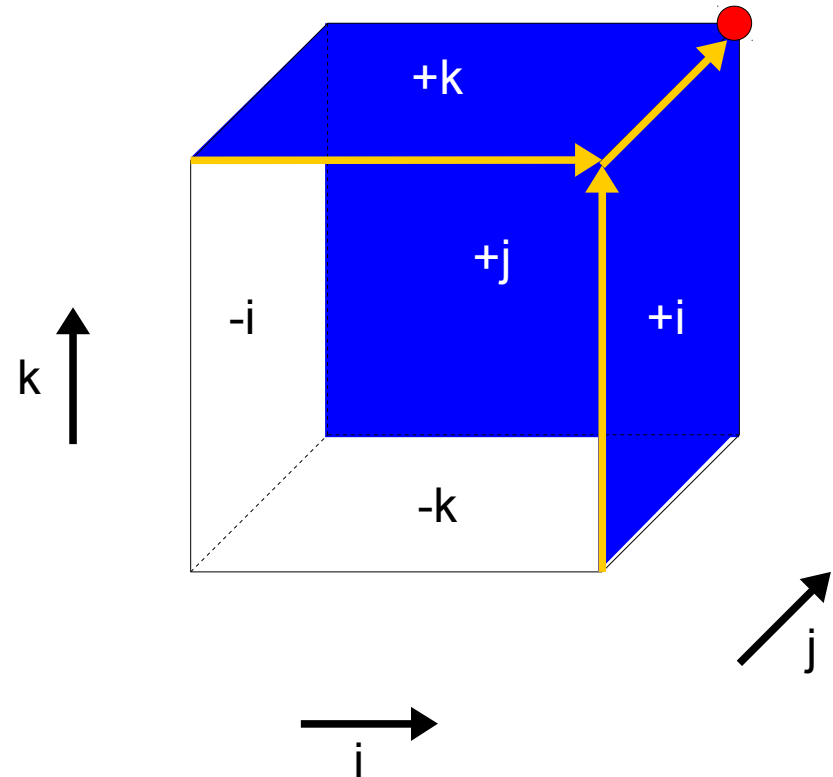
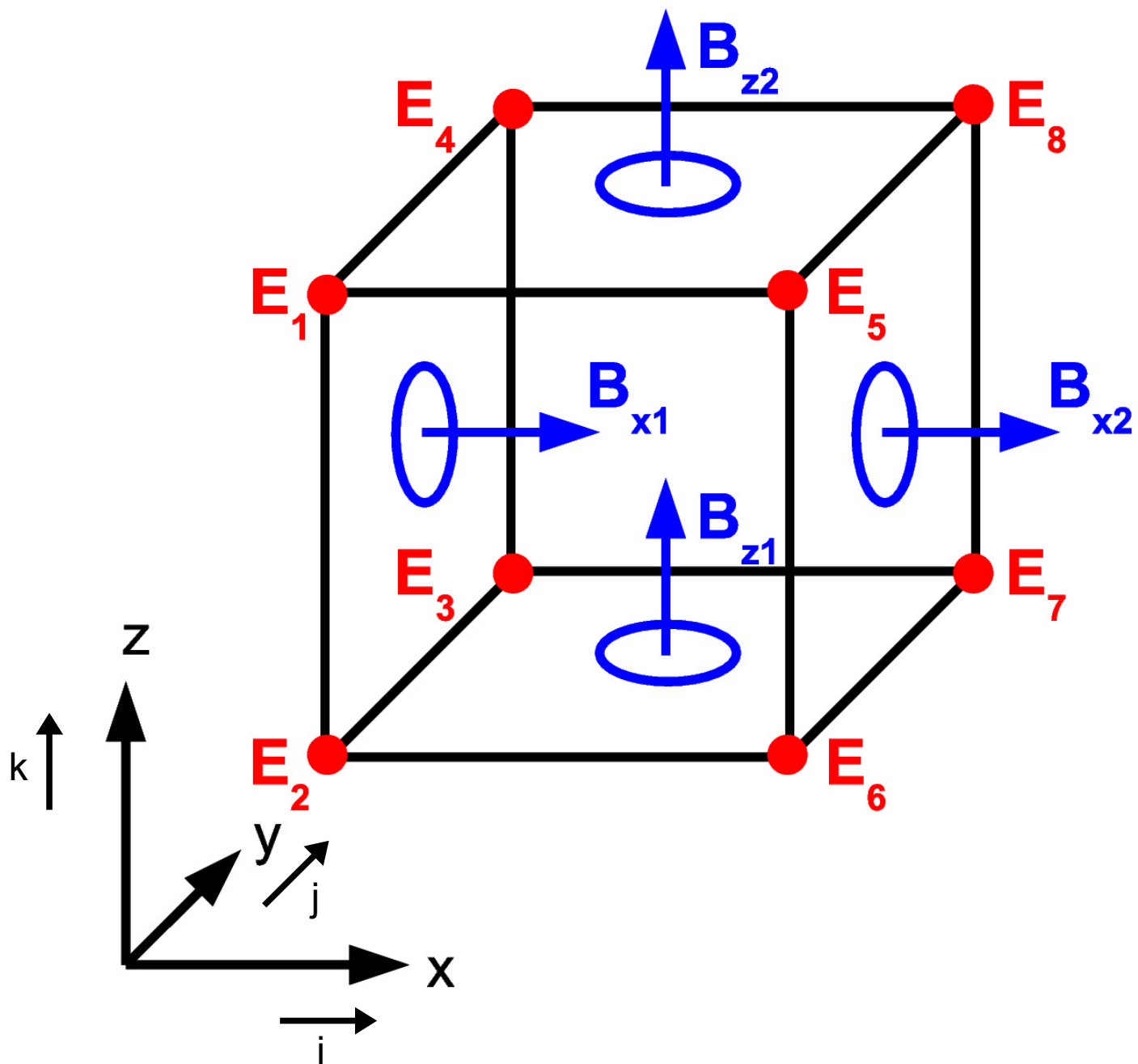


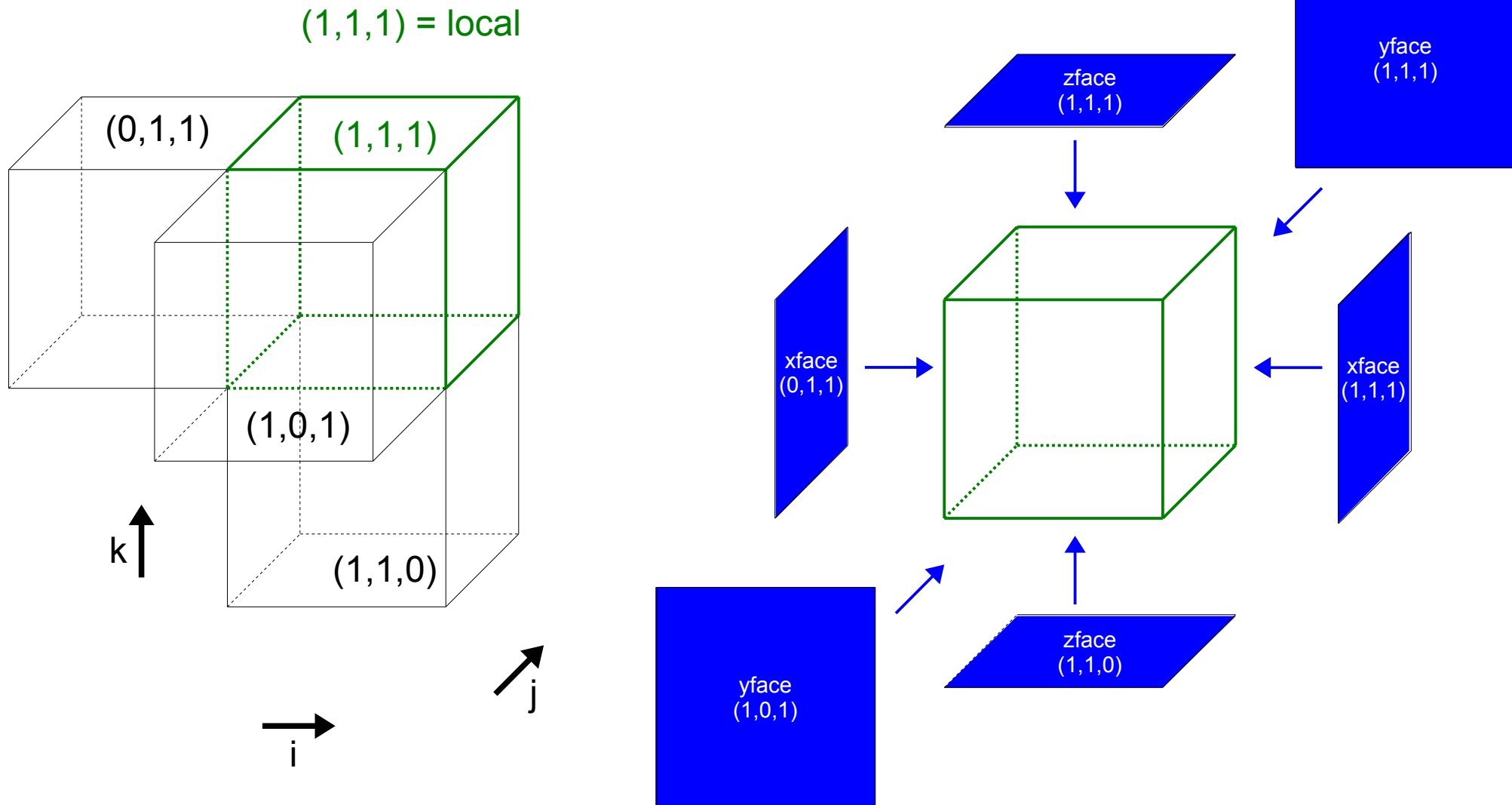
RHybrid cell indexing

- cell volume average
 - B_x, B_y, B_z
- upper front corner **node** $(+i, +j, +k)$
 - E_x, E_y, E_z
 - B_x, B_y, B_z
 - J_x, J_y, J_z
- three **face** surface averages
 - $\Phi_x = dA \times B_x$ (+i face)
 - $\Phi_x = dA \times B_y$ (+j face)
 - $\Phi_z = dA \times B_z$ (+k face)
- **edges**
 - J_x, J_y, J_z



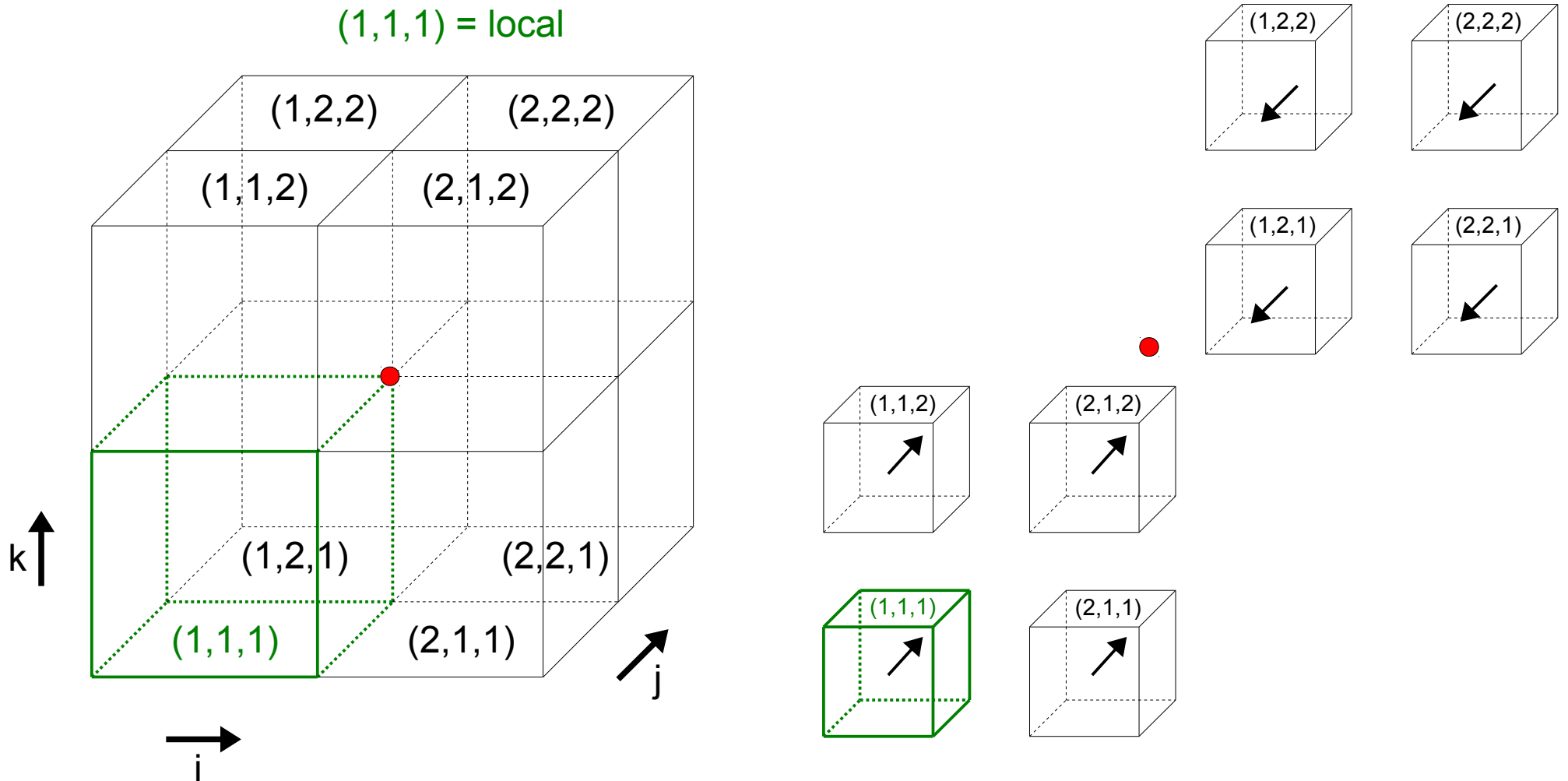


face to cell interpolation



$$\begin{aligned}
 \text{celldata}_x &= 0.5 \times (\text{facedata}_x(1,1,1) + \text{facedata}_x(0,1,1)) \\
 \text{celldata}_y &= 0.5 \times (\text{facedata}_y(1,1,1) + \text{facedata}_y(1,0,1)) \\
 \text{celldata}_z &= 0.5 \times (\text{facedata}_z(1,1,1) + \text{facedata}_z(1,1,0))
 \end{aligned}$$

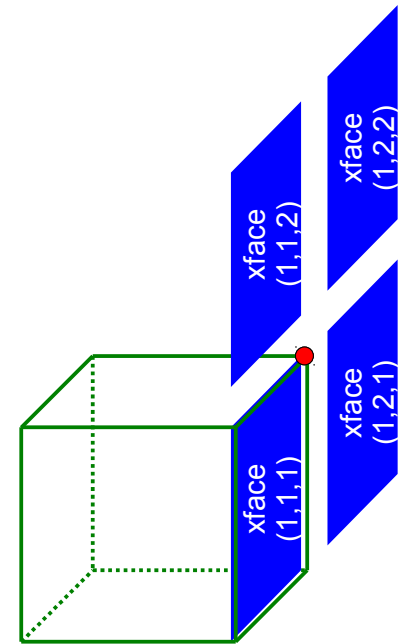
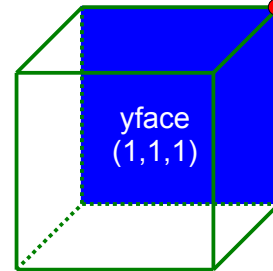
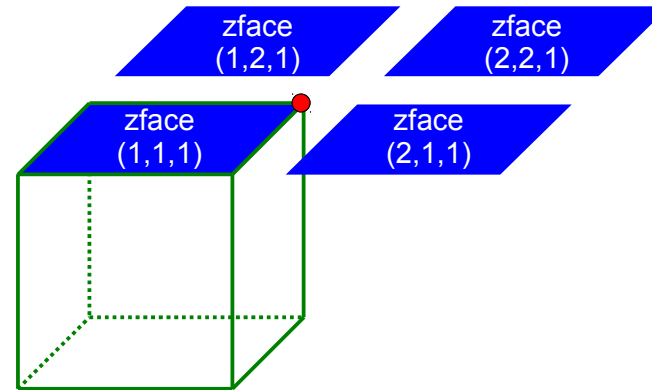
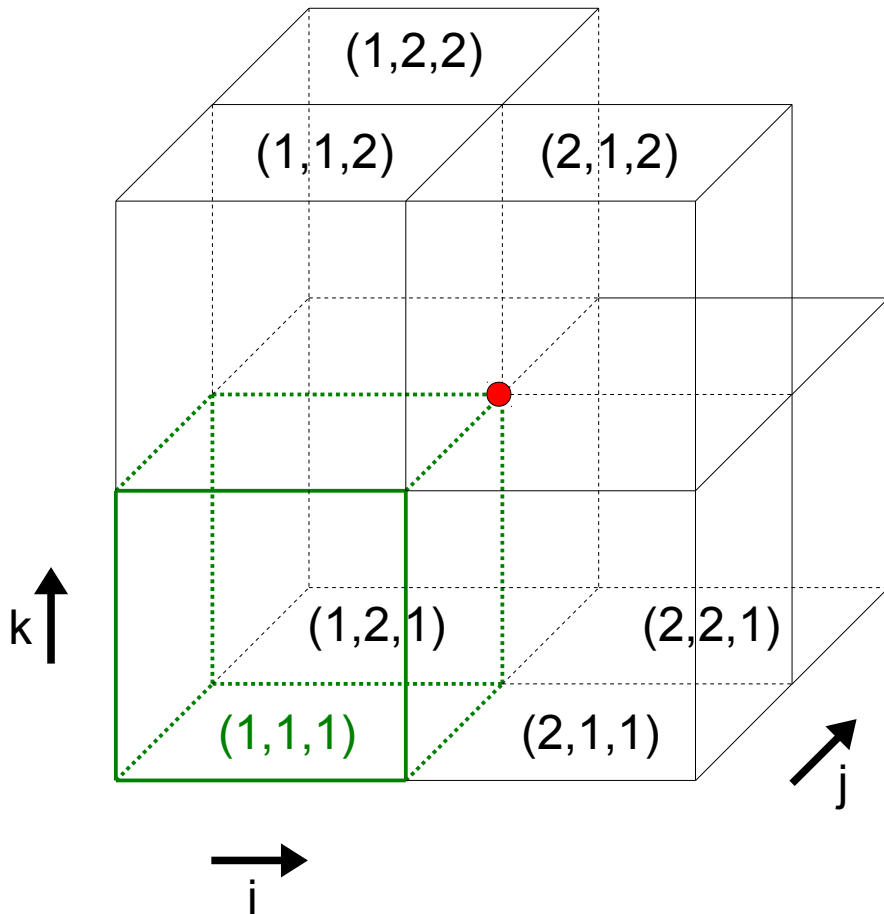
cell to node interpolation



$$\text{nodedata}_i = 0.125 \times (\text{celldata}_i(1,1,1) + \text{celldata}_i(2,1,1) + \text{celldata}_i(1,1,2) + \text{celldata}_i(2,1,2) + \text{celldata}_i(1,2,1) + \text{celldata}_i(2,2,1) + \text{celldata}_i(1,2,2) + \text{celldata}_i(2,2,2))$$

