

LA-UR-16-26192

Approved for public release; distribution is unlimited.

Title: An Adaptive Mesh Algorithm: Mapping the Mesh Variables

Author(s): Scannapieco, Anthony J.

Intended for: LA-UR

Issued: 2016-08-11

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

An Adaptive Mesh Algorithm: Mapping the Mesh Variables

A. J. Scannapieco

July 25, 2016

1 Mapping the physical variables

Both thermodynamic and kinematic variables must be mapped. The kinematic variables are defined on a separate kinematic mesh; it is the dual mesh to the thermodynamic mesh. The map of the kinematic variables is done by calculating the contributions of kinematic variables on the *old* thermodynamic mesh, mapping the kinematic variable contributions onto the *new* thermodynamic mesh and then synthesizing the mapped kinematic variables on the *new* kinematic mesh. In this document the map of the thermodynamic variables will be described.

After the new mesh structure has been generated the physical variables are mapped from the *old* to the *new* mesh. The map is done in three parts.

1. The physical variables in zones that are unchanged are collected and saved for redistribution on the *new* mesh, which includes all mixed zones.
2. The physical variables in zones that are created by coarsening the mesh are mapped from the *old* to the *new* mesh.
3. The physical variables in zones that are created by refining the mesh are mapped from the *old* to the *new* mesh.

Two types of physical variables are mapped: extensive variables that are naturally conserved upon being mapped (e.g. density) and non-extensive variables that are not conserved when mapped (e.g. equivalent plastic strain).

Before the map, non-extensive variables are multiplied by the material density. After the map of the non-extensive variables they are divided by the *new* densities to recover the *new* non-extensive variables. It is important to note that the mixed zones are not mapped, because they remain on the finest mesh. Therefore, the mixed zone data are also retained unchanged.

1.1 Map the coarsened thermodynamic variables

The *new* extensive variables, $\phi(N_z)$ associated with the *new* coarsened zone N_z are obtained by volume summing the contributions made by the *old* daughter zones that comprise the *new* coarsened zone.

$$\phi(N_z) = \sum_{i=1}^{2^{\varkappa}} \nu(n_i) \psi(n_i) / \sum_{i=1}^{2^{\varkappa}} \nu(n_i) \quad (1)$$

In equation 1. $\varkappa$ is the problem dimension, $\nu(n_i)$ is the volume of daughter zone n_i , and $\psi(n_i)$ is the extensive variable of the *old* daughter zone n_i .

The *new* non-extensive variables, $\phi(N_z)$ associated with the *new* coarsened zone N_z are obtained by density summing the contributions made by the daughter zones that comprise the *new* coarsened zone.

$$\phi(N_z) = \sum_{i=1}^{2^{\varkappa}} \rho(n_i) \psi(n_i) / \sum_{i=1}^{2^{\varkappa}} \rho(n_i) \quad (2)$$

In equation 2 $\rho(n_i)$ is the density of the *old* daughter zone n_i , and $\psi(n_i)$ is a non-extensive variable of daughter zone n_i . As can be seen, the map of the *new* coarsened zones is conservative in the extensive variables and conservative for the non-extensive variables insofar as those variables are associated with a zonal mass of material. The set of coarsened zones and the *new* data is collected and stored to be used in the synthesis of the *new* mesh. In two and three dimensions the *new* coarsened zone N_z and the *old* daughter zones that contribute to zone N_z are displayed in Figures 1 and 2.

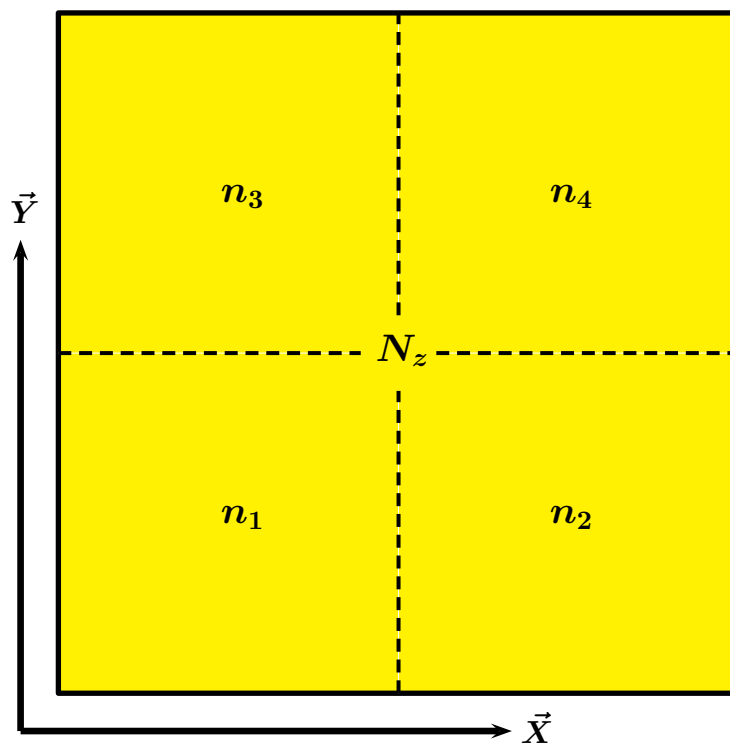


Figure 1: Two dimensional logical daughter index layout for the map of the coarsened zone N_z .

