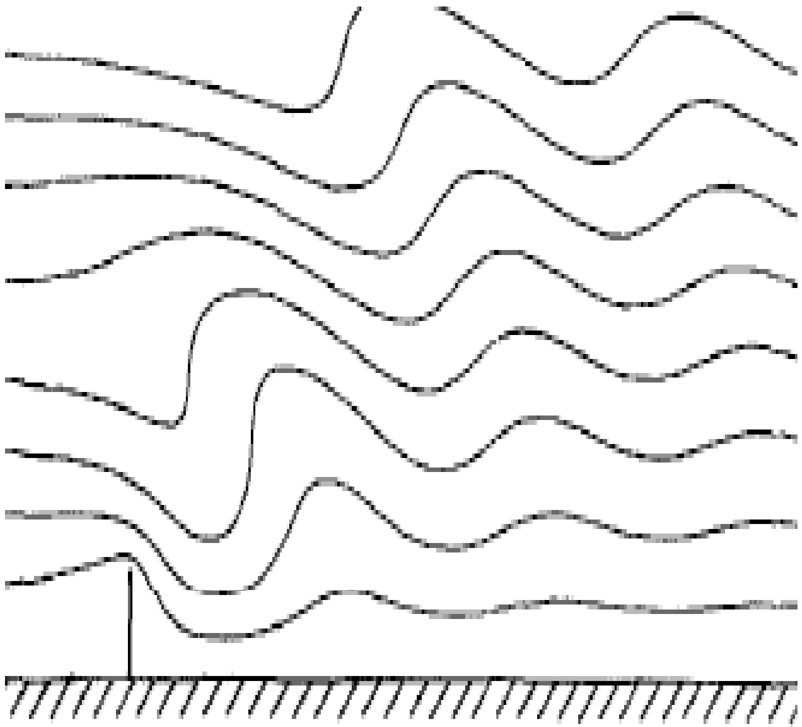
b) horizontal combination

θ (t=45 min)



analytical solution

Streamline



(Huppert&Miles 1969)

Sayaca-2D with horizontal combinations successfully reproduces flow over a cliff with slopes of over 80° .

Contents

A new cut cell method is introduced to
a nonhydrostatic atmospheric model “Sayaca-2D”.

- Cell-merging is used for the small cell problem.
- A unique arrangement of variables simplifies the calculation near the boundary.
- Sayaca-2D successfully reproduced flows over a wide range of slopes, from a gently sloping mountain to an extremely steep cliff.

Sayaca-2D has the advantage for ultra-high resolution simulations with complex topography.

Contents