face to node interpolation

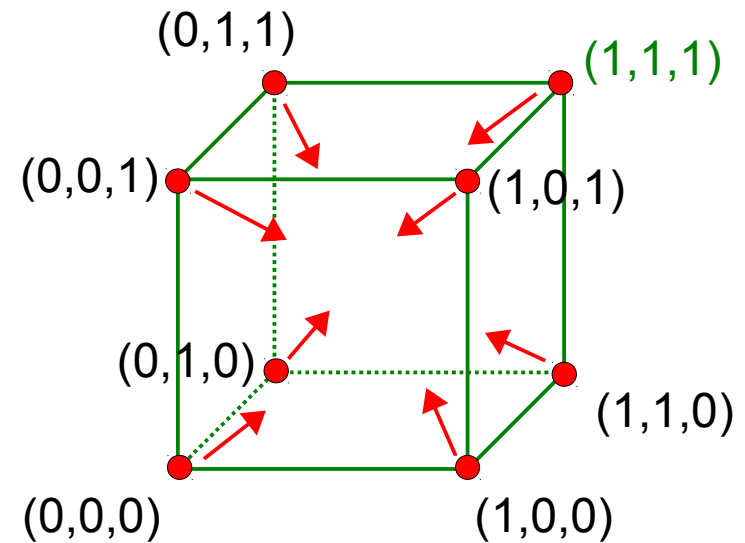
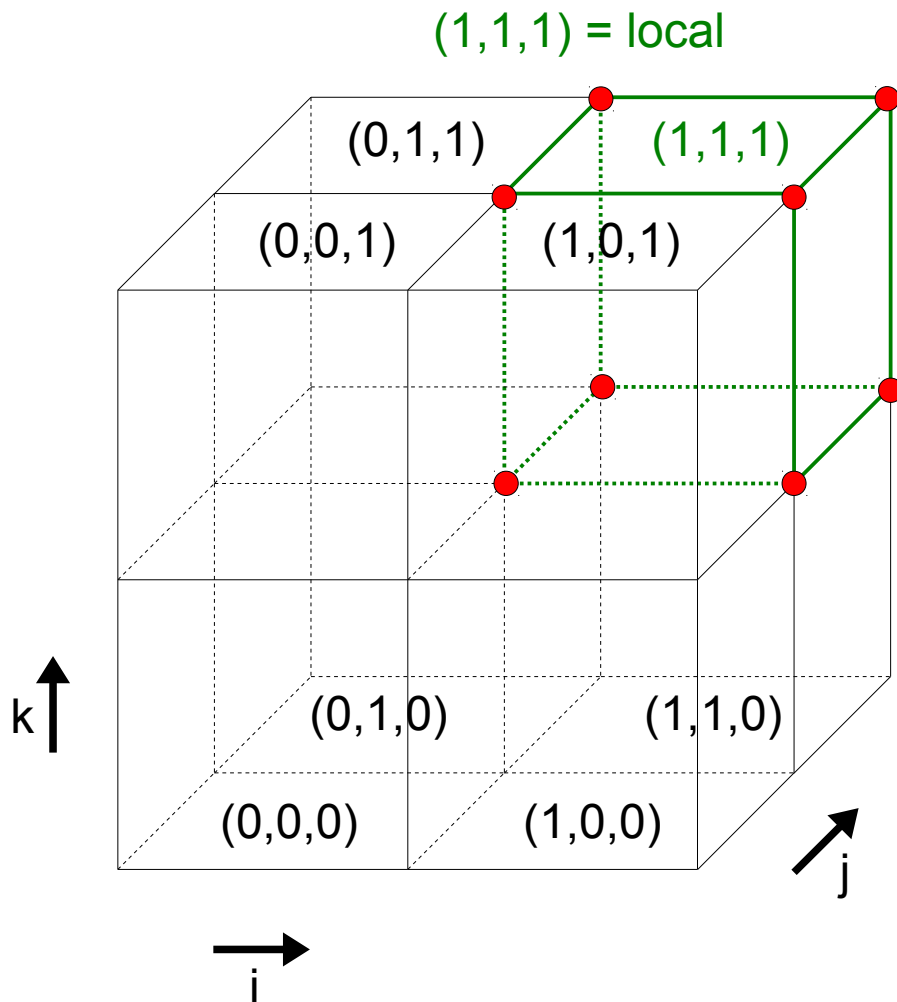
(1,1,1) = local



```

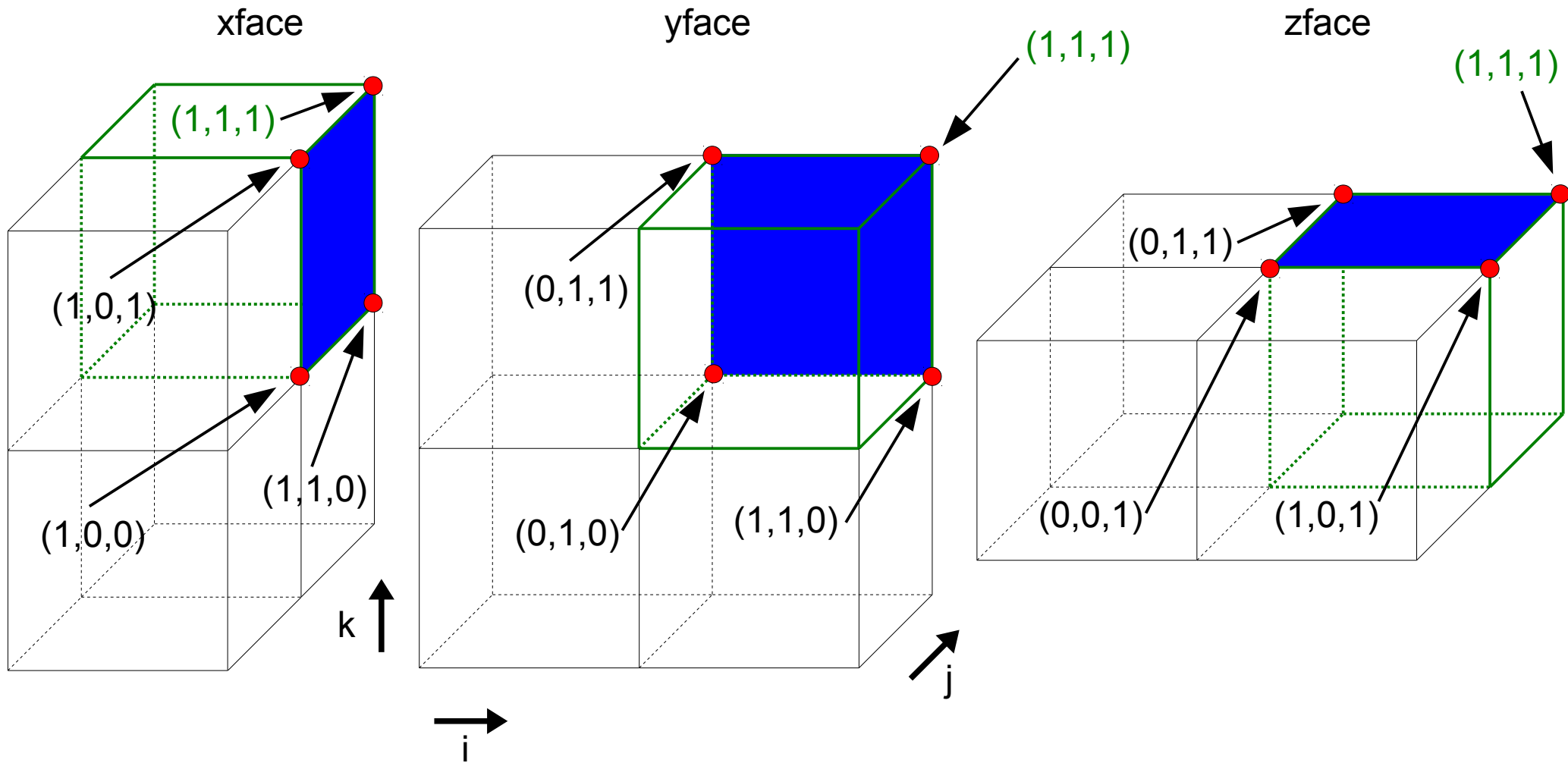
nodedata_x = 0.25 × ( facedata_x(1,1,1) + facedata_x(1,1,2) + facedata_x(1,2,2) + facedata_x(1,2,1) )
nodedata_y = 0.25 × ( facedata_y(1,1,1) + facedata_y(1,1,2) + facedata_y(2,1,2) + facedata_y(2,1,1) )
nodedata_z = 0.25 × ( facedata_z(1,1,1) + facedata_z(1,2,1) + facedata_z(2,2,1) + facedata_z(2,1,1) )
    
```

node to cell interpolation



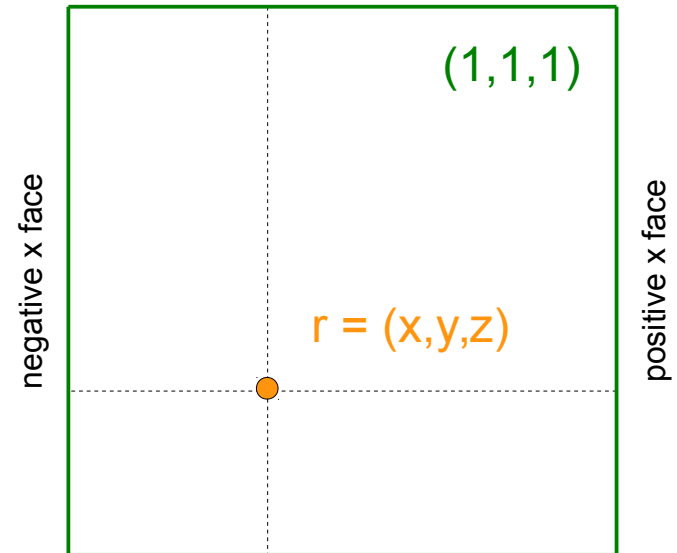
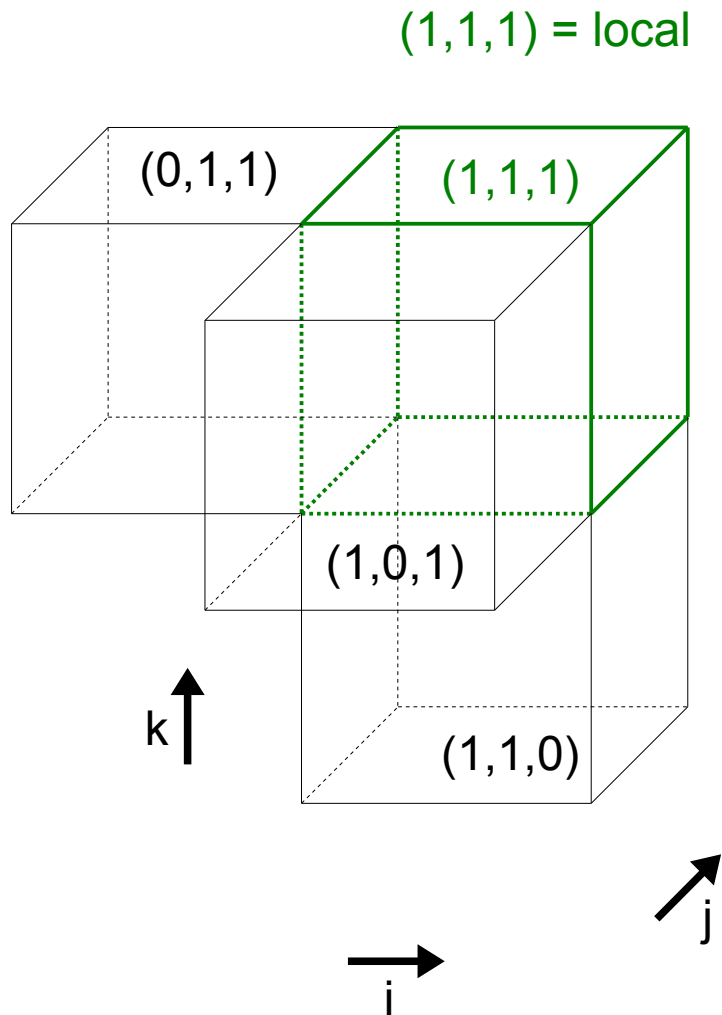
$$\begin{aligned} \text{celldata}_i = 0.125 \times & (\text{nodedata}_i(1,1,1) + \text{nodedata}_i(0,1,1) \\ & + \text{nodedata}_i(0,0,1) + \text{nodedata}_i(1,0,1) \\ & + \text{nodedata}_i(1,1,0) + \text{nodedata}_i(0,1,0) \\ & + \text{nodedata}_i(0,0,0) + \text{nodedata}_i(1,0,0)) \end{aligned}$$

node to face interpolation



```
facedata_x = 0.25 × ( nodedata_x(1,1,1) + nodedata_x(1,0,1) + nodedata_x(1,0,0) + nodedata_x(1,1,0) )
facedata_y = 0.25 × ( nodedata_y(1,1,1) + nodedata_y(1,1,0) + nodedata_y(0,1,0) + nodedata_y(0,1,1) )
facedata_z = 0.25 × ( nodedata_z(1,1,1) + nodedata_z(0,1,1) + nodedata_z(0,0,1) + nodedata_z(1,0,1) )
```

face to r interpolation

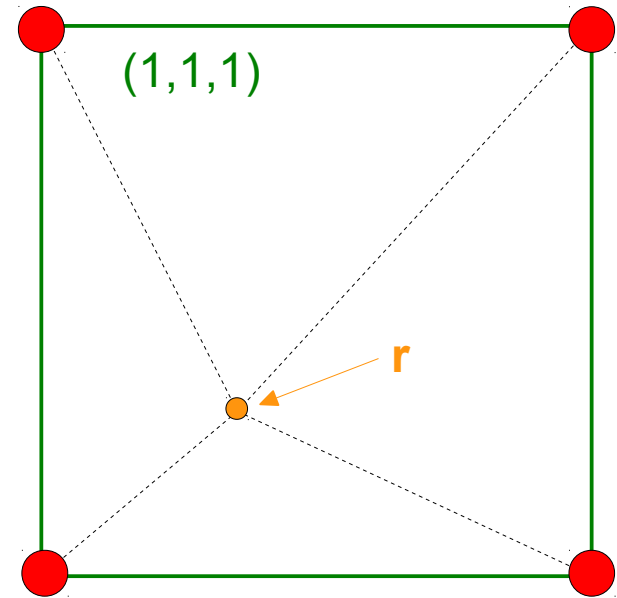
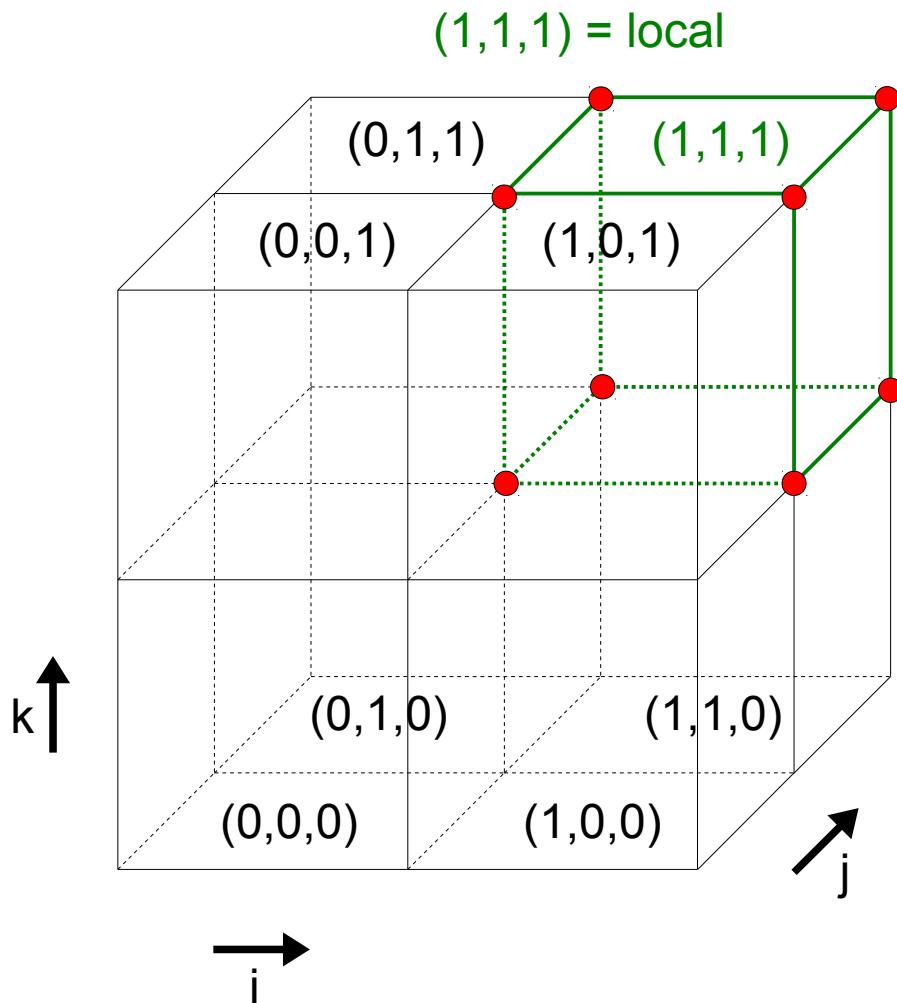


$$\begin{array}{cc} \leftarrow & \rightarrow \\ x - x_{\text{face}}(0,1,1) & x_{\text{face}}(1,1,1) - x \end{array}$$

$$\begin{aligned} w111x &= +(x - x_{\text{face}}(0,1,1)) / dx \\ w111y &= +(y - y_{\text{face}}(1,0,1)) / dx \\ w111z &= +(z - z_{\text{face}}(1,1,0)) / dx \\ w011x &= -(x - x_{\text{face}}(1,1,1)) / dx \\ w101y &= -(y - y_{\text{face}}(1,1,1)) / dx \\ w110z &= -(z - z_{\text{face}}(1,1,1)) / dx \end{aligned}$$

$$\begin{aligned} \text{data_x}(r) &= w111x \times \text{facedata_x}(1,1,1) + w011x \times \text{facedata_x}(0,1,1) \\ \text{data_y}(r) &= w111y \times \text{facedata_y}(1,1,1) + w101y \times \text{facedata_y}(1,0,1) \\ \text{data_z}(r) &= w111z \times \text{facedata_z}(1,1,1) + w110z \times \text{facedata_z}(1,1,0) \end{aligned}$$

node to r interpolation

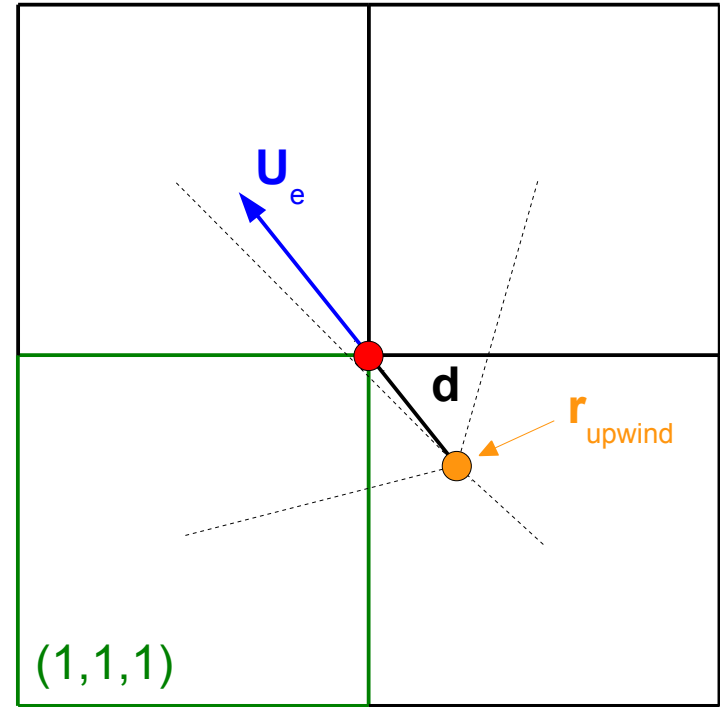
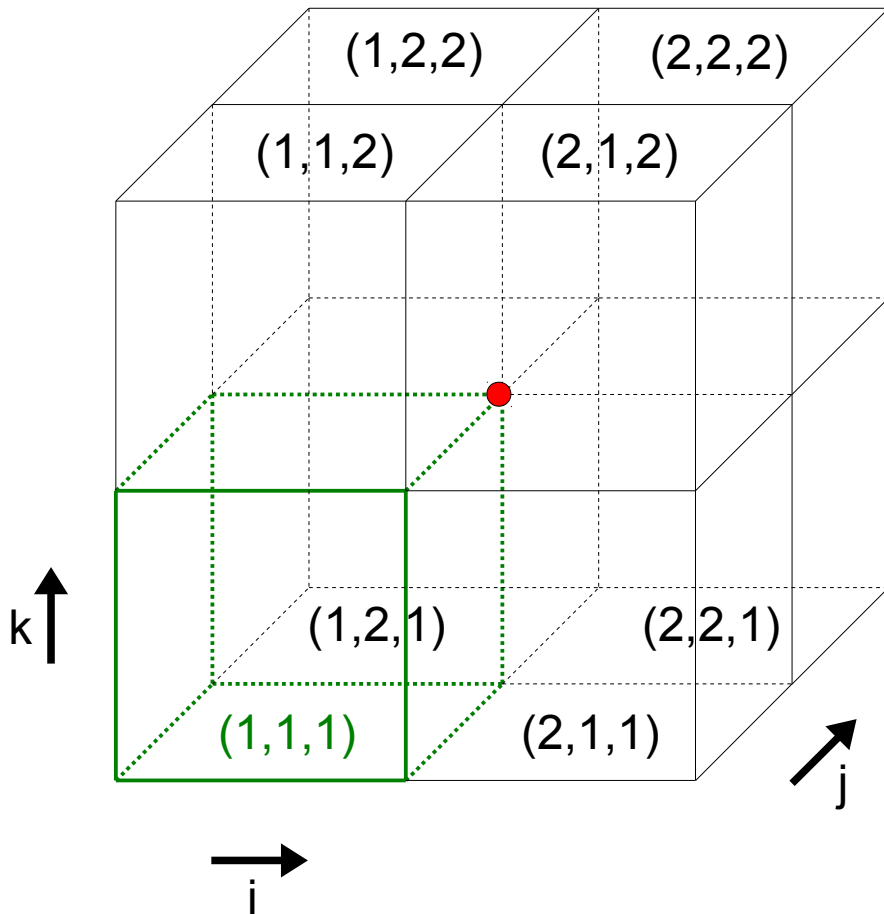


Weight factors: $w_i = 1/|\mathbf{r}_{\text{node}_i} - \mathbf{r}|$
 Sum of weights: $w_{\text{sum}} = \sum_i (w_i)$

$$\begin{aligned} \text{data}(\mathbf{r}) = & (\quad w(1,1,1) \times \text{nodedata}(1,1,1) \quad + w(0,1,1) \times \text{nodedata}(0,1,1) \quad + \\ & w(1,0,1) \times \text{nodedata}(1,0,1) \quad + w(0,0,1) \times \text{nodedata}(0,0,1) \quad + \\ & w(1,1,0) \times \text{nodedata}(1,1,0) \quad + w(0,1,0) \times \text{nodedata}(0,1,0) \quad + \\ & w(1,0,0) \times \text{nodedata}(1,0,0) \quad + w(0,0,0) \times \text{nodedata}(0,0,0) \quad) / w_{\text{sum}} \end{aligned}$$

upwind node data

(1,1,1) = local



Displacement vector: $\mathbf{d} = 0.5 \times \mathbf{U}_e / |\mathbf{U}_e|$

Upwind position: $\mathbf{r}_{\text{upwind}} = \mathbf{r}_{\text{node}} - \mathbf{d}$

Weight factors: $w_i = 1 / |\mathbf{r}_{\text{cell}_i} - \mathbf{r}_{\text{upwind}}|$