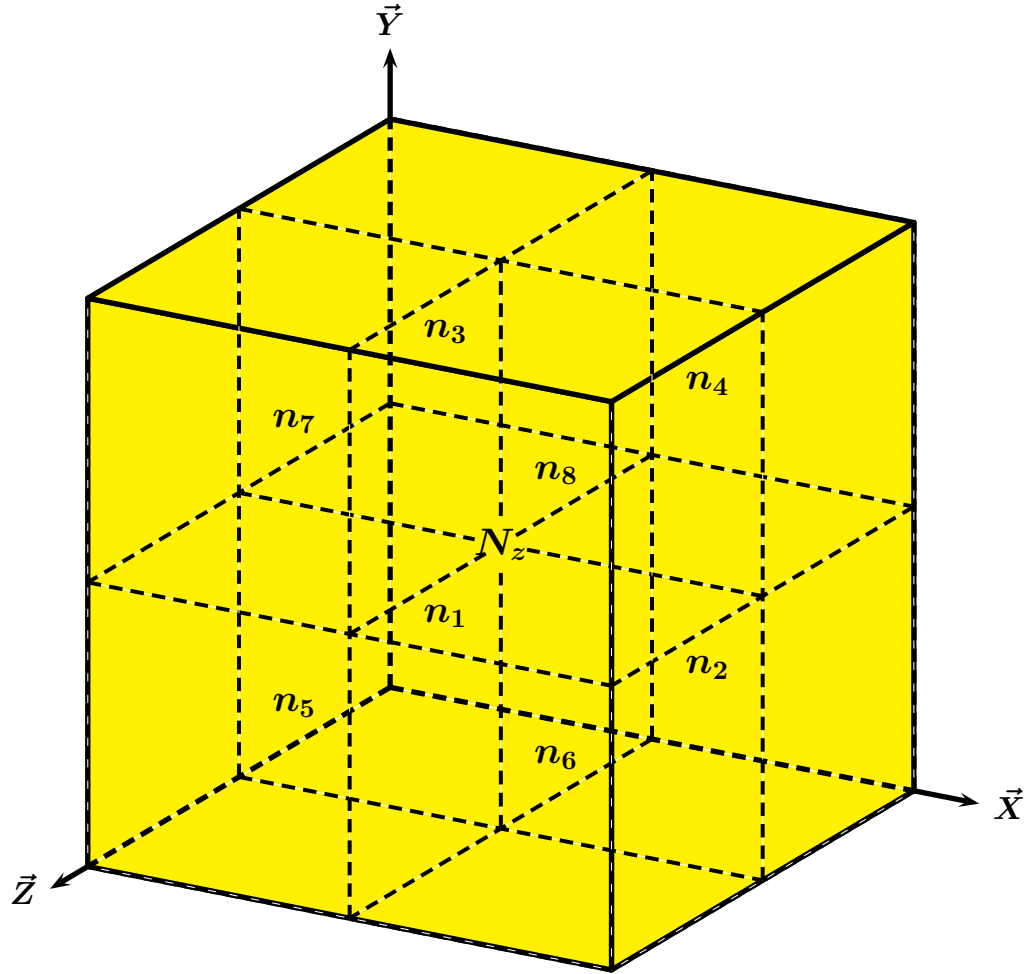


Figure 2: Three dimensional logical daughter index layout for the map of coarsened zone N_z .

1.2 Map the refined thermodynamic variables

Each *new* refined zone shares a common vertex of the *old* zone that is to be refined. The zone to be refined and the *satellite* zones surrounding the common vertex are the zones used to do the high-order, monotonic, conservative map of the *new* refined zone. The same set of zones are used to map every *new* refined zone that shares that vertex; this ensures consistency in the data and map of all the *new* zones that share that vertex. If $\varkappa$ is the problem dimension, then the number of zones that share the vertex is, *at most*, $2^\varkappa$. Table 1. defines the variables used in calculating the pseudo anti-diffusive fluxes $\psi(i, j, k)$ and the *new* extensive variable $\phi(i, j, k)$.

$n_e(i, j, k, ib)$	Zonal index of neighbor zone (<i>ib</i>) at vertex (<i>i, j, k</i>).
$d(n_e(i, j, k, ib))$	Material density in neighbor zone (<i>ib</i>) at vertex (<i>i, j, k</i>).
$f(n_e(i, j, k, ib))$	Physical variable in neighbor zone (<i>ib</i>) at vertex (<i>i, j, k</i>).
$f_x(i, j, k, ib)$	Old extensive variable in neighbor zone (<i>ib</i>) at vertex (<i>i, j, k</i>). $f_x(i, j, k, ib) = f(n_e(i, j, k, ib))$ if <i>f</i> is an extensive variable. $f_x(i, j, k, ib) = f(n_e(i, j, k, ib))d(n_e(i, j, k, ib))$ if <i>f</i> is not an extensive variable.
$f_{mn}(i, j, k)$	Minimum value of $f_x(i, j, k, ib)$ at vertex (<i>i, j, k</i>).
$f_{mx}(i, j, k)$	Maximum value of $f_x(i, j, k, ib)$ at vertex (<i>i, j, k</i>).
$\delta f_{x_{mn}}(i, j, k, ib)$	Minimum change of f_x relative to $f_x(i, j, k, 0)$ at vertex (<i>i, j, k</i>). $\delta f_{x_{mn}}(i, j, k, ib) = f_{mn}(i, j, k) - f_x(i, j, k, 0)$
$\delta f_{x_{mx}}(i, j, k, ib)$	Maximum change of f_x relative to $f_x(i, j, k, 0)$ at vertex (<i>i, j, k</i>). $\delta f_{x_{mx}}(i, j, k, ib) = f_{mx}(i, j, k) - f_x(i, j, k, 0)$
$x(i, j, k, ib, idir)$	Fractional distance with respect to zone 0 of zone center (<i>i, j, k, ib</i>) relative to zone center (<i>i, j, k, 0</i>) in the <i>idir</i> direction.
$g(i, j, k, ib, ic)$	Geometrical contribution of neighbor zone <i>ib</i> to zone (<i>i, j, k</i>) of matrix element <i>ic</i> .
$\phi(i, j, k)$	New extensive variable in <i>new</i> zone (<i>i, j, k</i>).