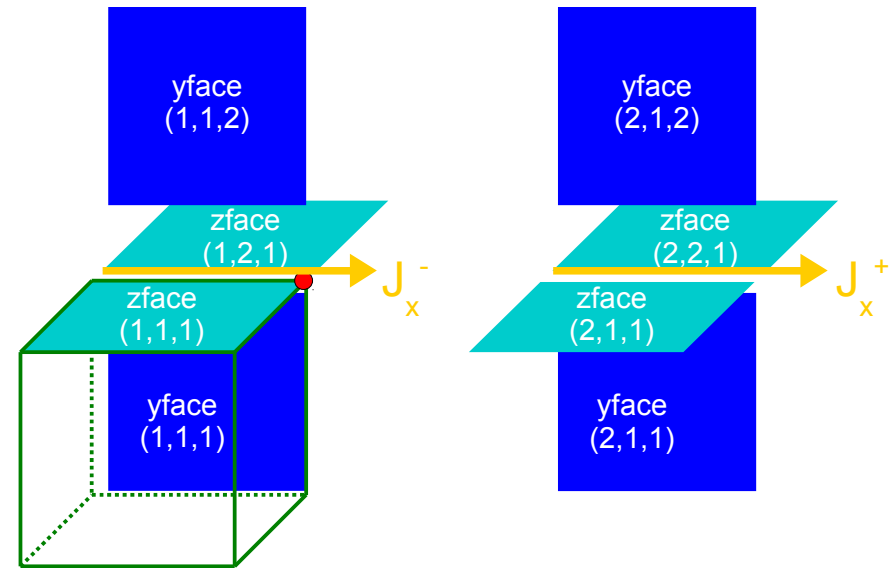
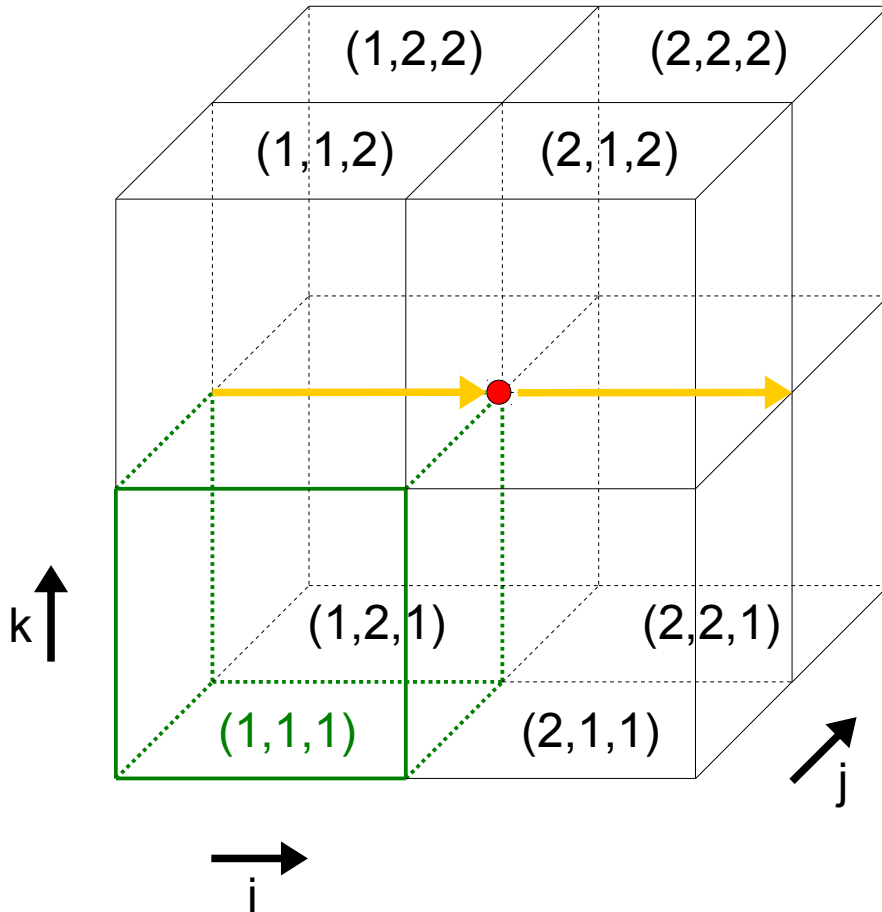
Sum of weights: $w_{\text{sum}} = \sum_i (w_i)$

`Nodedata(1,1,1) → nodedata(1,1,1) =`

```
(
    w(1,1,1) × celldata(1,1,1) + w(1,1,2) × celldata(1,1,2) +
    w(1,2,1) × celldata(1,2,1) + w(2,1,1) × celldata(2,1,1) +
    w(1,2,2) × celldata(1,2,2) + w(2,2,1) × celldata(2,2,1) +
    w(2,1,2) × celldata(2,1,2) + w(2,2,2) × celldata(2,2,2) ) / wsum
```

Calculation of Node Jx

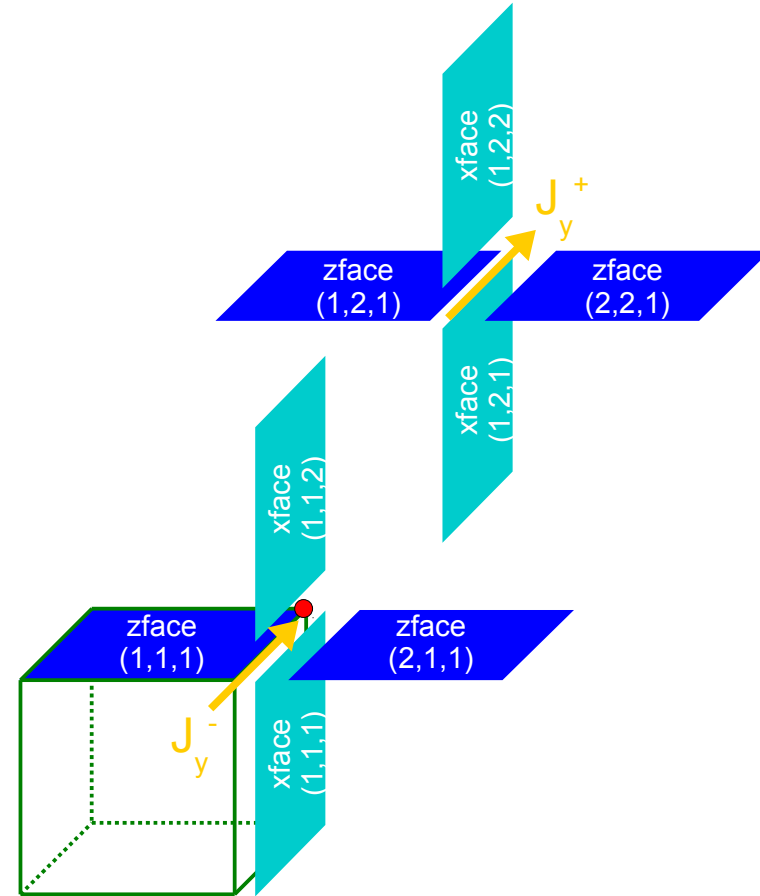
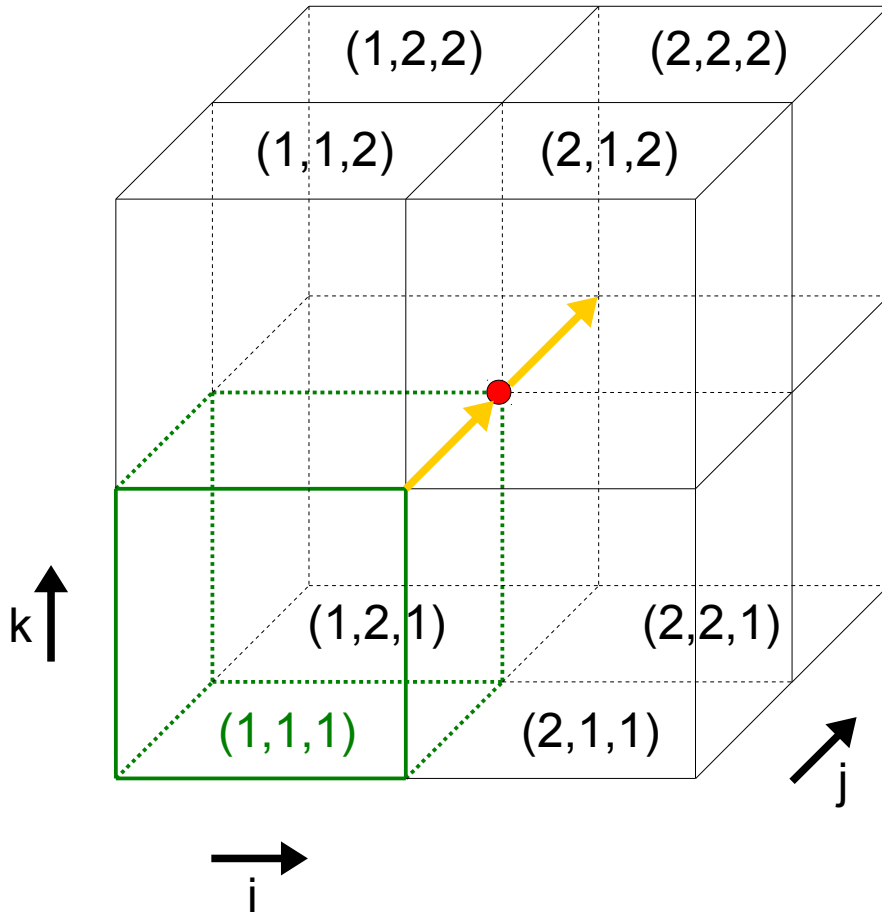
(1,1,1) = local



$$\begin{aligned} \text{edgeJx}^- &= (-\text{faceBz}(1,1,1) + \text{faceBy}(1,1,1) + \text{faceBz}(1,2,1) - \text{faceBy}(1,1,2)) / (dx \times \mu_0) \\ \text{edgeJx}^+ &= (-\text{faceBz}(2,1,1) + \text{faceBy}(2,1,1) + \text{faceBz}(2,2,1) - \text{faceBy}(2,1,2)) / (dx \times \mu_0) \\ \text{nodeJx} &= 0.5 \times (\text{edgeJx}^+ + \text{edgeJx}^-) \end{aligned}$$

Calculation of Node Jy

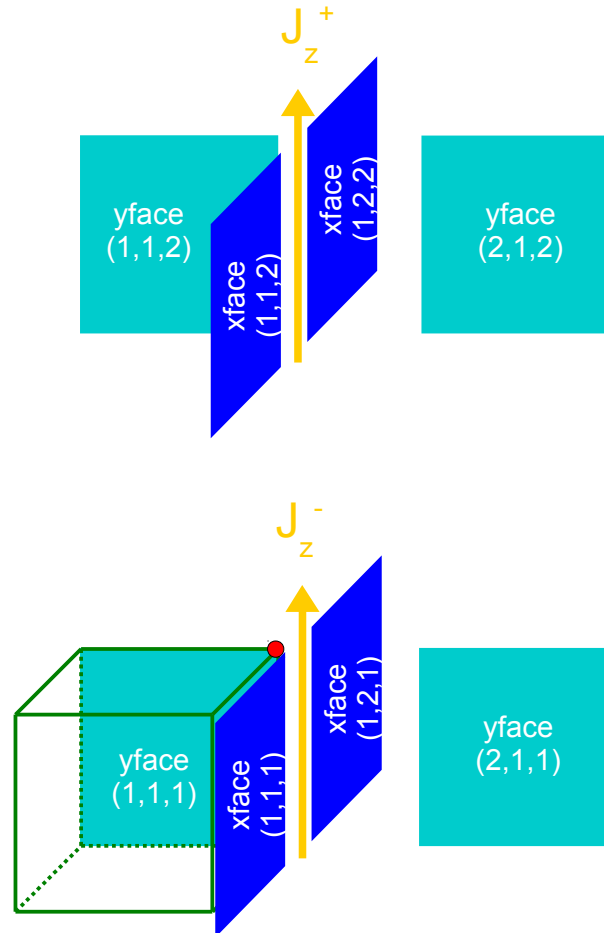
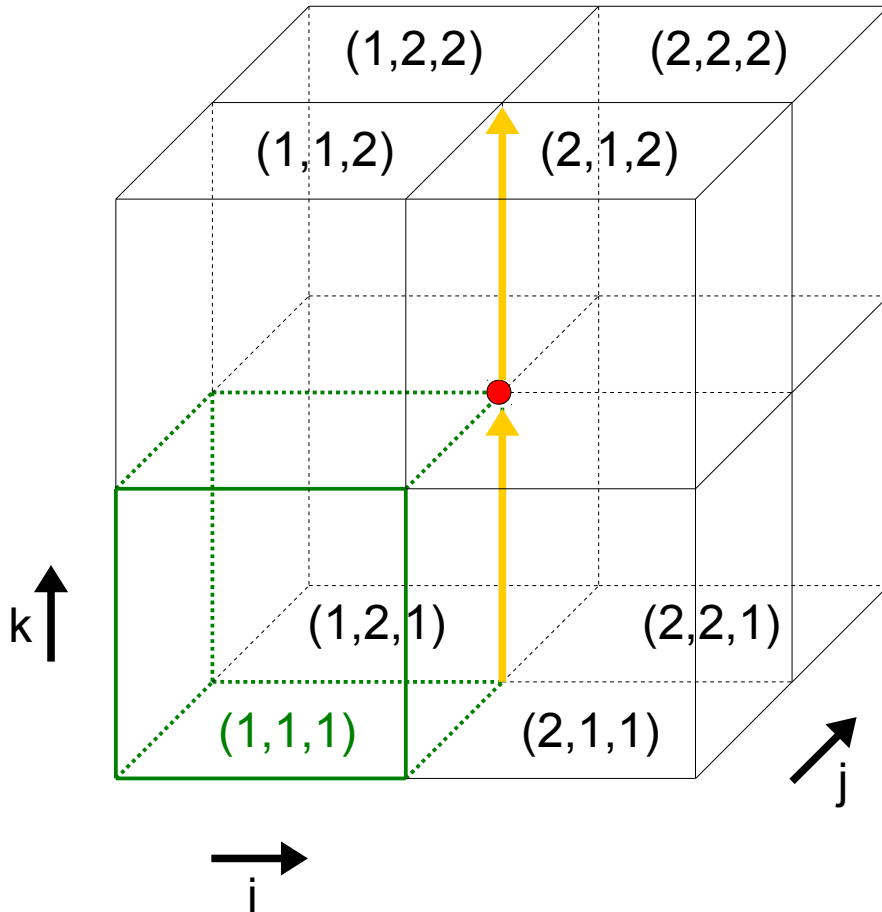
(1,1,1) = local



$$\begin{aligned} \text{edgeJy}^- &= (+\text{faceBz}(1,1,1) + \text{faceBx}(1,1,2) - \text{faceBz}(2,1,1) - \text{faceBx}(1,1,1)) / (dx \times \mu_0) \\ \text{edgeJy}^+ &= (+\text{faceBz}(1,2,1) + \text{faceBx}(1,2,2) - \text{faceBz}(2,2,1) - \text{faceBx}(1,2,1)) / (dx \times \mu_0) \\ \text{nodeJy} &= 0.5 \times (\text{edgeJy}^+ + \text{edgeJy}^-) \end{aligned}$$

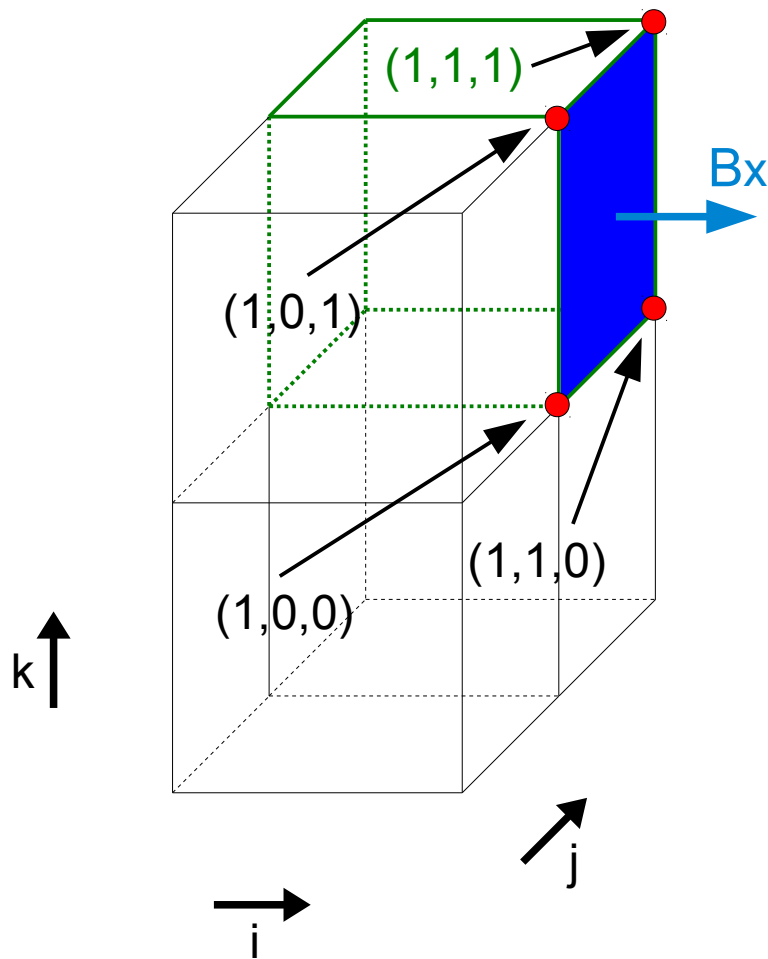
Calculation of Node Jz

(1,1,1) = local

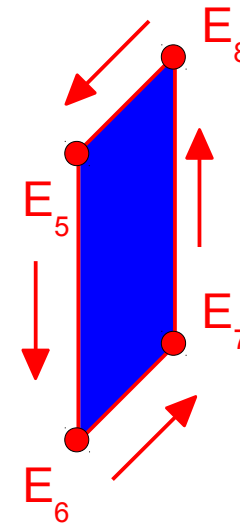


$$\begin{aligned} \text{edgeJz}^- &= (-\text{faceBy}(1,1,1) + \text{faceBx}(1,1,1) + \text{faceBy}(2,1,1) - \text{faceBx}(1,2,1)) / (dx \times \mu_0) \\ \text{edgeJz}^+ &= (-\text{faceBy}(1,1,2) + \text{faceBx}(1,1,2) + \text{faceBy}(2,1,2) - \text{faceBx}(1,2,2)) / (dx \times \mu_0) \\ \text{nodeJz} &= 0.5 \times (\text{edgeJz}^+ + \text{edgeJz}^-) \end{aligned}$$

propagation of B on xface / xface curl



face curl



$$E_5 = E(1,0,1)$$

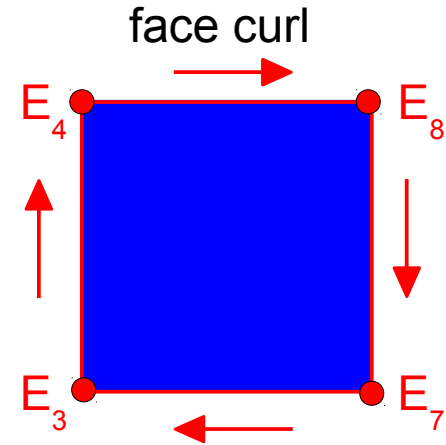
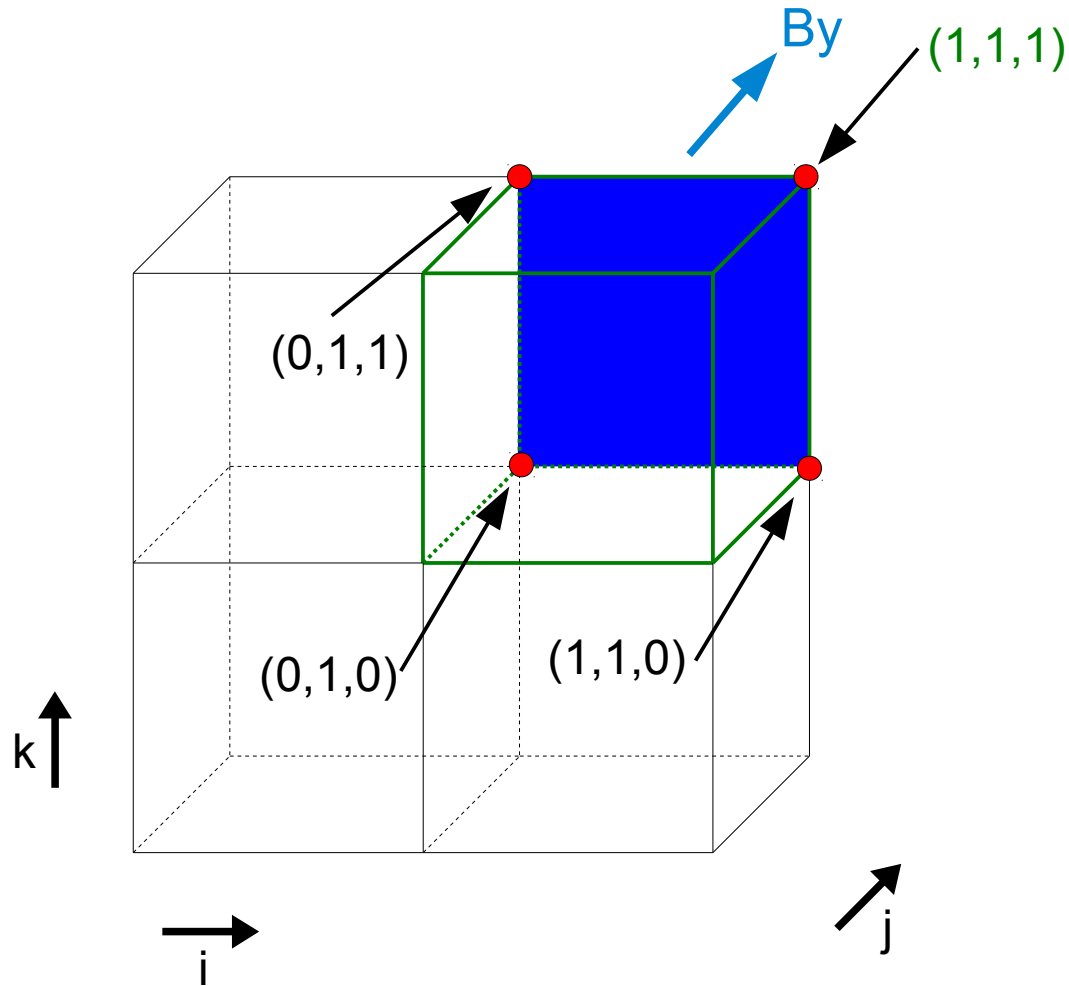
$$E_6 = E(1,0,0)$$

$$E_7 = E(1,1,0)$$

$$E_8 = E(1,1,1) = \text{local}$$

$$\begin{aligned} \frac{\partial(\oint d\mathbf{A}_x \cdot \mathbf{B})}{\partial t} &= -\oint d\mathbf{A}_x \cdot (\nabla \times \mathbf{E}) \\ \mathbf{J}_x &= (\nabla \times \mathbf{B})_x / \mu_0 \end{aligned} \quad \begin{aligned} &= 0.5 \times dx \times (E_{5z} + E_{6z} - E_{6y} - E_{7y} - E_{7z} - E_{8z} + E_{8y} + E_{5y}) \\ &= -0.5 \times dx \times (B_{5z} + B_{6z} - B_{6y} - B_{7y} - B_{7z} - B_{8z} + B_{8y} + B_{5y}) \end{aligned}$$

propagation of B on yface / yface curl



$$E_4 = E(0,1,1)$$

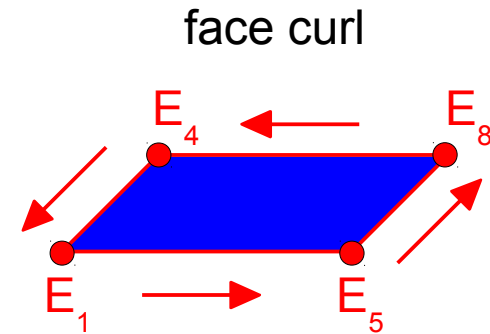
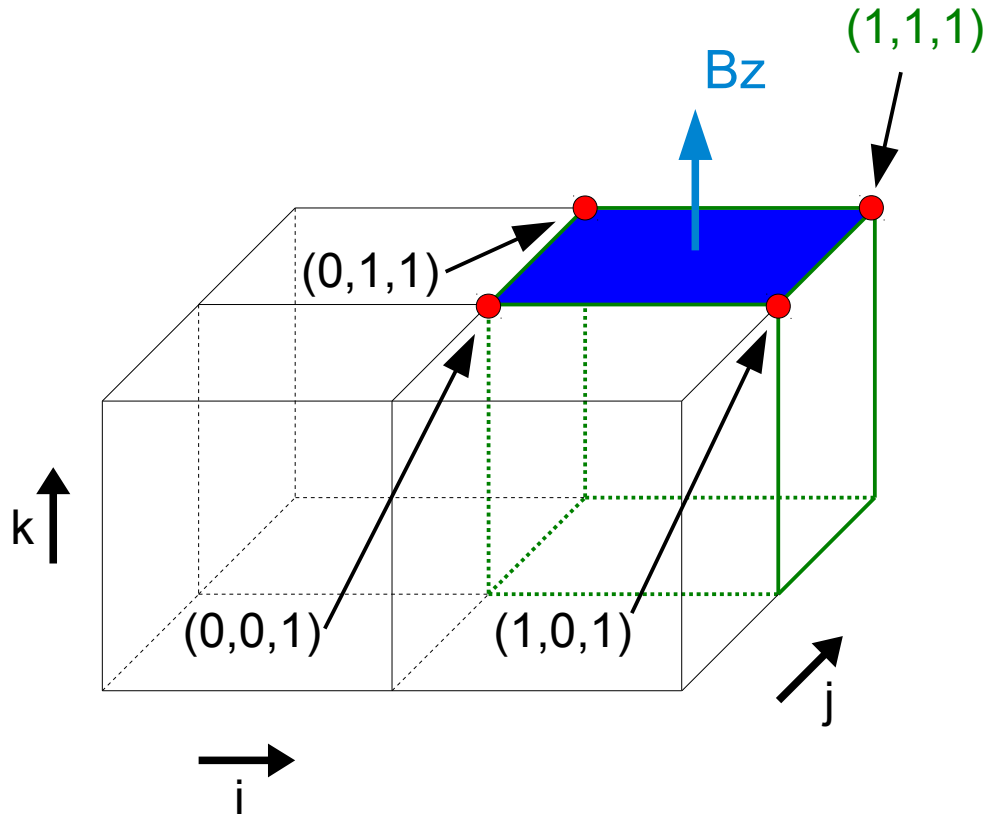
$$E_8 = E(1,1,1) = \text{local}$$

$$E_7 = E(1,1,0)$$

$$E_3 = E(0,1,0)$$

$$\begin{aligned} \frac{\partial(\oint d\mathbf{A}_y \cdot \mathbf{B})}{\partial t} &= -\oint d\mathbf{A}_y \cdot (\nabla \times \mathbf{E}) \\ \mathbf{J}_y &= (\nabla \times \mathbf{B})_y / \mu_0 \end{aligned} \quad \begin{aligned} &= 0.5 \times dx \times (-E_{4x} - E_{8x} + E_{8z} + E_{7z} + E_{7x} + E_{3x} - E_{3z} - E_{4z}) \\ &= -0.5 \times dx \times (-B_{4x} - B_{8x} + B_{8z} + B_{7z} + B_{7x} + B_{3x} - B_{3z} - B_{4z}) \end{aligned}$$

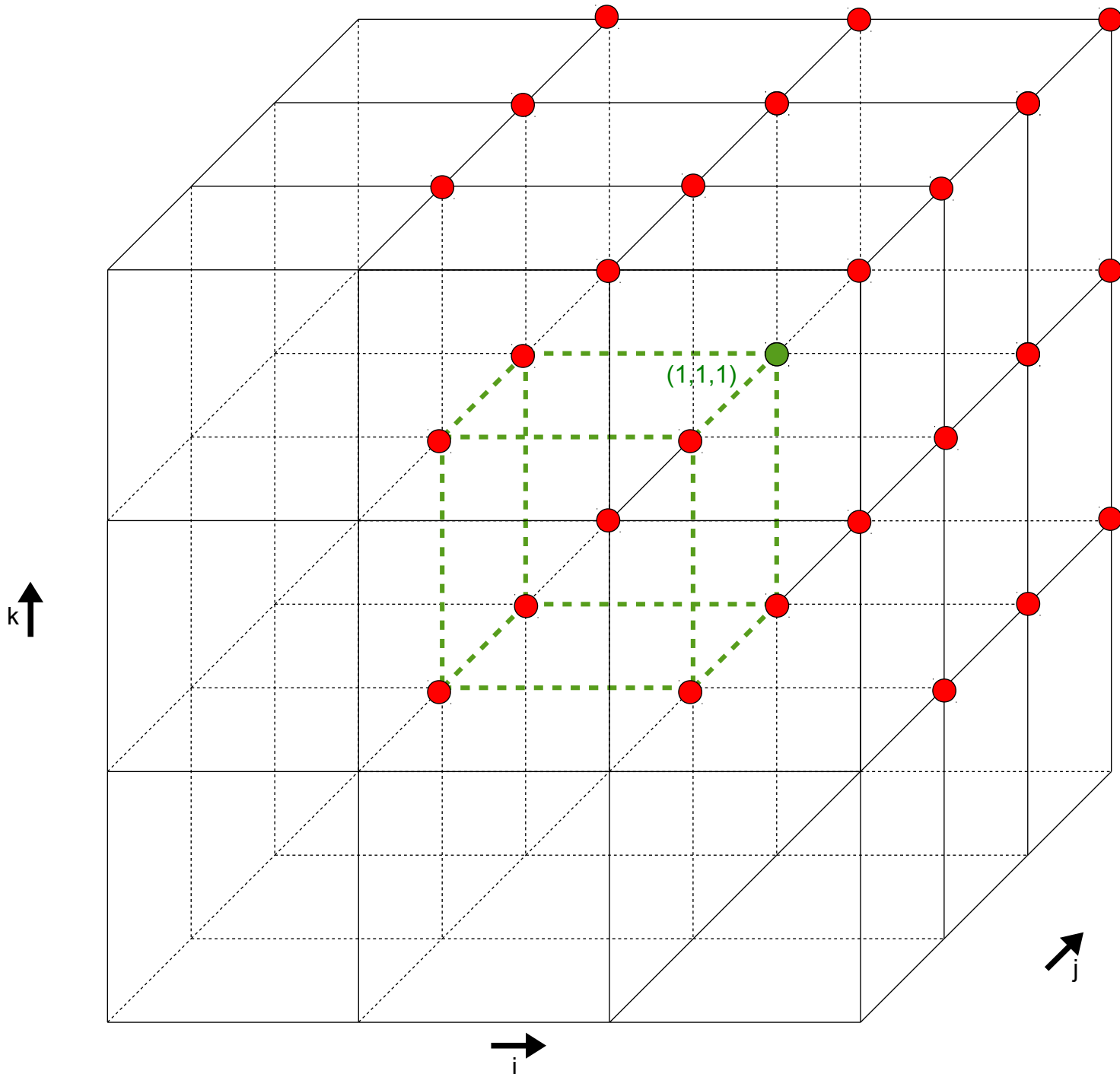
propagation of B on zface / zface curl



$$\begin{aligned} E_1 &= E(0,0,1) \\ E_5 &= E(1,0,1) \\ E_8 &= E(1,1,1) = \text{local} \\ E_4 &= E(0,1,1) \end{aligned}$$

$$\begin{aligned} \frac{\partial (\int d\mathbf{A}_z \cdot \mathbf{B})}{\partial t} &= - \int d\mathbf{A}_z \cdot (\nabla \times \mathbf{E}) \\ \mathbf{J}_z &= (\nabla \times \mathbf{B})_z / \mu_0 \end{aligned} \quad \begin{aligned} &= 0.5 \times dx \times (-E_{1x} - E_{5x} - E_{5y} - E_{8y} + E_{8x} + E_{4x} + E_{4y} + E_{1y}) \\ &= -0.5 \times dx \times (-B_{1x} - B_{5x} - B_{5y} - B_{8y} + B_{8x} + B_{4x} + B_{4y} + B_{1y}) \end{aligned}$$

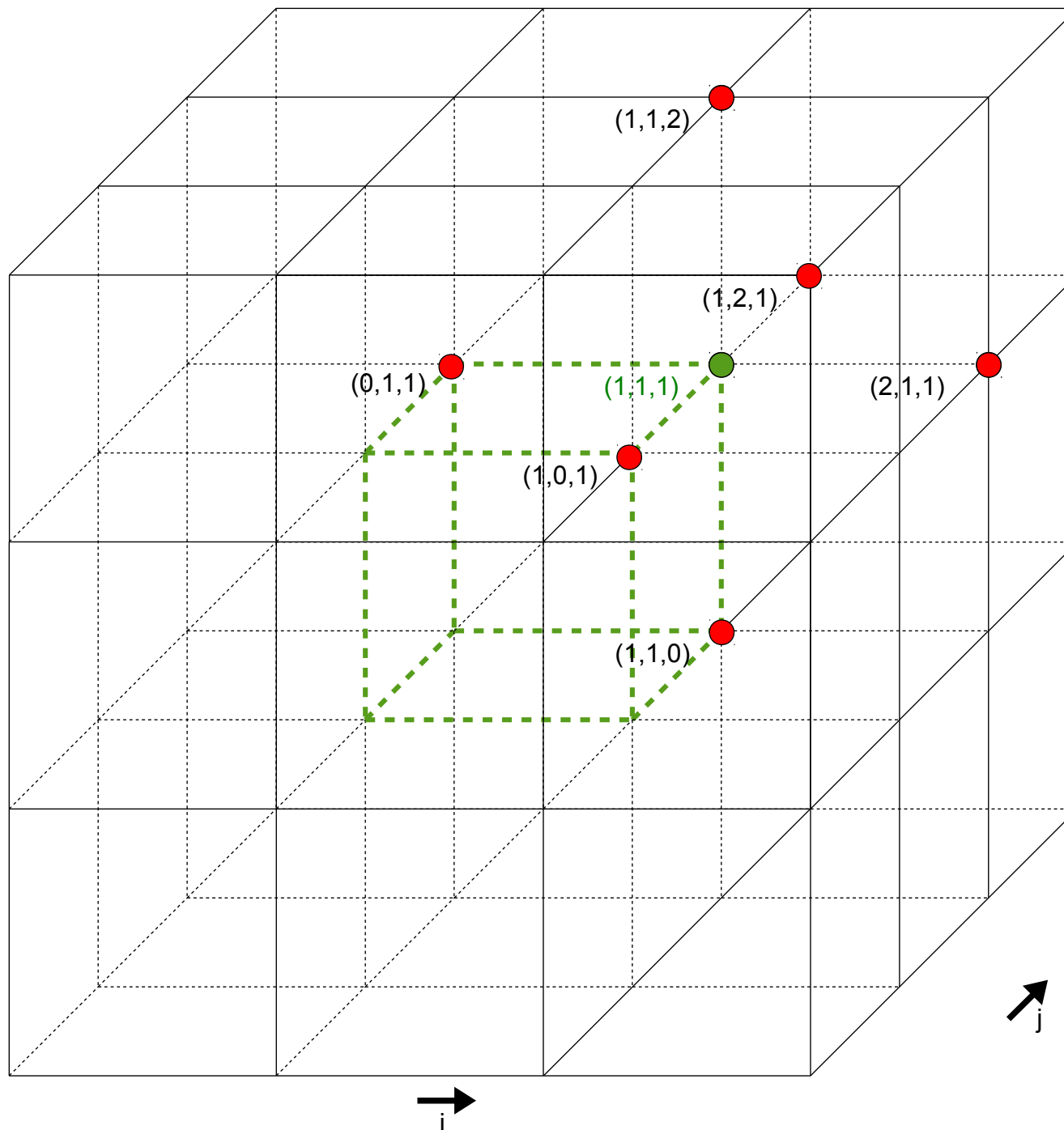
node2node average



```

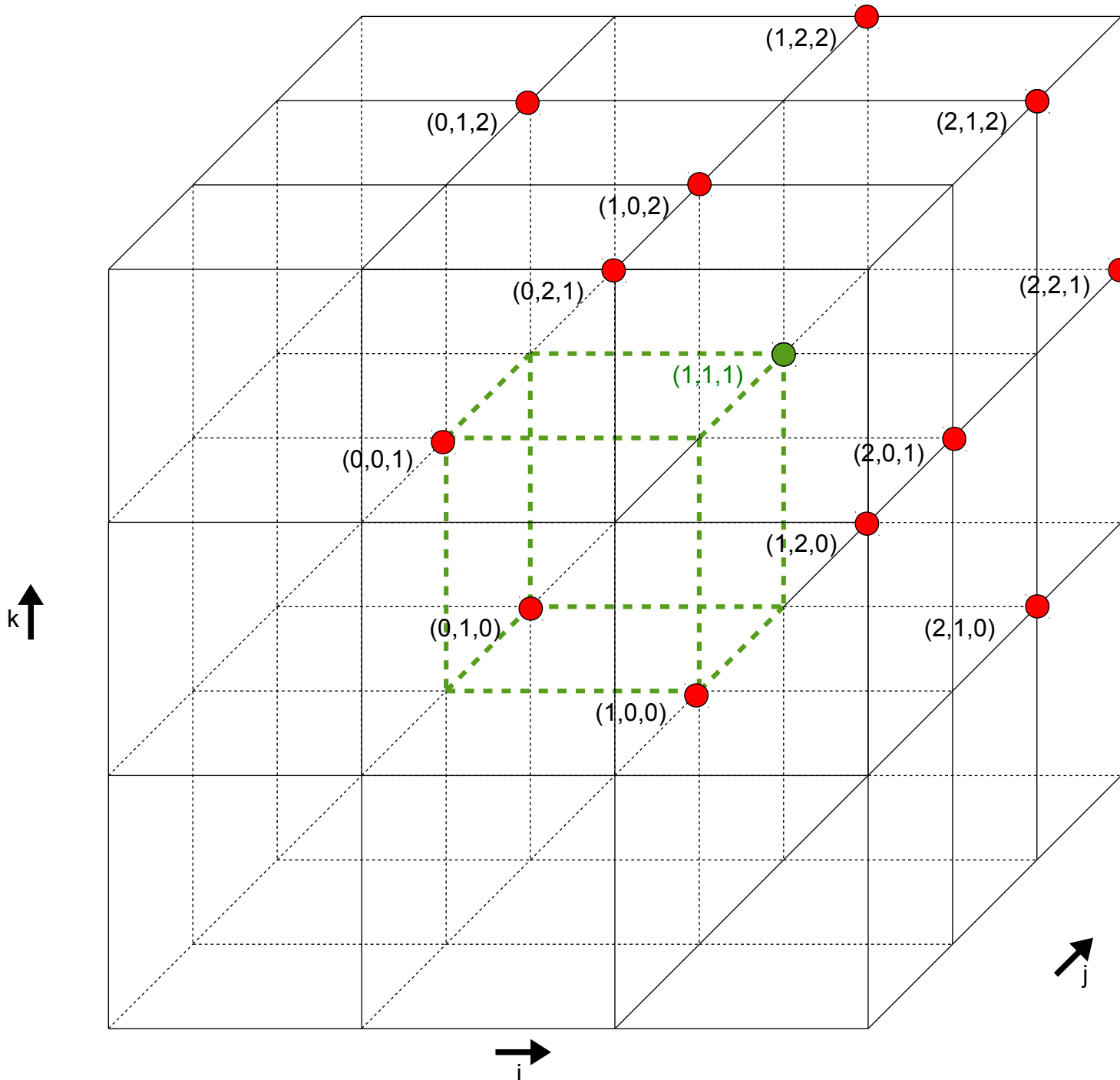
nodedata(1,1,1) =
C1*nodedata(1,1,1) +
C2*[ # direct neighbors
nodedata(1,1,0) +
nodedata(0,1,1) +
nodedata(1,2,1) +
nodedata(2,1,1) +
nodedata(1,0,1) +
nodedata(1,1,2)] +
C3*[ # diagonal sqrt(2)
nodedata(0,1,0) +
nodedata(1,0,0) +
nodedata(1,2,0) +
nodedata(2,1,0) +
nodedata(0,0,1) +
nodedata(0,2,1) +
nodedata(2,2,1) +
nodedata(2,0,1) +
nodedata(0,1,2) +
nodedata(1,2,2) +
nodedata(2,1,2) +
nodedata(1,0,2)] +
C4*[ # diagonal sqrt(3)
nodedata(0,0,2) +
nodedata(0,2,2) +
nodedata(2,0,2) +
nodedata(2,2,2) +
nodedata(0,0,0) +
nodedata(0,2,0) +
nodedata(2,0,0) +
nodedata(2,2,0)]
    
```

node2node average: six direct neighbor nodes



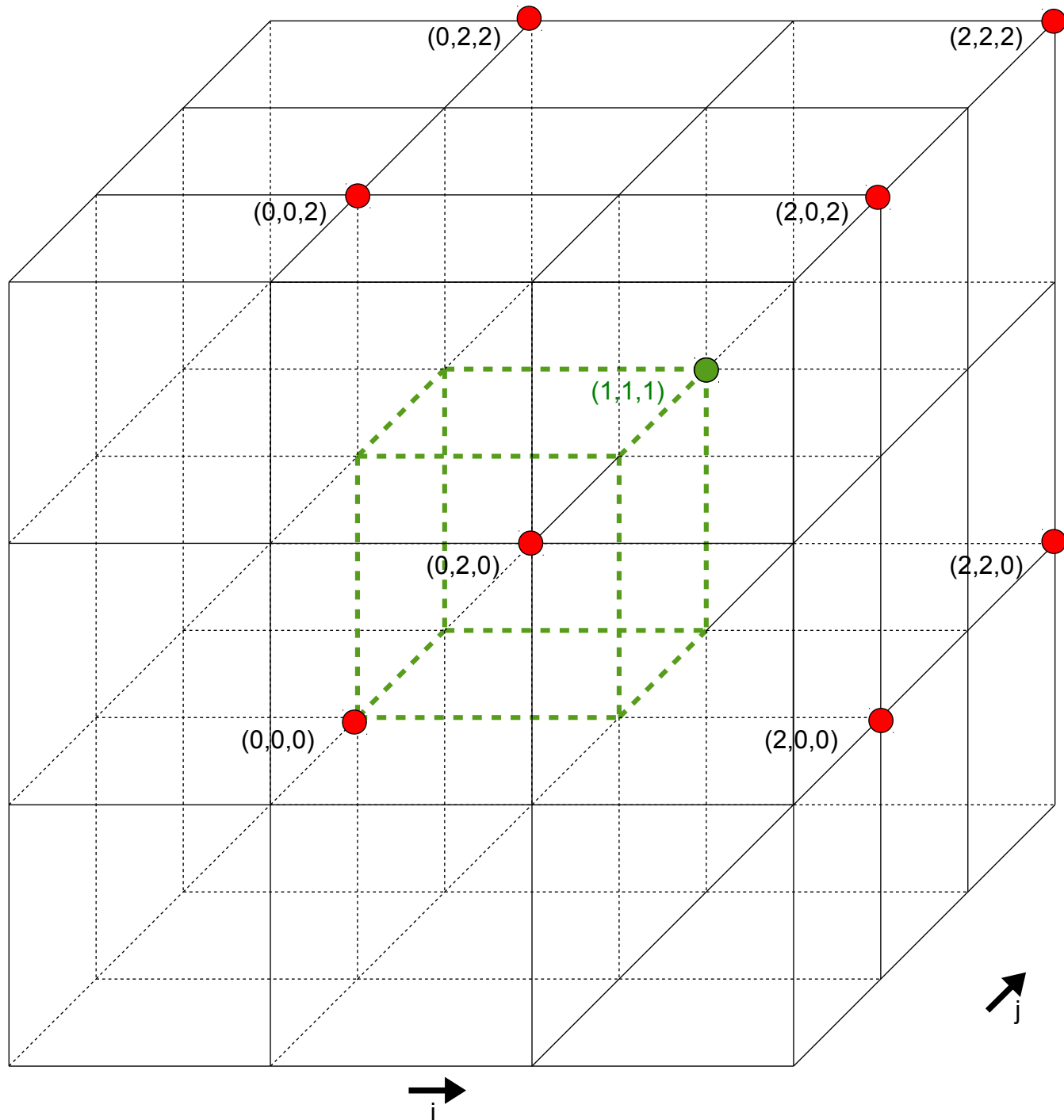
```
nodedata(1,1,1) =  
C1*nodedata(1,1,1) +  
C2*[ # direct neighbors  
nodedata(1,1,0) +  
nodedata(0,1,1) +  
nodedata(1,2,1) +  
nodedata(2,1,1) +  
nodedata(1,0,1) +  
nodedata(1,1,2)] +  
C3*[ # diagonal sqrt(2)  
nodedata(0,1,0) +  
nodedata(1,0,0) +  
nodedata(1,2,0) +  
nodedata(2,1,0) +  
nodedata(0,0,1) +  
nodedata(0,2,1) +  
nodedata(2,2,1) +  
nodedata(2,0,1) +  
nodedata(0,1,2) +  
nodedata(1,2,2) +  
nodedata(2,1,2) +  
nodedata(1,0,2)] +  
C4*[ # diagonal sqrt(3)  
nodedata(0,0,2) +  
nodedata(0,2,2) +  
nodedata(2,0,2) +  
nodedata(2,2,2) +  
nodedata(0,0,0) +  
nodedata(0,2,0) +  
nodedata(2,0,0) +  
nodedata(2,2,0)]
```

node2node average: twelve sqrt(2) diagonal neighbor nodes



```
nodedata(1,1,1) =  
C1*nodedata(1,1,1) +  
C2*[ # direct neighbors  
nodedata(1,1,0) +  
nodedata(0,1,1) +  
nodedata(1,2,1) +  
nodedata(2,1,1) +  
nodedata(1,0,1) +  
nodedata(1,1,2)] +  
C3*[ # diagonal sqrt(2)  
nodedata(0,1,0) +  
nodedata(1,0,0) +  
nodedata(1,2,0) +  
nodedata(2,1,0) +  
nodedata(0,0,1) +  
nodedata(0,2,1) +  
nodedata(2,2,1) +  
nodedata(2,0,1) +  
nodedata(0,1,2) +  
nodedata(1,2,2) +  
nodedata(2,1,2) +  
nodedata(1,0,2)] +  
C4*[ # diagonal sqrt(3)  
nodedata(0,0,2) +  
nodedata(0,2,2) +  
nodedata(2,0,2) +  
nodedata(2,2,2) +  
nodedata(0,0,0) +  
nodedata(0,2,0) +  
nodedata(2,0,0) +  
nodedata(2,2,0)]
```