Table 1: Glossary of variables used in the high-order refinement map.

In two and three dimensions the *new* zone and the *satellite* zones, along with the associated indexing are displayed in Figures 3 through 14. The (i, j, k) index of the *new* refined zone is defined relative to the 0 zone.

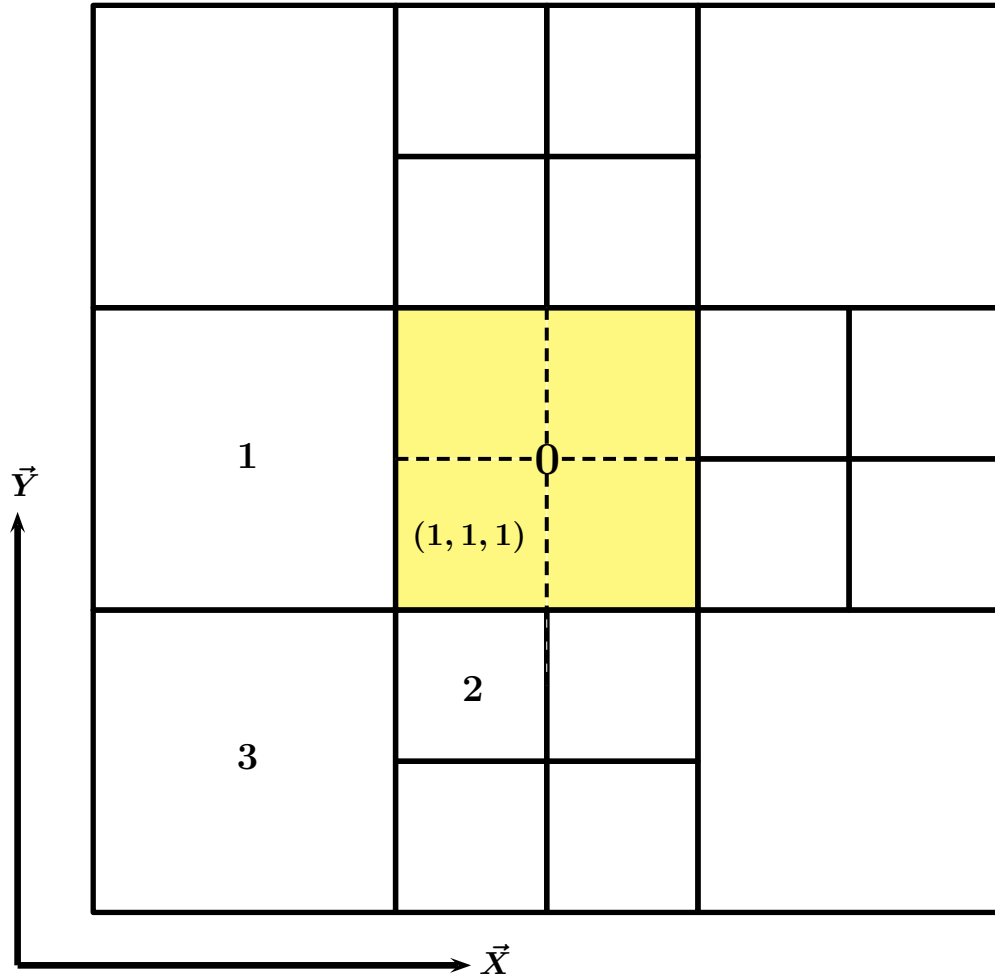


Figure 3: Two dimensional logical neighbor index layout for high order map of the refined zone $(1, 1, 1)$. $n_e(1, 1, 1, ib) = ib$, $0 \leq ib \leq 3$.