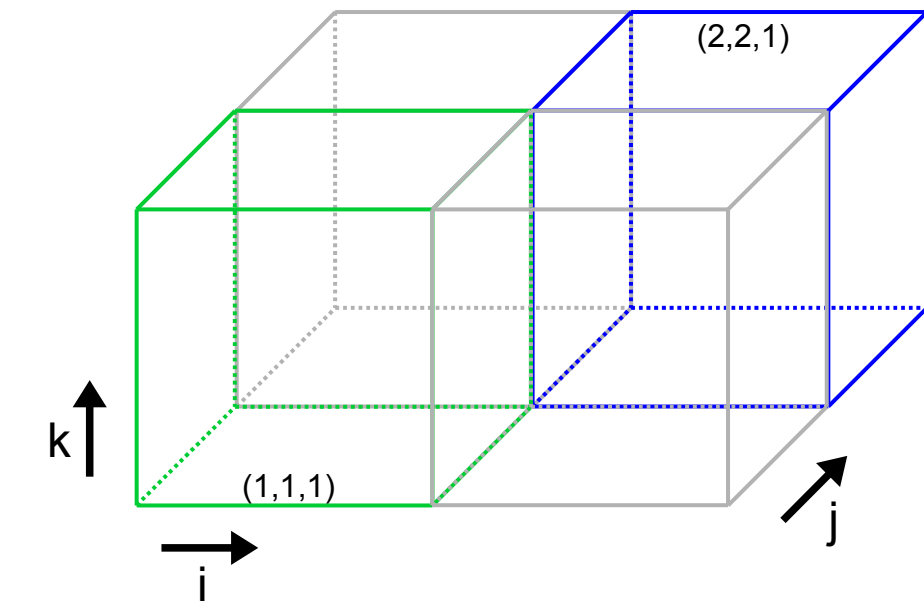
node2node average: eight $\sqrt{3}$ diagonal neighbor nodes



```
nodedata(1,1,1) =  
C1*nodata(1,1,1) +  
C2*[ # direct neighbors  
nodedata(1,1,0) +  
nodedata(0,1,1) +  
nodedata(1,2,1) +  
nodedata(2,1,1) +  
nodedata(1,0,1) +  
nodedata(1,1,2)] +  
C3*[ # diagonal sqrt(2)  
nodedata(0,1,0) +  
nodedata(1,0,0) +  
nodedata(1,2,0) +  
nodedata(2,1,0) +  
nodedata(0,0,1) +  
nodedata(0,2,1) +  
nodedata(2,2,1) +  
nodedata(2,0,1) +  
nodedata(0,1,2) +  
nodedata(1,2,2) +  
nodedata(2,1,2) +  
nodedata(1,0,2)] +  
C4*[ # diagonal sqrt(3)  
nodedata(0,0,2) +  
nodedata(0,2,2) +  
nodedata(2,0,2) +  
nodedata(2,2,2) +  
nodedata(0,0,0) +  
nodedata(0,2,0) +  
nodedata(2,0,0) +  
nodedata(2,2,0)]
```

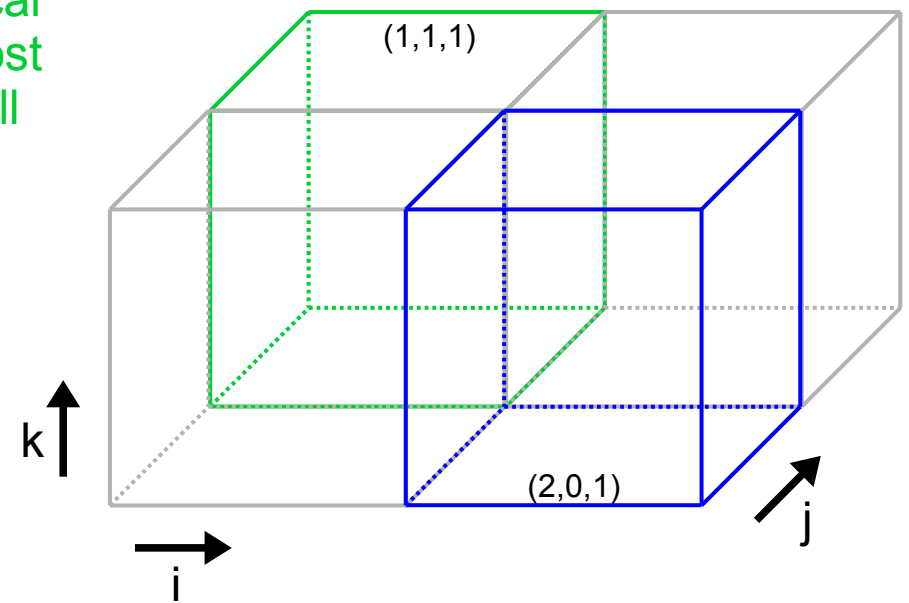
Neumann boundaries for edge cells

$(-x, -y)$ edge



`celldata(1,1,1) = celldata(2,2,1)`

$(-x, +y)$ edge



`celldata(1,1,1) = celldata(2,0,1)`

Ghost
cells

Cells in the
simulation
domain

Local
ghost
cell

Neumann boundaries for edge cells

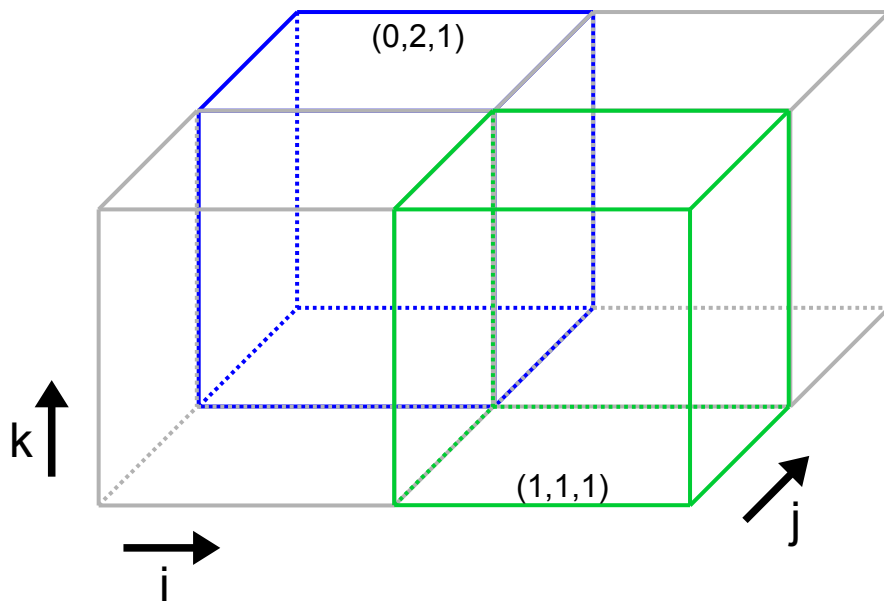
(+x,-y) edge

(+x,+y) edge

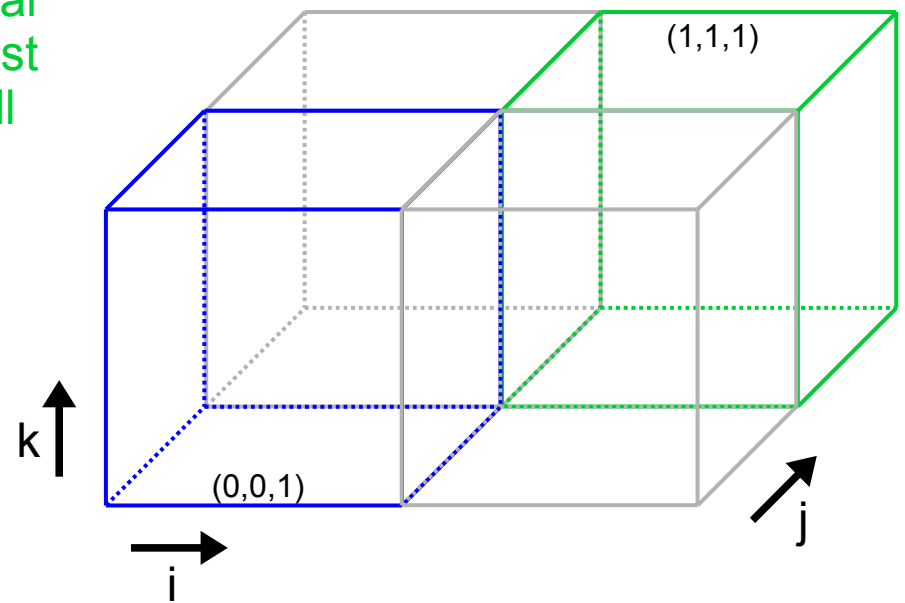
Ghost
cells

Cells in the
simulation
domain

Local
ghost
cell



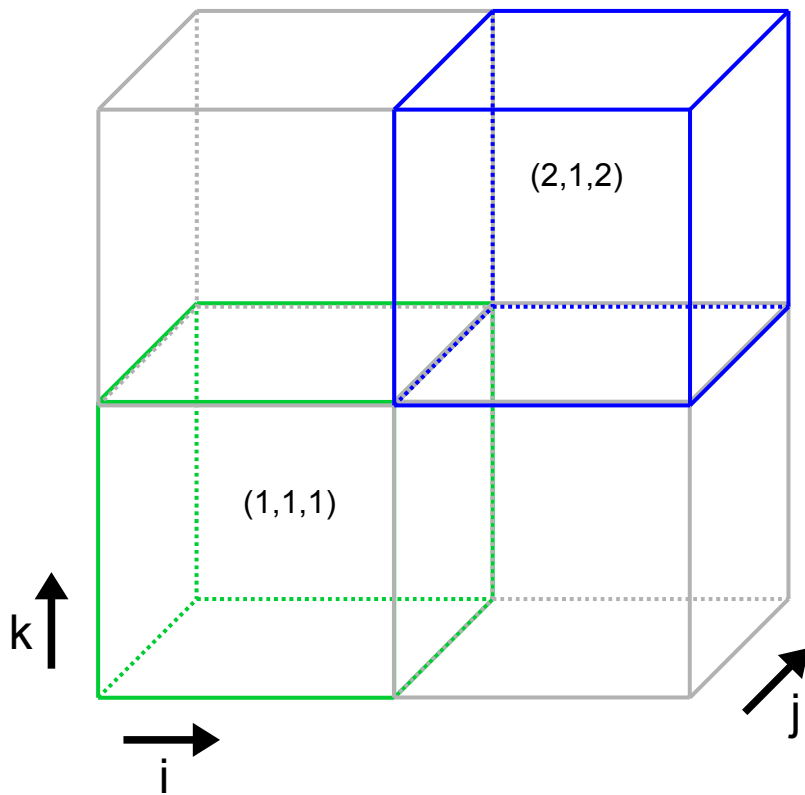
`celldata(1,1,1) = celldata(0,2,1)`



`celldata(1,1,1) = celldata(0,0,1)`

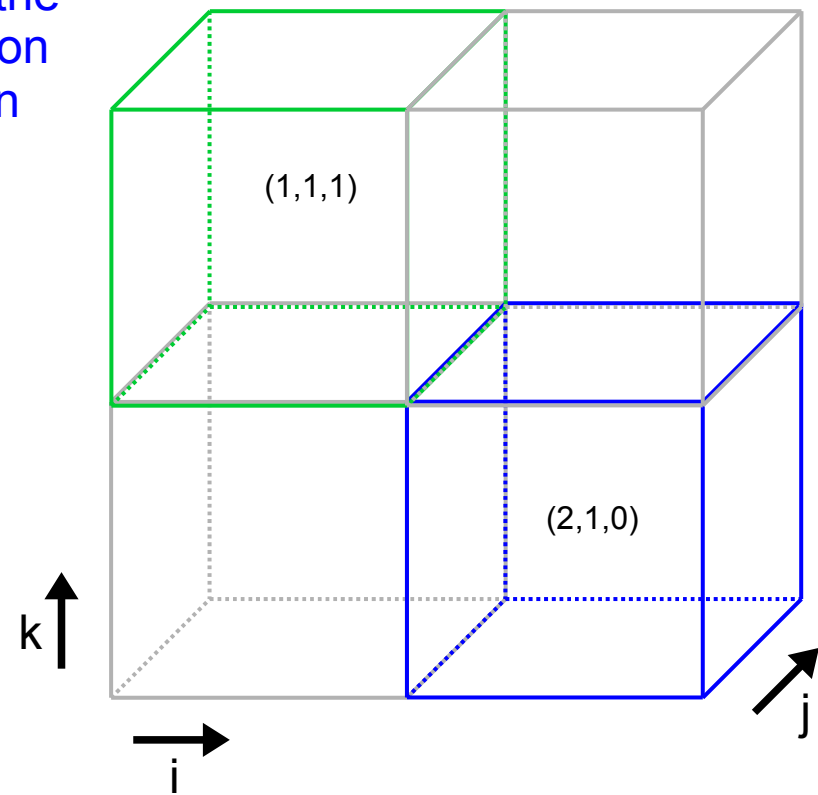
Neumann boundaries for edge cells

$(-x, -z)$ edge



`celldata(1,1,1) = celldata(2,1,2)`

$(-x, +z)$ edge



`celldata(1,1,1) = celldata(2,1,0)`

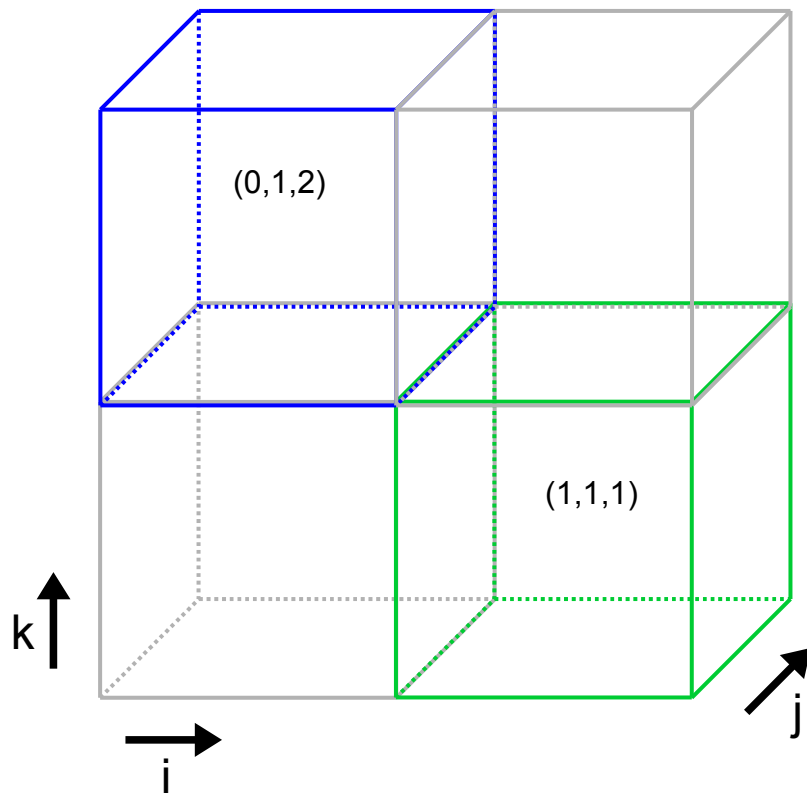
Ghost
cells

Cells in the
simulation
domain

Local
ghost
cell

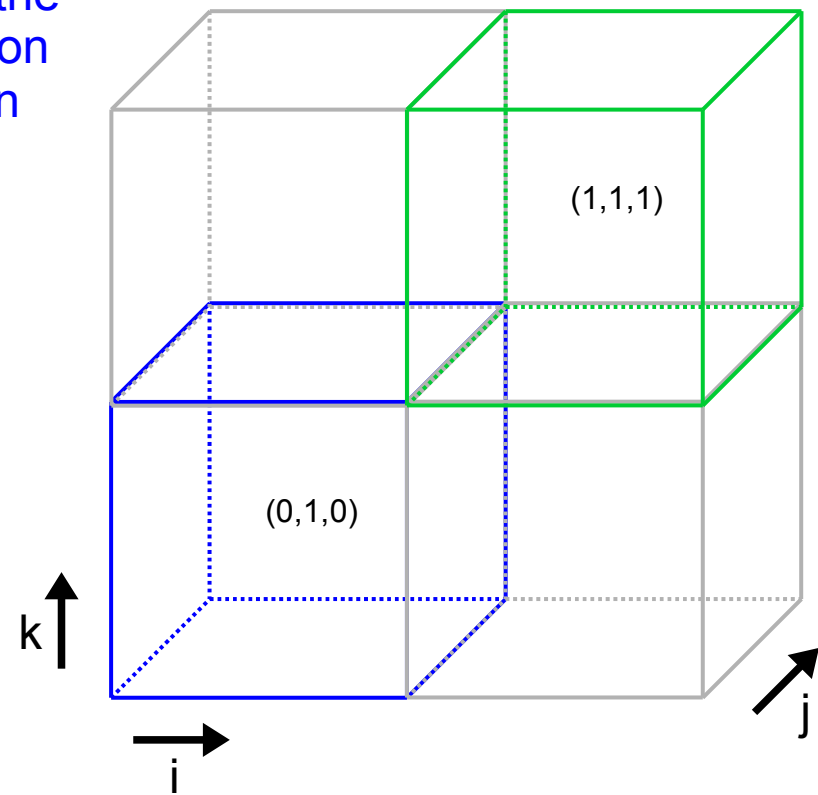
Neumann boundaries for edge cells

(+x,-z) edge



`celldata(1,1,1) = celldata(0,1,2)`

(+x,+z) edge



`celldata(1,1,1) = celldata(0,1,0)`

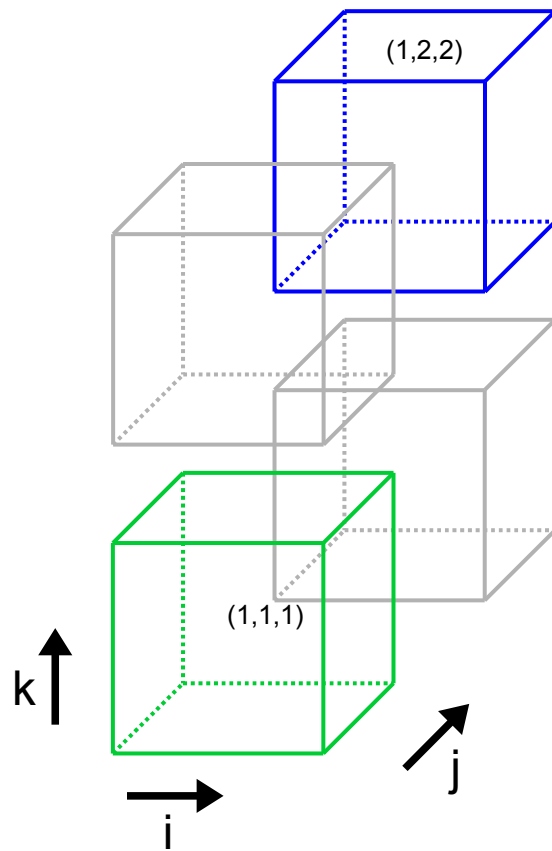
Ghost
cells

Cells in the
simulation
domain

Local
ghost
cell

Neumann boundaries for edge cells

$(-y, -z)$ edge

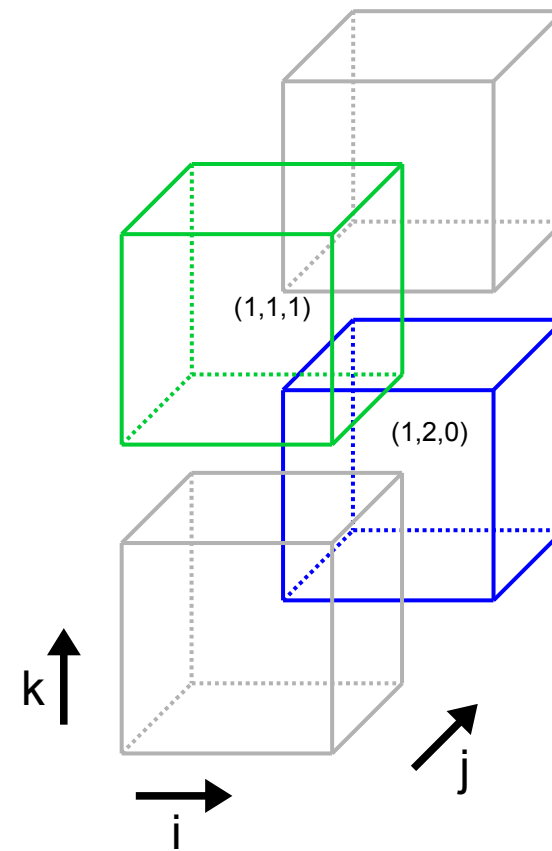


`celldata(1,1,1) = celldata(1,2,2)`

$(-y, +z)$ edge

Ghost cells
Cells in the simulation domain

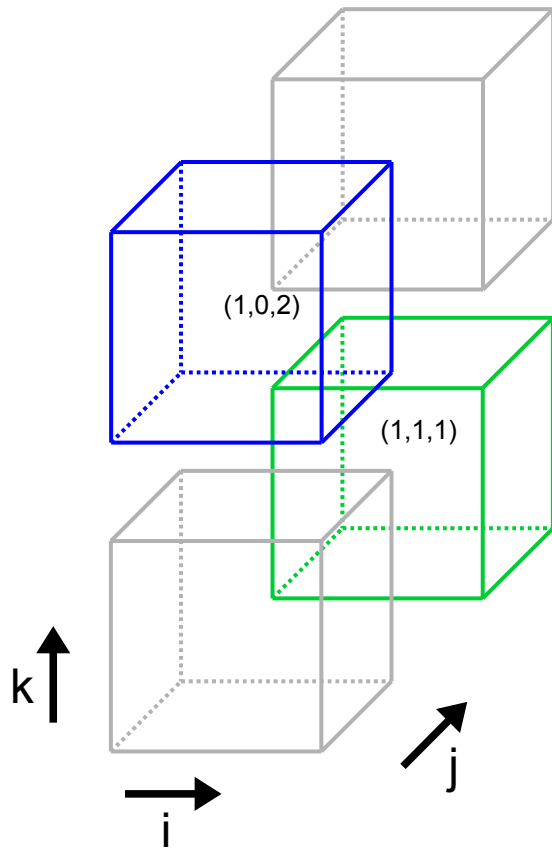
Local ghost cell



`celldata(1,1,1) = celldata(1,2,0)`

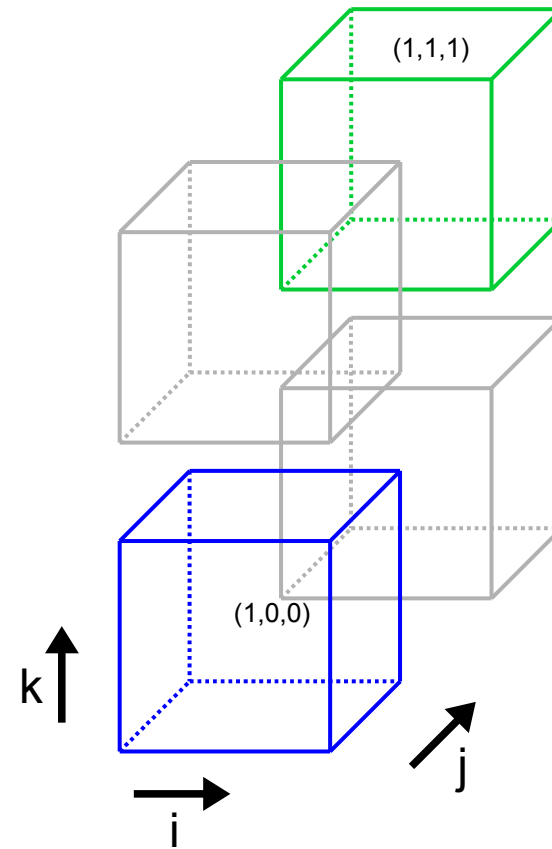
Neumann boundaries for edge cells

(+y,-z) edge



```
celldata(1,1,1) = celldata(1,0,2)
```

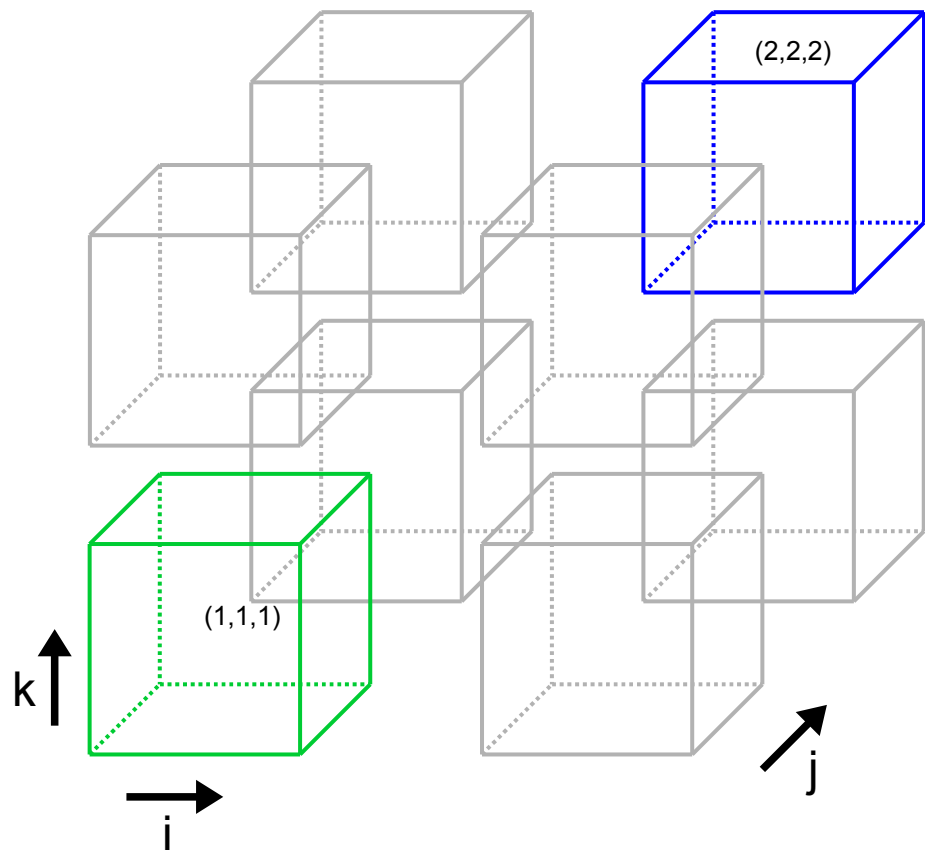
(+y,+z) edge



```
celldata(1,1,1) = celldata(1,0,0)
```

Neumann boundaries for corner cells

$(-x, -y, -z)$ corner



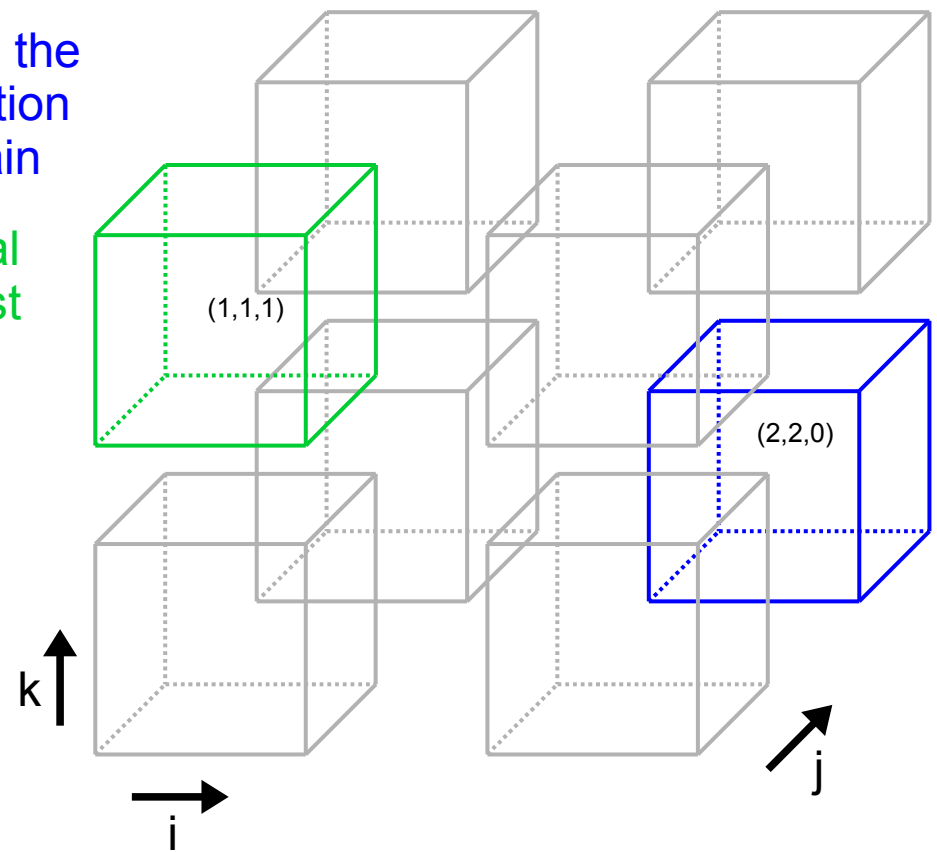
$$\text{celldata}(1,1,1) = \text{celldata}(2,2,2)$$

Ghost
cells

Cells in the
simulation
domain

Local
ghost
cell

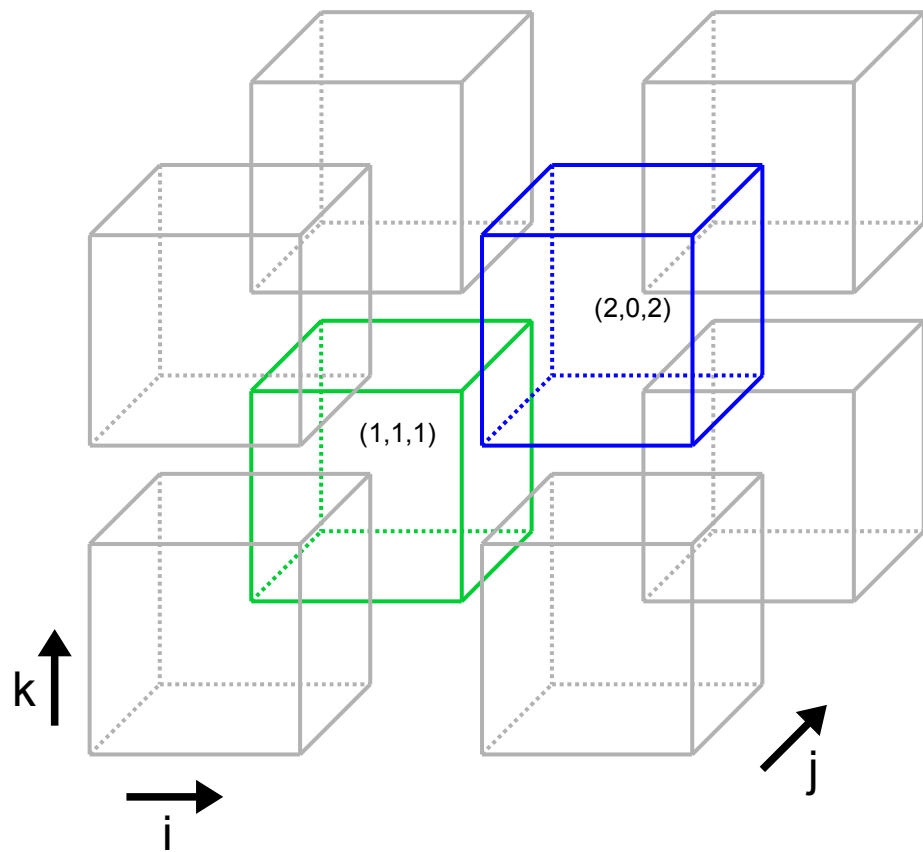
$(-x, -y, +z)$ corner



$$\text{celldata}(1,1,1) = \text{celldata}(2,2,0)$$

Neumann boundaries for corner cells

$(-x, +y, -z)$ corner



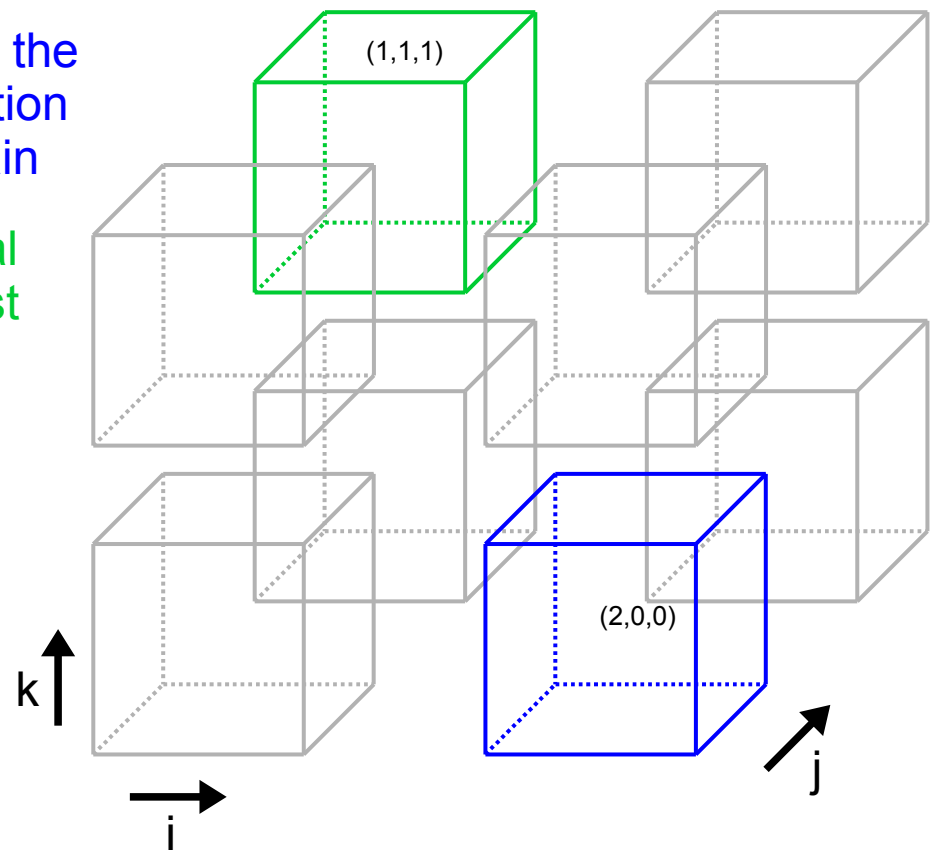
`celldata(1,1,1) = celldata(2,0,2)`

$(-x, +y, +z)$ corner

Ghost cells

Cells in the simulation domain

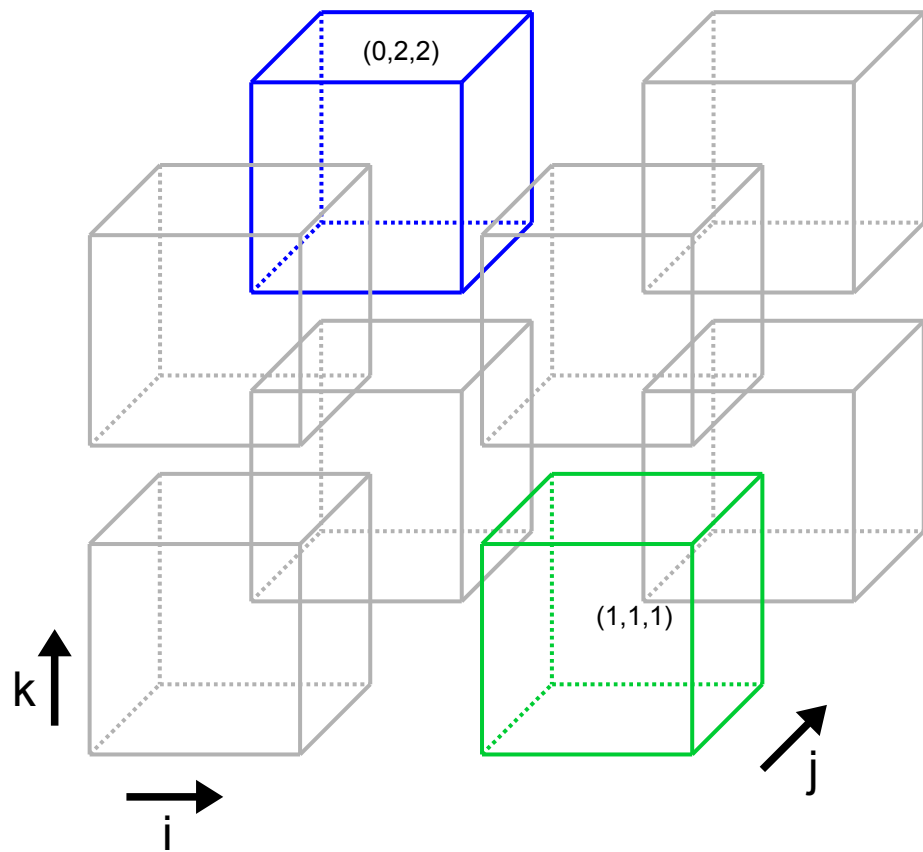
Local ghost cell



`celldata(1,1,1) = celldata(2,0,0)`

Neumann boundaries for corner cells

(+x,-y,-z) corner



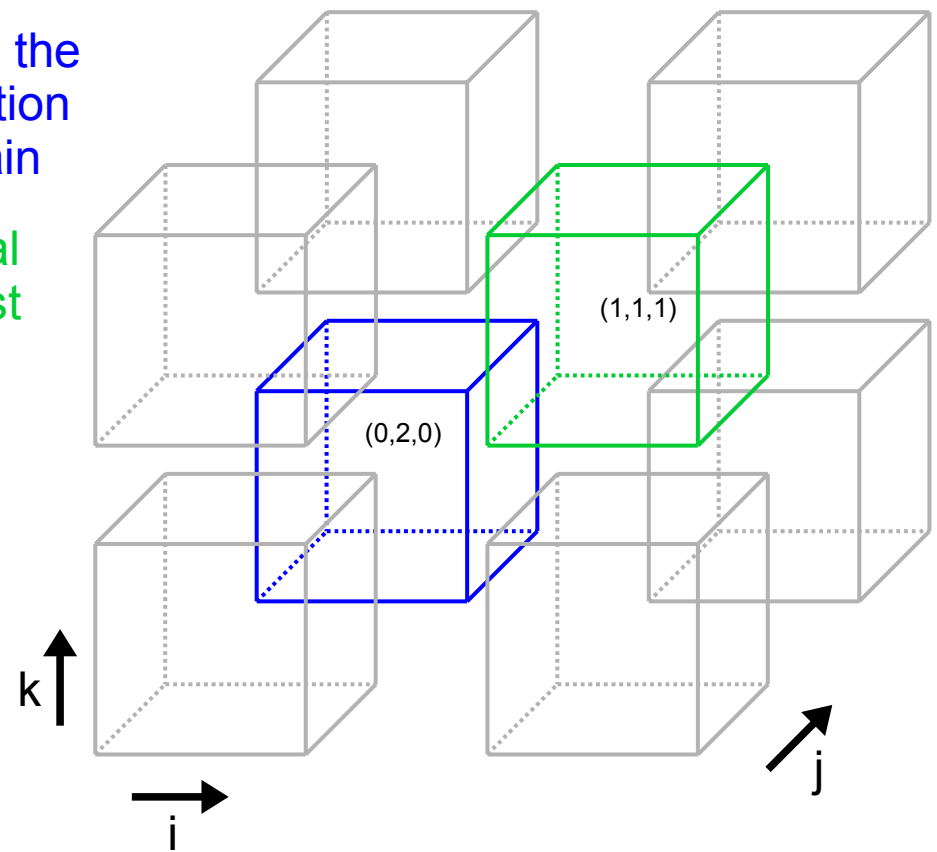
`celldata(1,1,1) = celldata(0,2,2)`

Ghost
cells

Cells in the
simulation
domain

Local
ghost
cell

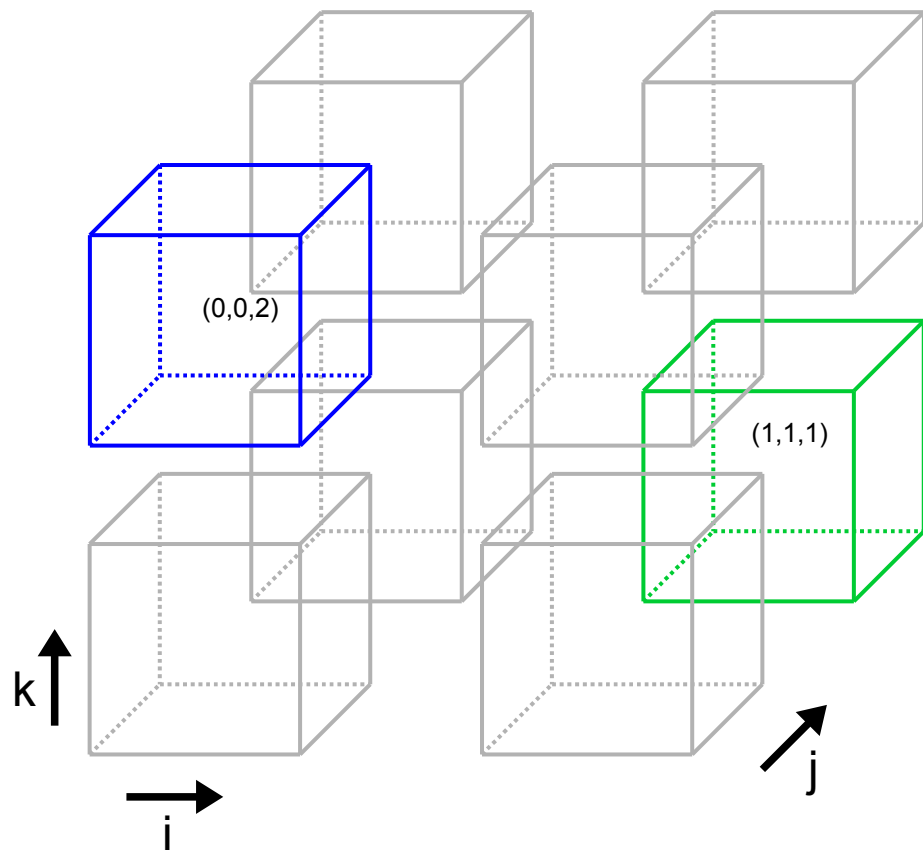
(+x,-y,+z) corner



`celldata(1,1,1) = celldata(0,2,0)`

Neumann boundaries for corner cells

(+x,+y,-z) corner



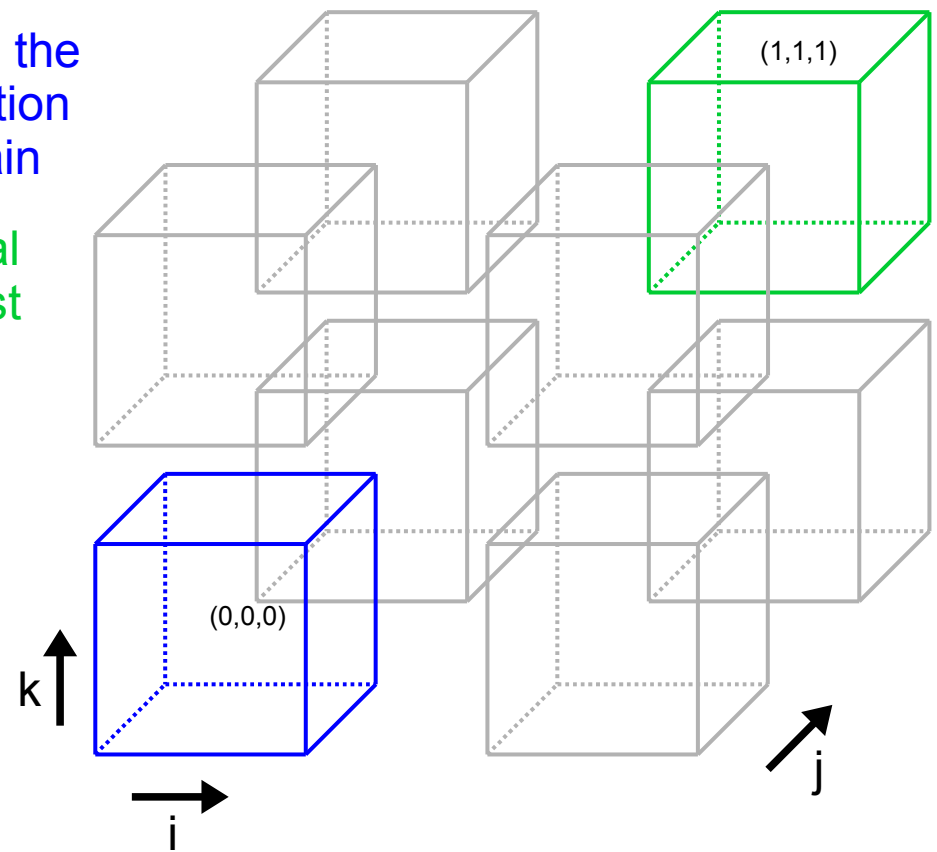
$$\text{celldata}(1,1,1) = \text{celldata}(0,0,2)$$