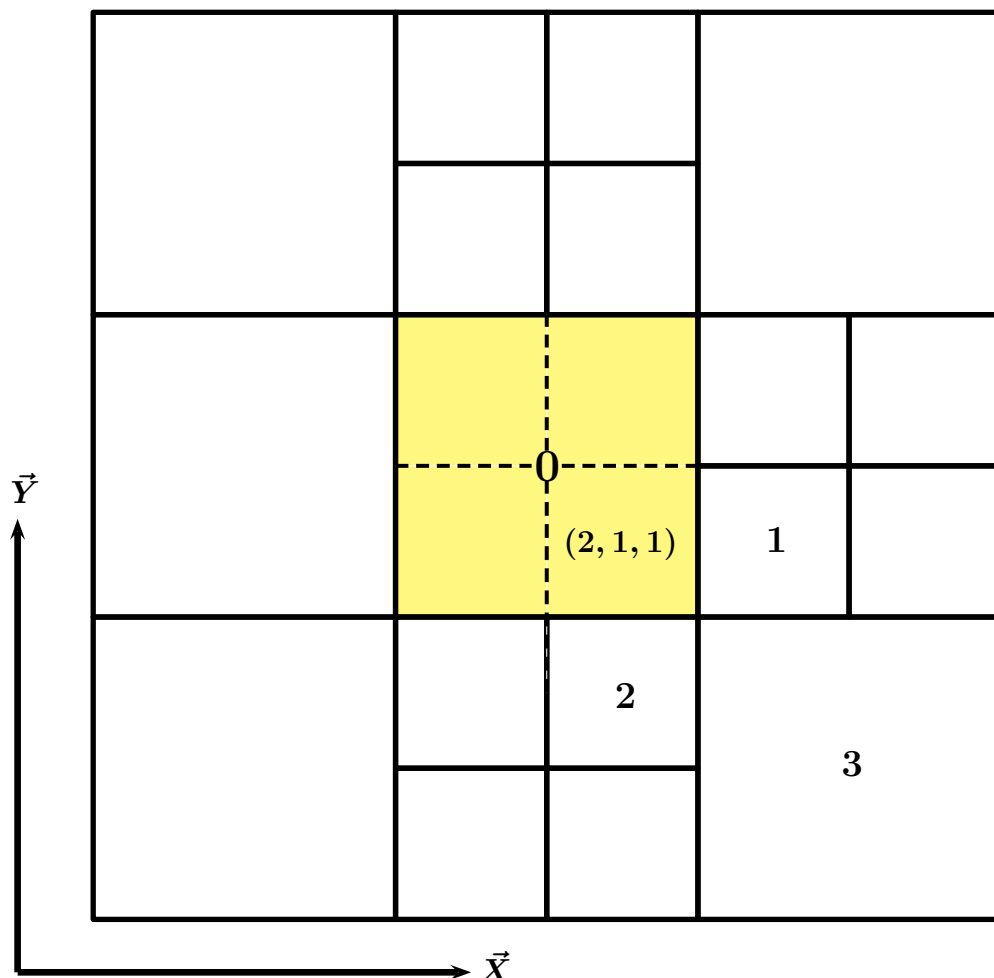


Figure 4: Two dimensional logical neighbor index layout for high order map of the refined zone $(2, 1, 1)$. $n_e(2, 1, 1, ib) = ib$, $0 \leq ib \leq 3$.

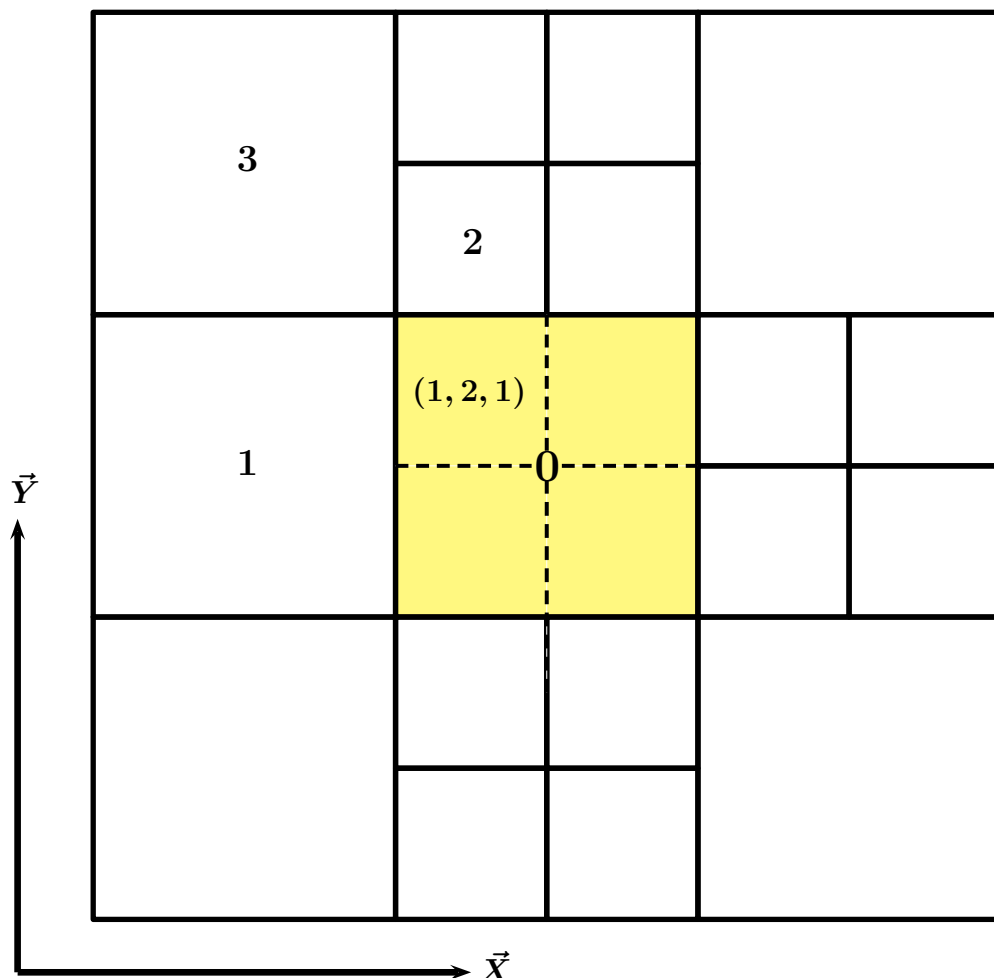


Figure 5: Two dimensional logical neighbor index layout for high order map of the refined zone $(1, 2, 1)$. $n_e(1, 2, 1, ib) = ib$, $0 \leq ib \leq 3$.