Ghost cells

Cells in the simulation domain

Local ghost cell

(+x,+y,+z) corner



$$\text{celldata}(1,1,1) = \text{celldata}(0,0,0)$$

Neumann boundaries for edge faces

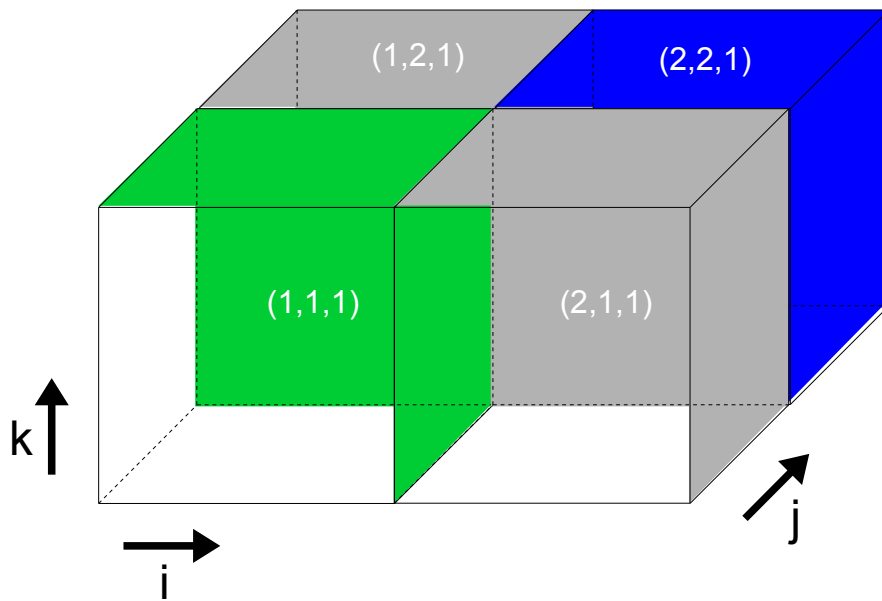
$(-x, -y)$ edge

$(-x, +y)$ edge

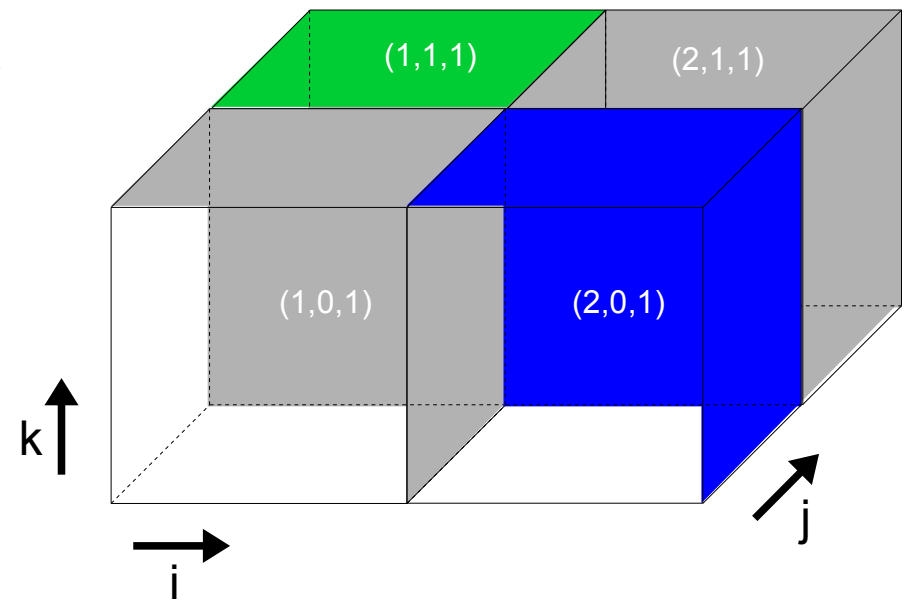
Ghost cells

Cells in the simulation domain

Local ghost cell



$\text{facedata}_x(1,1,1) = \text{facedata}_x(1,2,1)$
 $\text{facedata}_y(1,1,1) = \text{facedata}_y(2,1,1)$
 $\text{facedata}_z(1,1,1) = \text{facedata}_z(2,2,1)$



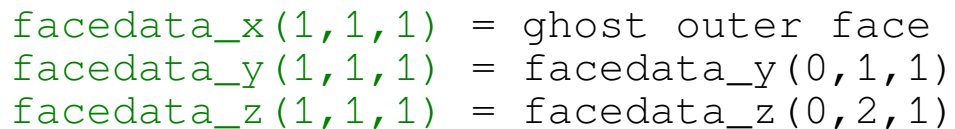
$\text{facedata}_x(1,1,1) = \text{facedata}_x(1,0,1)$
 $\text{facedata}_y(1,1,1) = \text{ghost outer face}$
 $\text{facedata}_z(1,1,1) = \text{facedata}_y(2,0,1)$

(+x,-y) edge

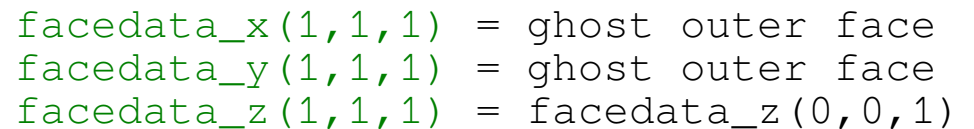
Ghost cells

Cells in the simulation domain

Local
ghost
cell

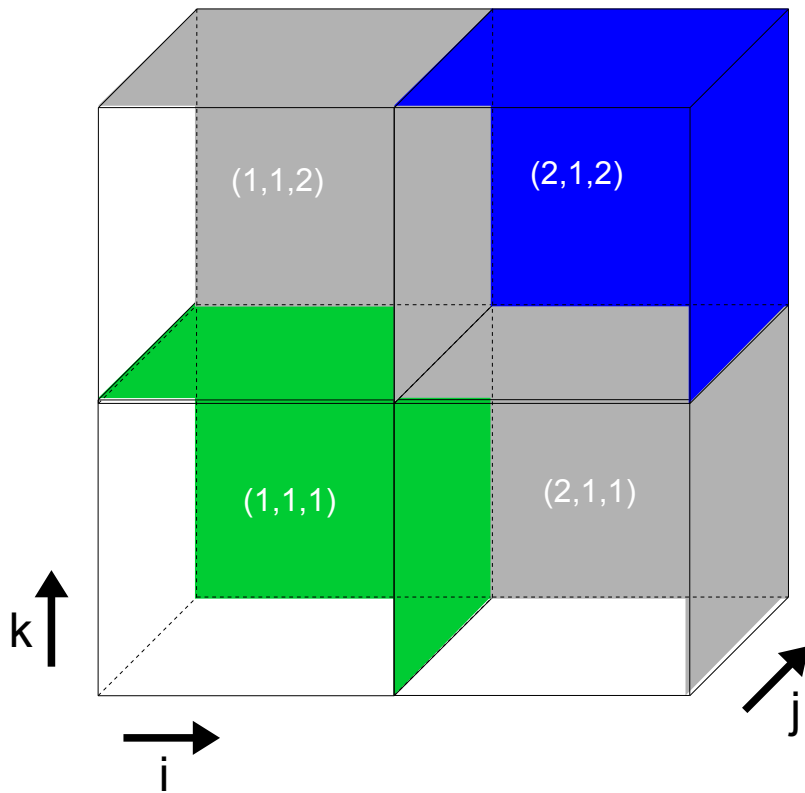


the
on
n



Neumann boundaries for edge faces

$(-x, -z)$ edge



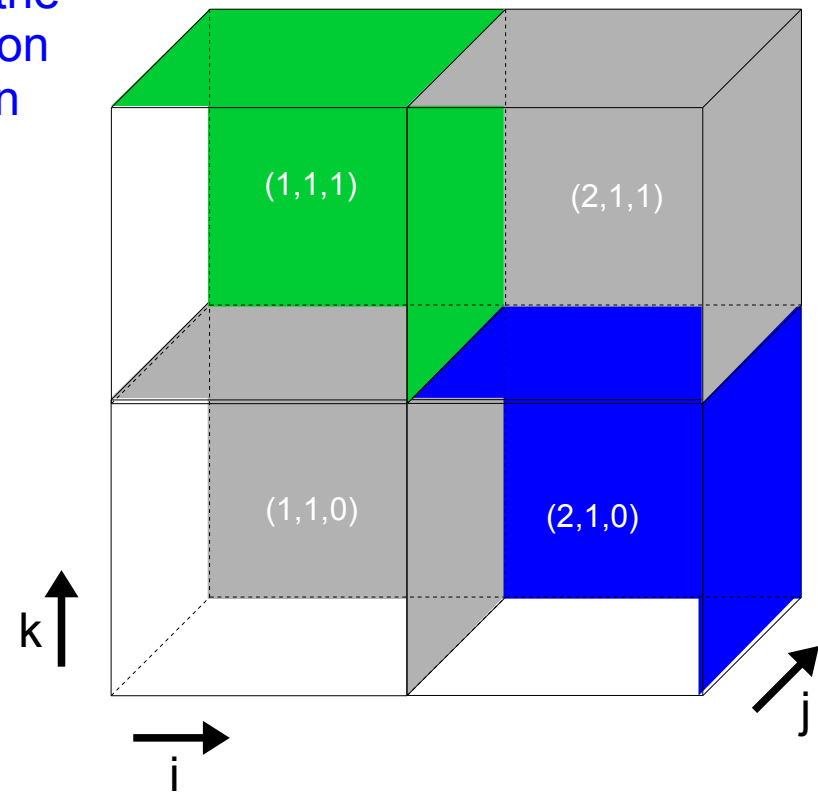
$\text{facedata}_x(1,1,1) = \text{facedata}_x(1,1,2)$
 $\text{facedata}_y(1,1,1) = \text{facedata}_y(2,1,2)$
 $\text{facedata}_z(1,1,1) = \text{facedata}_z(2,1,1)$

Ghost cells

Cells in the simulation domain

Local ghost cell

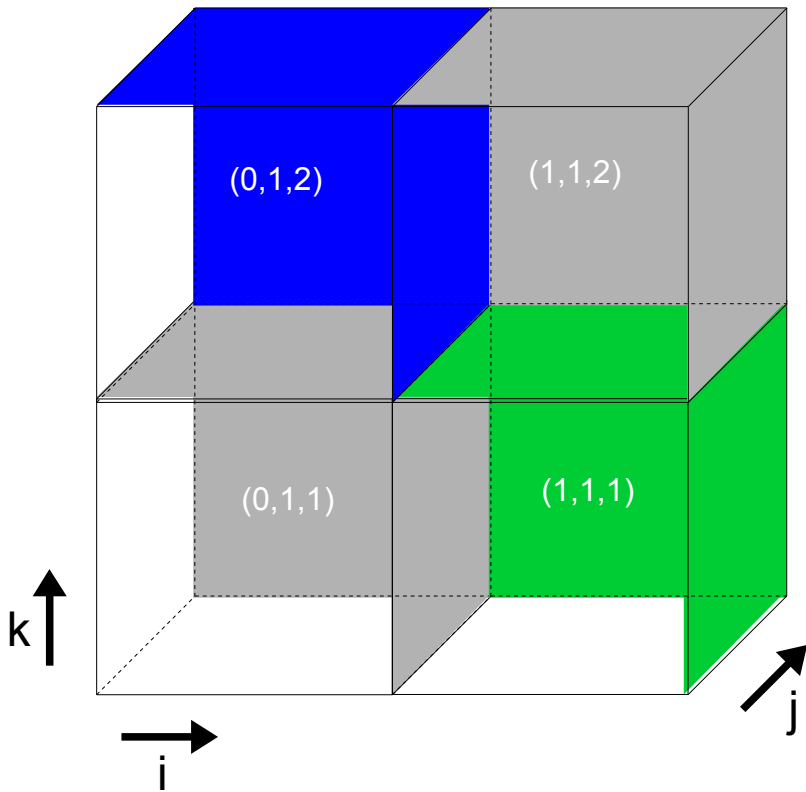
$(-x, +z)$ edge



$\text{facedata}_x(1,1,1) = \text{facedata}_x(1,1,0)$
 $\text{facedata}_y(1,1,1) = \text{facedata}_y(2,1,0)$
 $\text{facedata}_z(1,1,1) = \text{ghost outer face}$

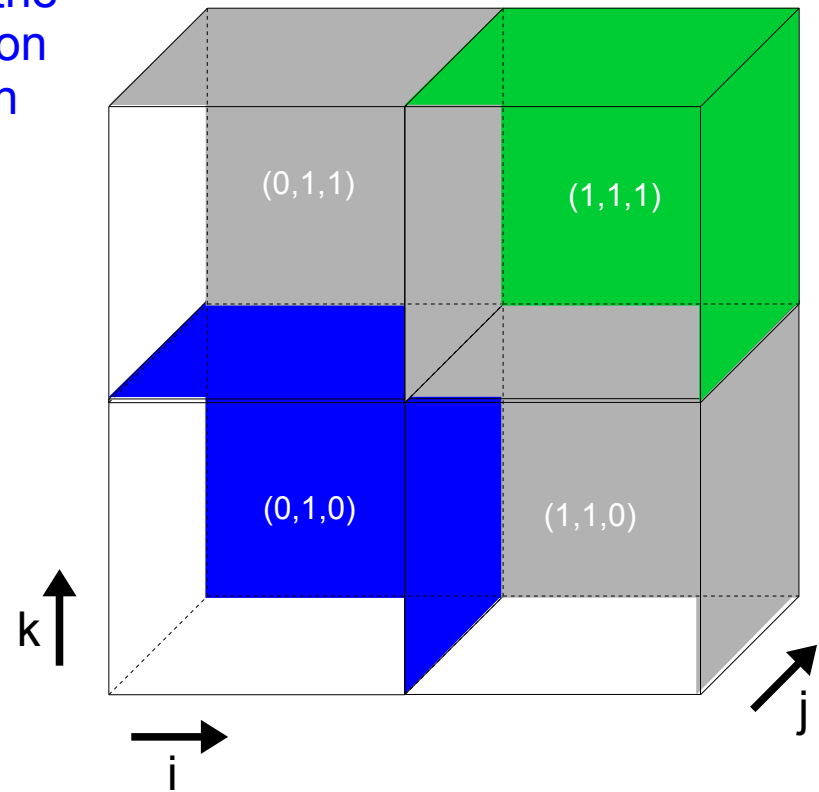
Neumann boundaries for edge faces

(+x,-z) edge



```
facedata_x(1,1,1) = ghost outer face
facedata_y(1,1,1) = facedata_y(0,1,2)
facedata_z(1,1,1) = facedata_z(0,1,1)
```

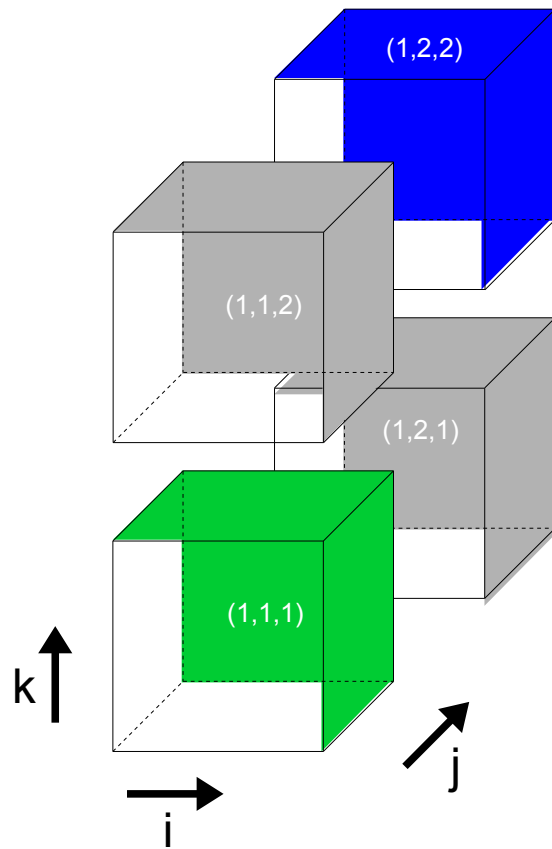
(+x, +z) edge



```
facedata_x(1,1,1) = ghost outer face
facedata_y(1,1,1) = facedata_y(0,1,0)
facedata_z(1,1,1) = ghost outer face
```

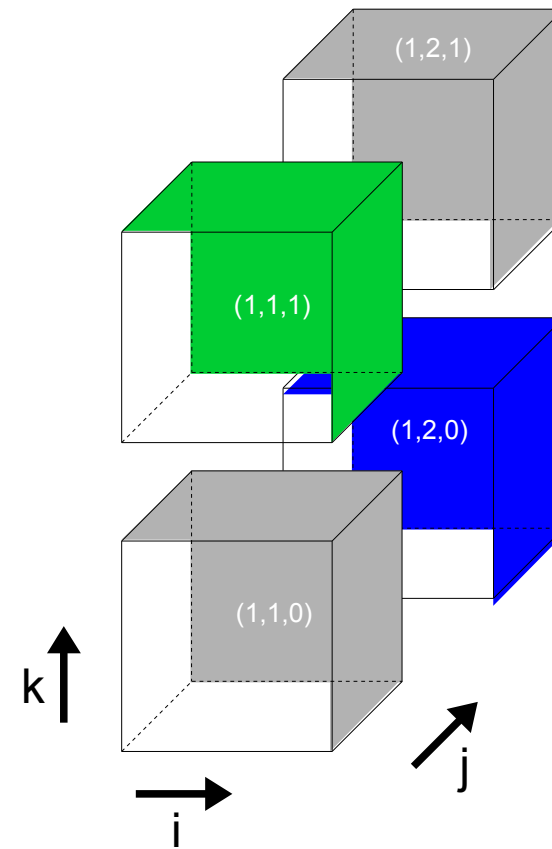
Neumann boundaries for edge faces

$(-y, -z)$ edge



$\text{facedata}_x(1,1,1) = \text{facedata}_x(1,2,2)$
 $\text{facedata}_y(1,1,1) = \text{facedata}_y(1,1,2)$
 $\text{facedata}_z(1,1,1) = \text{facedata}_z(1,2,1)$

$(-y, +z)$ edge



$\text{facedata}_x(1,1,1) = \text{facedata}_x(1,2,0)$
 $\text{facedata}_y(1,1,1) = \text{facedata}_y(1,1,0)$
 $\text{facedata}_z(1,1,1) = \text{ghost outer face}$

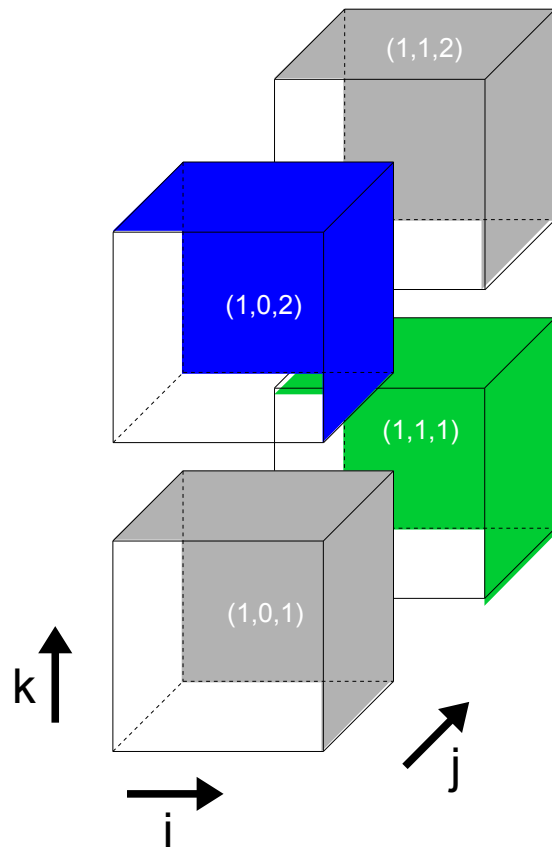
Ghost
cells

Cells in the
simulation
domain

Local
ghost
cell

Neumann boundaries for edge faces

(+y,-z) corner



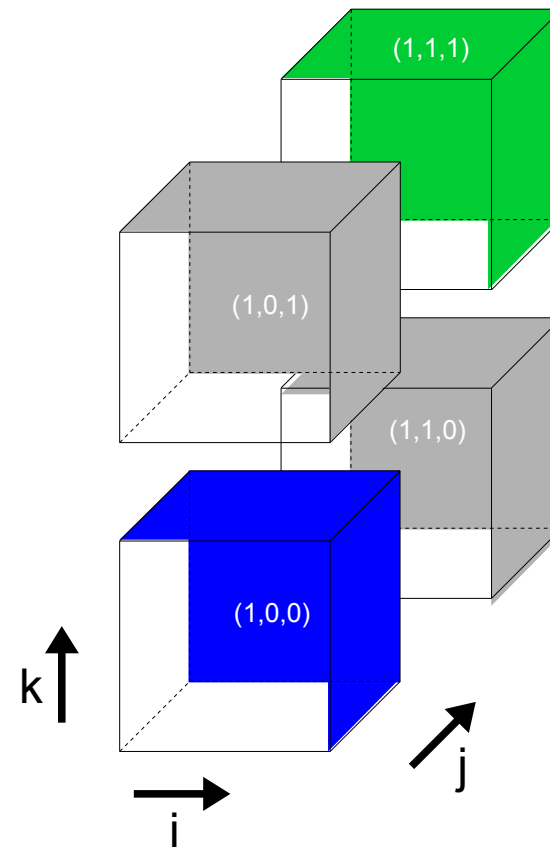
`facedata_x(1,1,1) = facedata_x(1,0,2)`
`facedata_y(1,1,1) = ghost outer face`
`facedata_z(1,1,1) = facedata_z(1,0,1)`