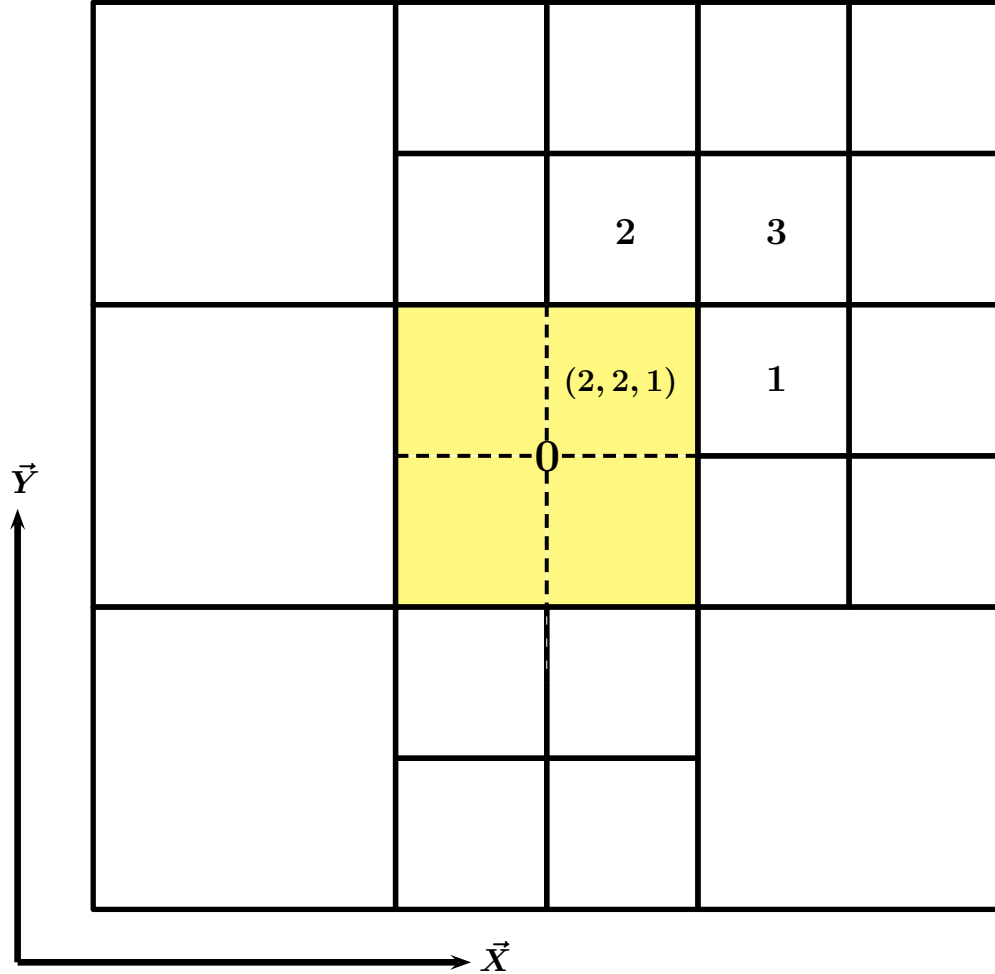


Figure 6: Two dimensional logical neighbor index layout for high order map of the refined zone $(2, 2, 1)$. $n_e(2, 2, 1, ib) = ib$, $0 \leq ib \leq 3$.

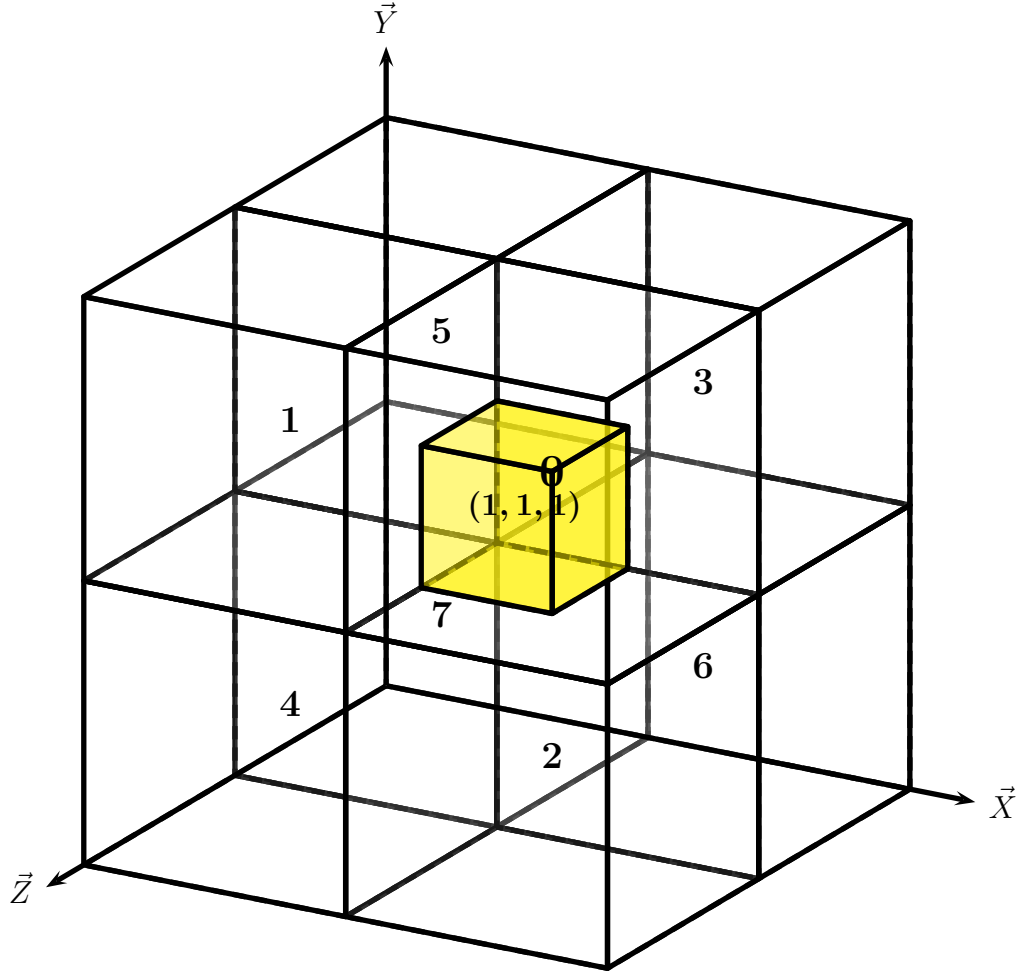


Figure 7: Three dimensional logical neighbor index layout for high order map of refined zone $(1, 1, 1)$. $n_e(1, 1, 1, ib) = ib$, $0 \leq ib \leq 7$.

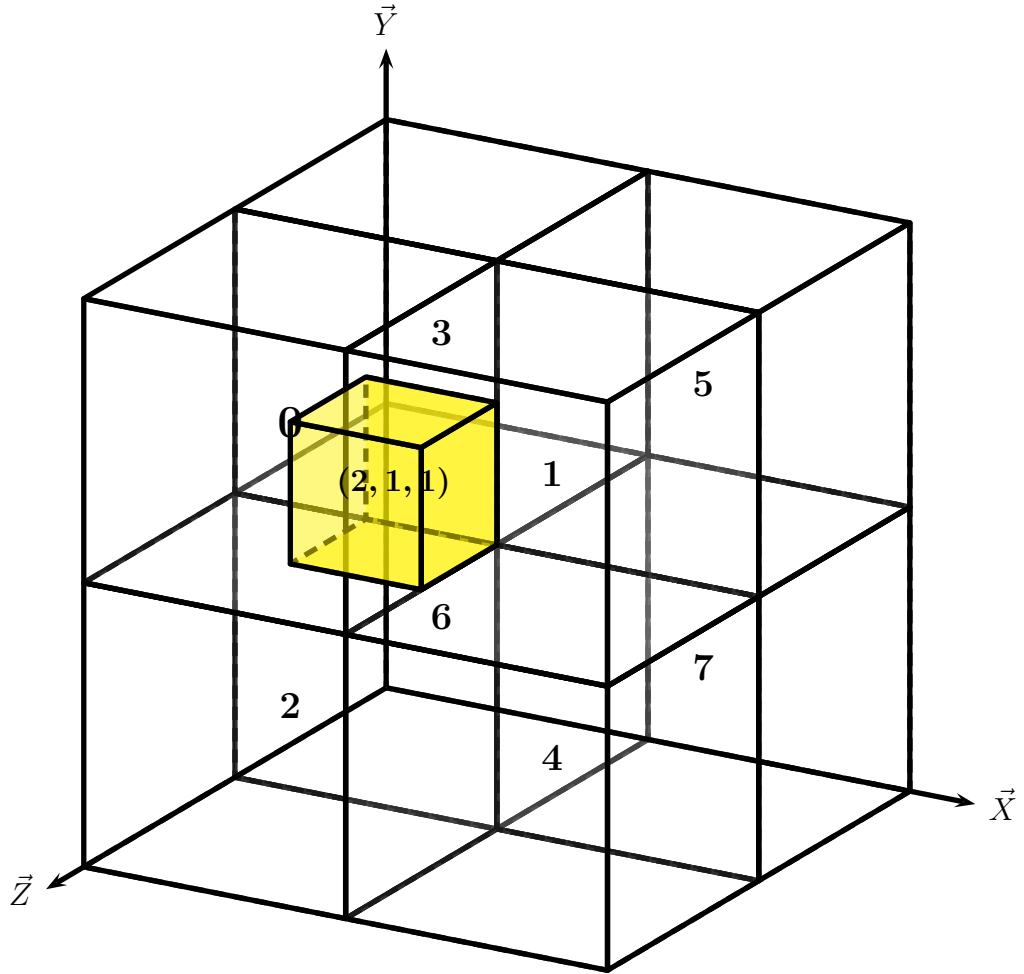


Figure 8: Three dimensional logical neighbor index layout for high order map of refined zone $(2, 1, 1)$. $n_e(2, 1, 1, ib) = ib$, $0 \leq ib \leq 7$.