Ghost
cells

Cells in the
simulation
domain

Local
ghost
cell

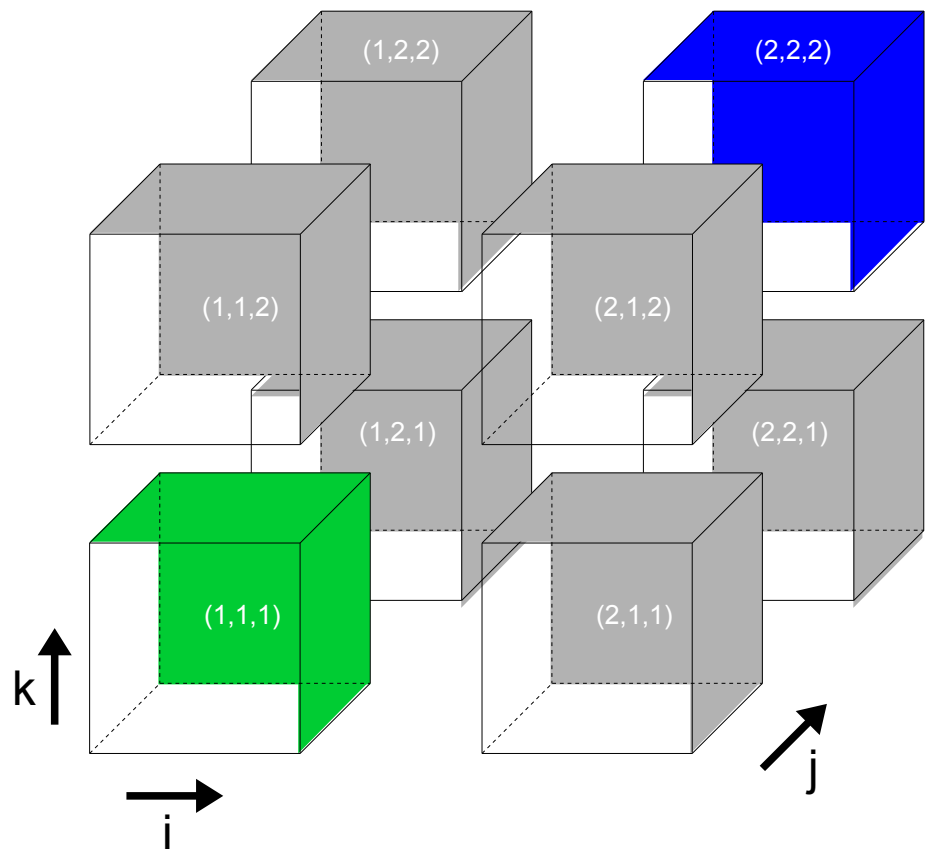
(+y,+z) edge



`facedata_x(1,1,1) = facedata_x(1,0,0)`
`facedata_y(1,1,1) = ghost outer face`
`facedata_z(1,1,1) = ghost outer face`

Neumann boundaries for corner faces

$(-x, -y, -z)$ corner



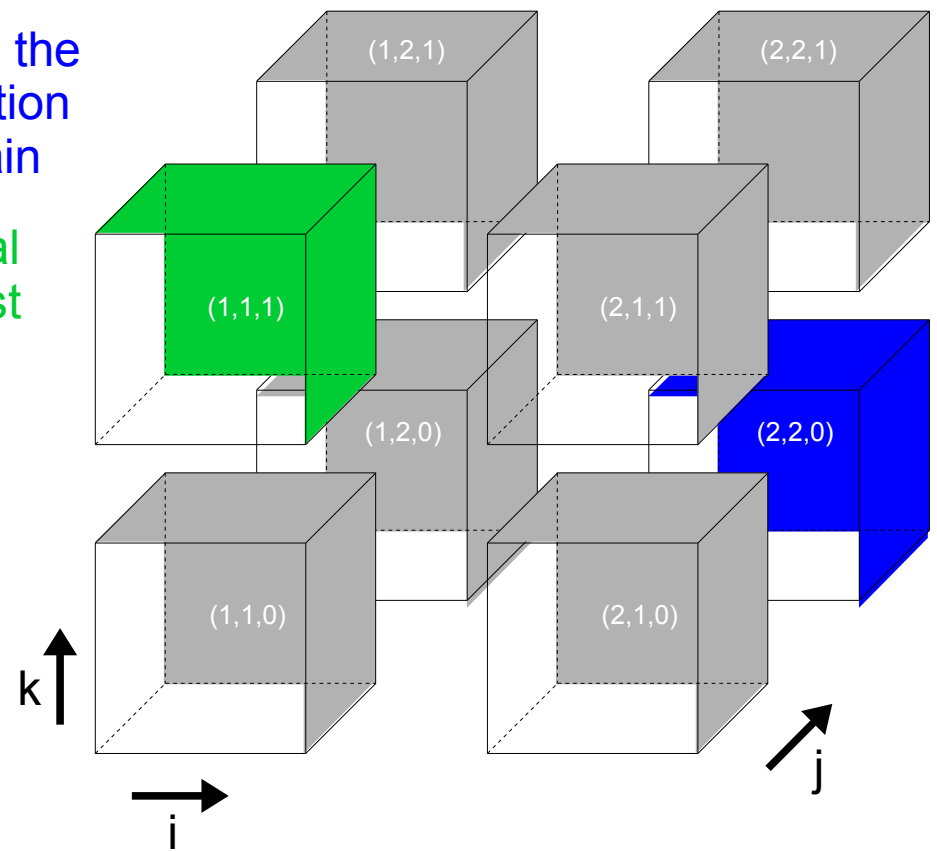
$\text{facedata}_x(1,1,1) = \text{facedata}_x(1,2,2)$
 $\text{facedata}_y(1,1,1) = \text{facedata}_y(2,1,2)$
 $\text{facedata}_z(1,1,1) = \text{facedata}_z(2,2,1)$

Ghost cells

Cells in the simulation domain

Local ghost cell

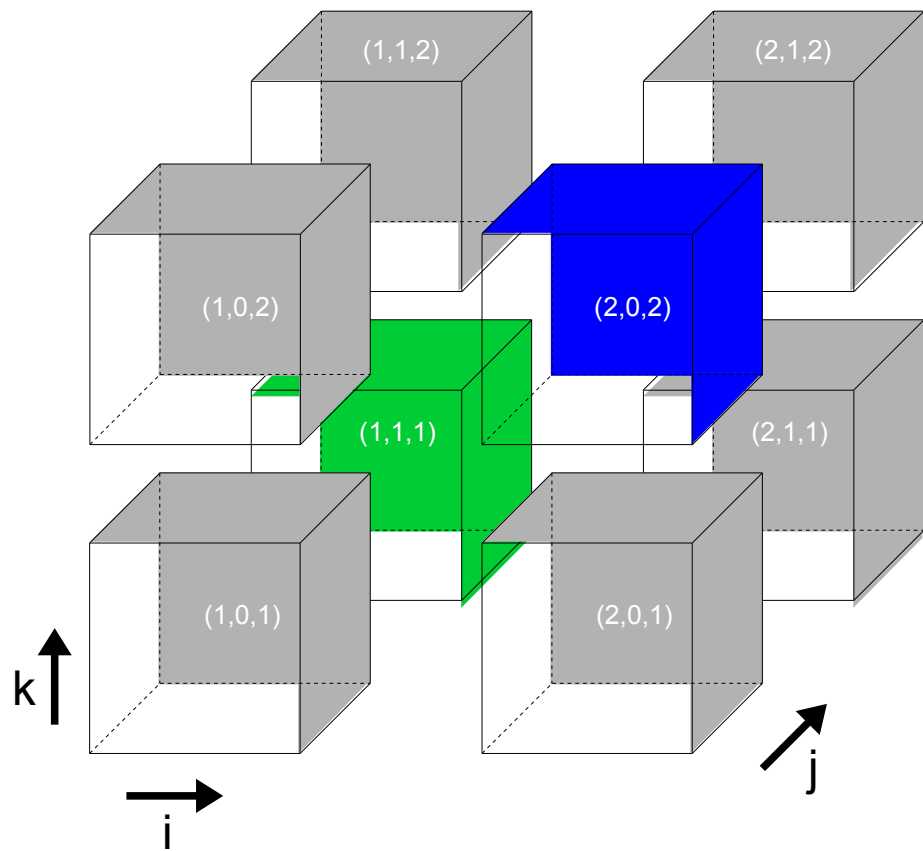
$(-x, -y, +z)$ corner



$\text{facedata}_x(1,1,1) = \text{facedata}_x(1,2,0)$
 $\text{facedata}_y(1,1,1) = \text{facedata}_y(2,1,0)$
 $\text{facedata}_z(1,1,1) = \text{ghost outer face}$

Neumann boundaries for corner faces

$(-x, +y, -z)$ corner



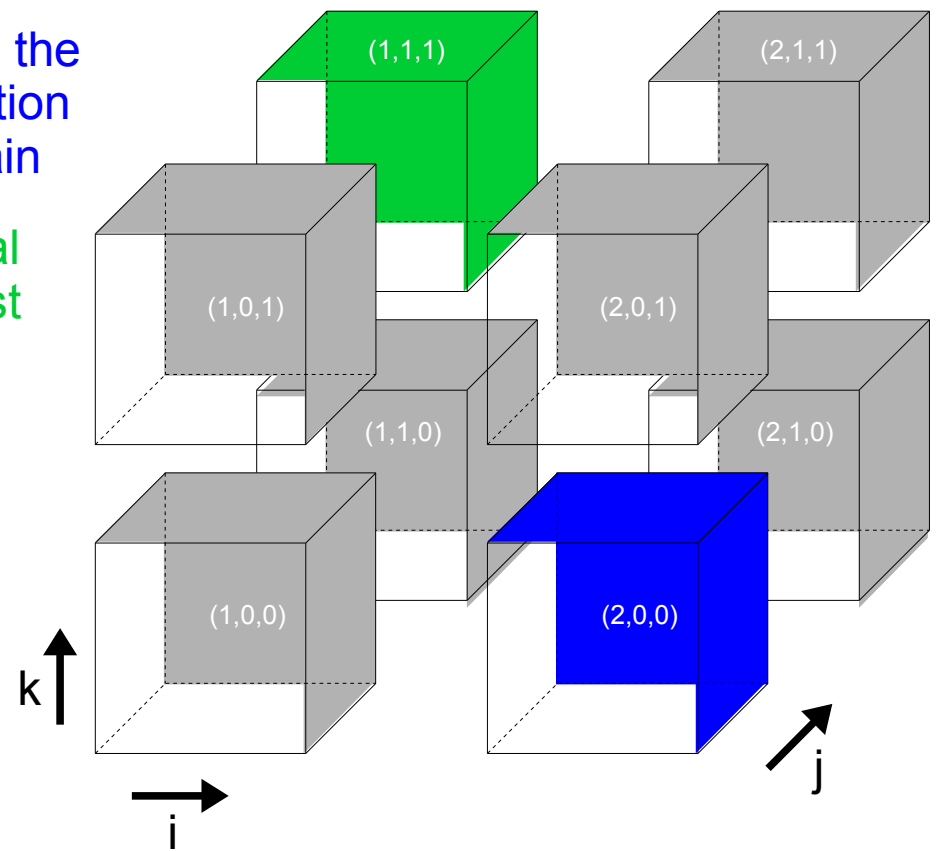
$\text{facedata}_x(1,1,1) = \text{facedata}_x(1,0,2)$
 $\text{facedata}_y(1,1,1) = \text{ghost outer face}$
 $\text{facedata}_z(1,1,1) = \text{facedata}_z(2,0,1)$

Ghost cells

Cells in the simulation domain

Local ghost cell

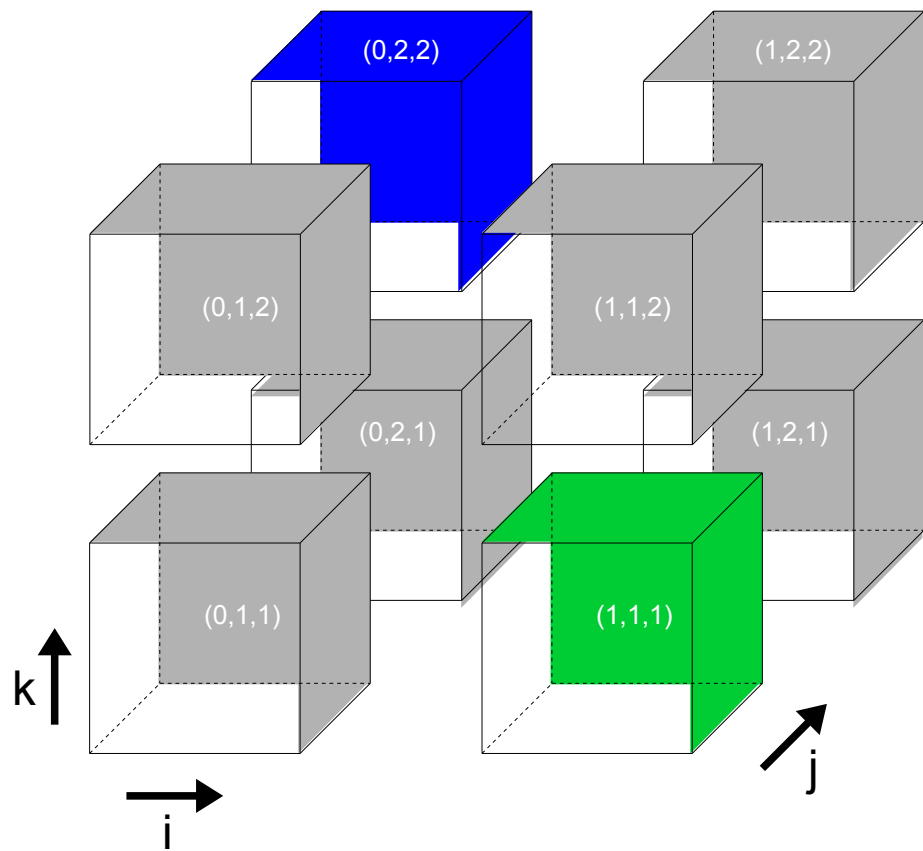
$(-x, +y, +z)$ corner



$\text{facedata}_x(1,1,1) = \text{facedata}_x(1,0,0)$
 $\text{facedata}_y(1,1,1) = \text{ghost outer face}$
 $\text{facedata}_z(1,1,1) = \text{ghost outer face}$

Neumann boundaries for corner faces

(+x,-y,-z) corner



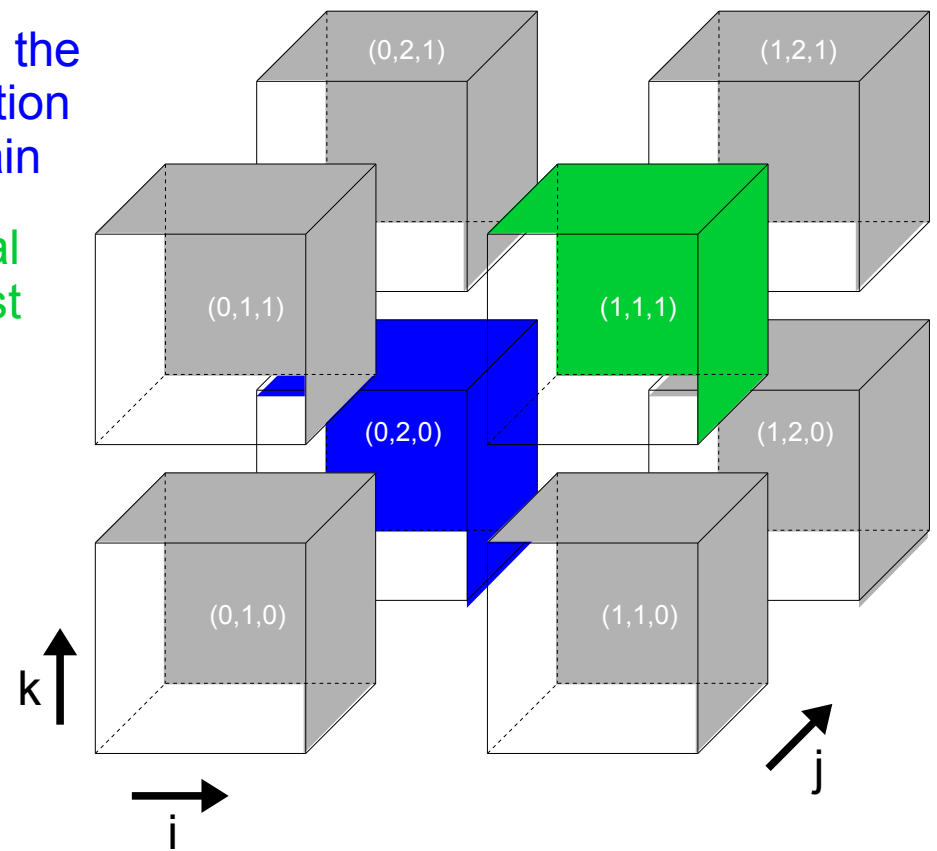
`facedata_x(1,1,1)` = ghost outer face
`facedata_y(1,1,1)` = `facedata_y(0,1,2)`
`facedata_z(1,1,1)` = `facedata_z(0,2,1)`

Ghost cells

Cells in the simulation domain

Local ghost cell

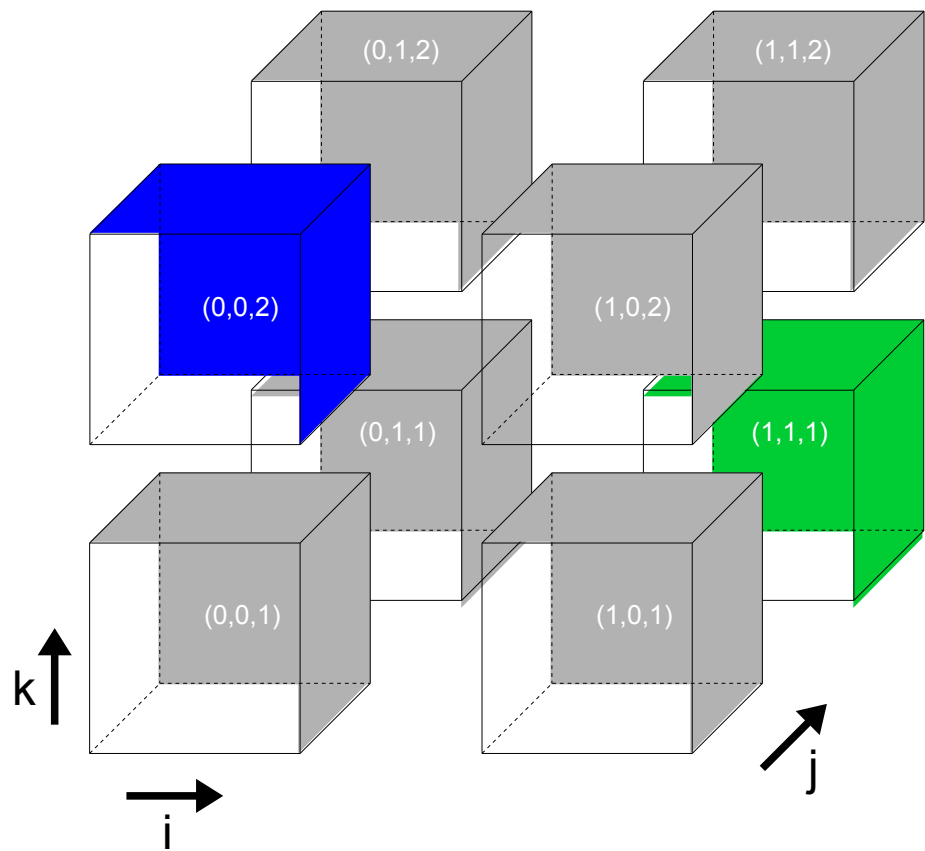
(+x,-y,+z) corner



`facedata_x(1,1,1)` = ghost outer face
`facedata_y(1,1,1)` = `facedata_y(0,1,0)`
`facedata_z(1,1,1)` = ghost outer face

Neumann boundaries for corner faces

(+x,+y,-z) corner



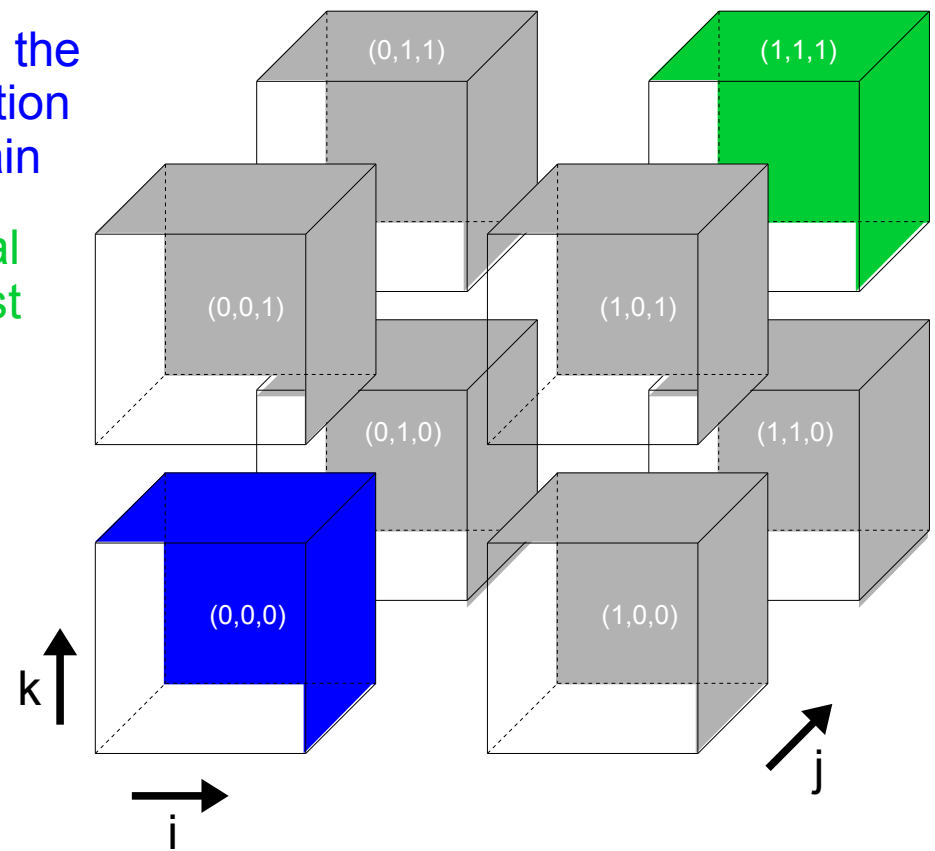
`facedata_x(1,1,1)` = ghost outer face
`facedata_y(1,1,1)` = ghost outer face
`facedata_z(1,1,1)` = `facedata_z(0,0,1)`

Ghost
cells

Cells in the
simulation
domain

Local
ghost
cell

(+x,+y,+z) corner



`facedata_x(1,1,1)` = ghost outer face
`facedata_y(1,1,1)` = ghost outer face
`facedata_z(1,1,1)` = ghost outer face