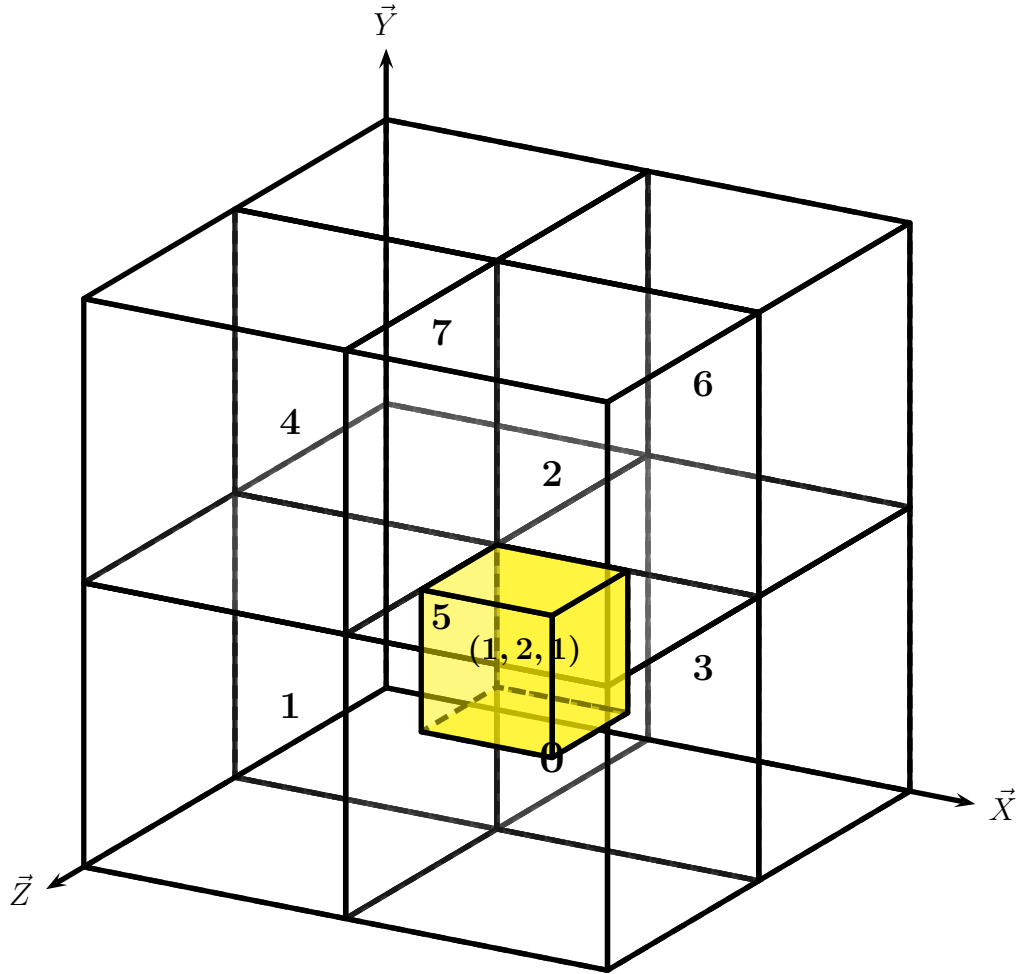


Figure 9: Three dimensional logical neighbor index layout for high order map of refined zone $(1, 2, 1)$. $n_e(1, 2, 1, ib) = ib$, $0 \leq ib \leq 7$.

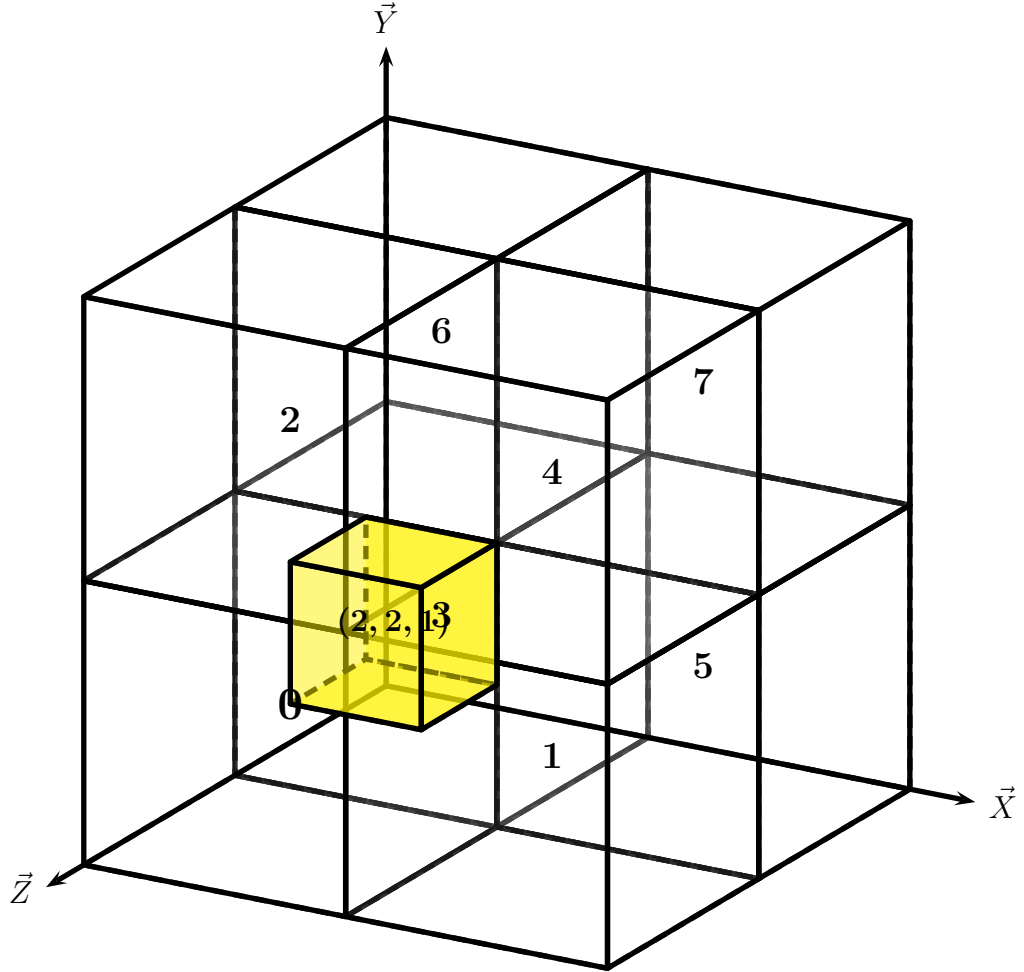


Figure 10: Three dimensional logical neighbor index layout for high order map of refined zone $(2, 2, 1)$. $n_e(2, 2, 1, ib) = ib$, $0 \leq ib \leq 7$.

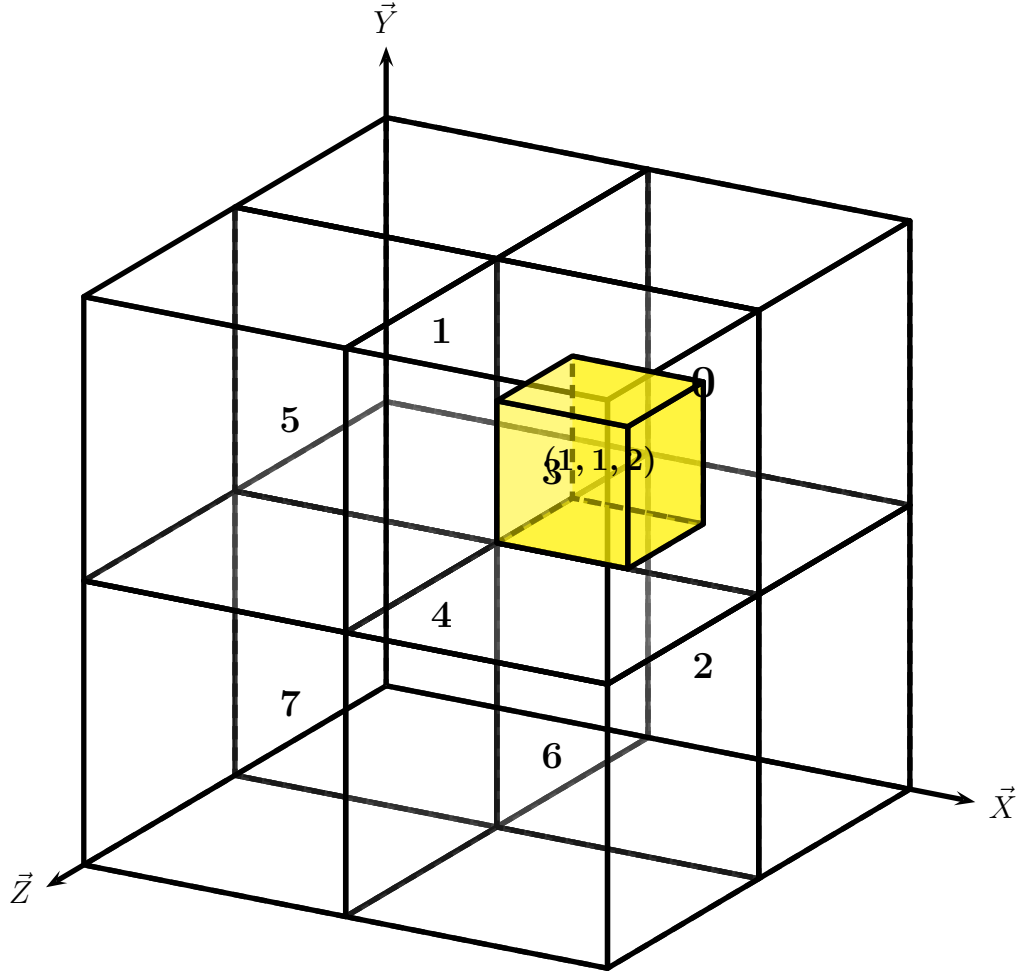


Figure 11: Three dimensional logical neighbor index layout for high order map of refined zone $(1, 1, 2)$. $n_e(1, 1, 2, ib) = ib$, $0 \leq ib \leq 7$.

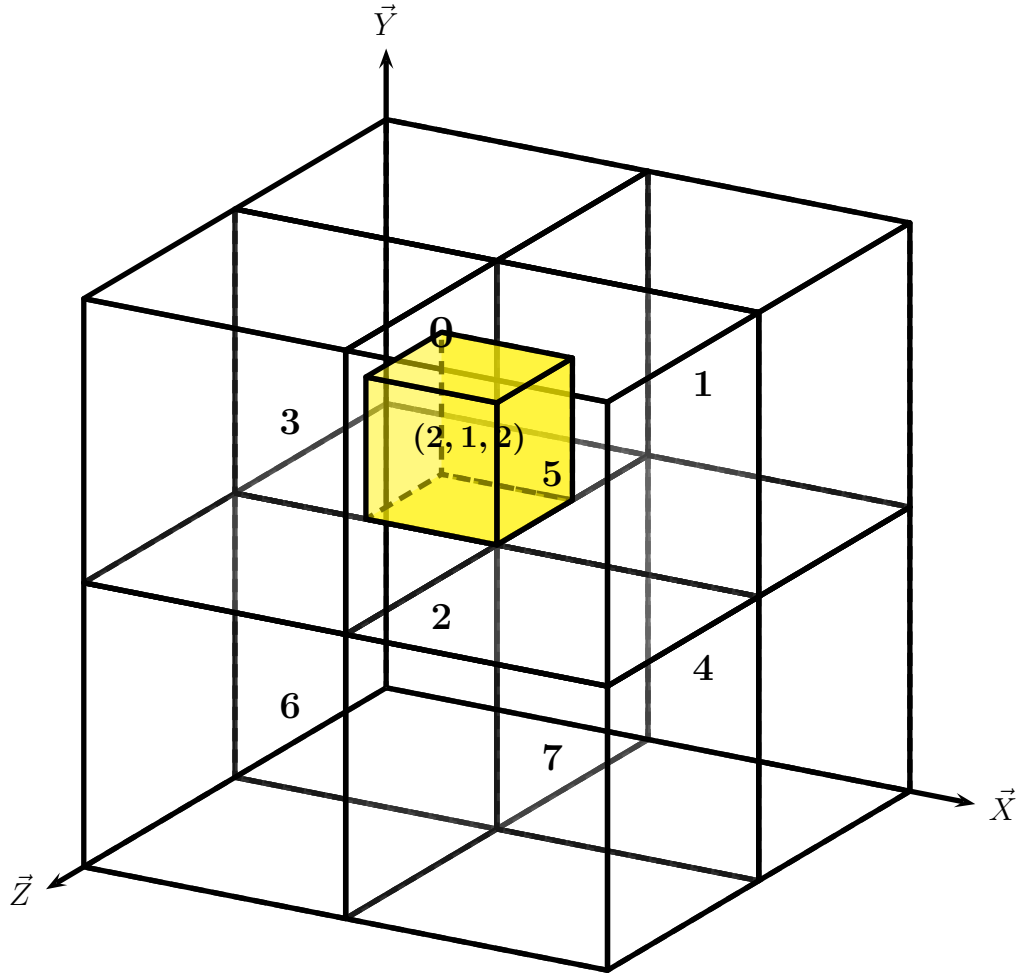


Figure 12: Three dimensional logical neighbor index layout for high order map of refined zone $(2, 1, 2)$. $n_e(2, 1, 2, ib) = ib$, $0 \leq ib \leq 7$.

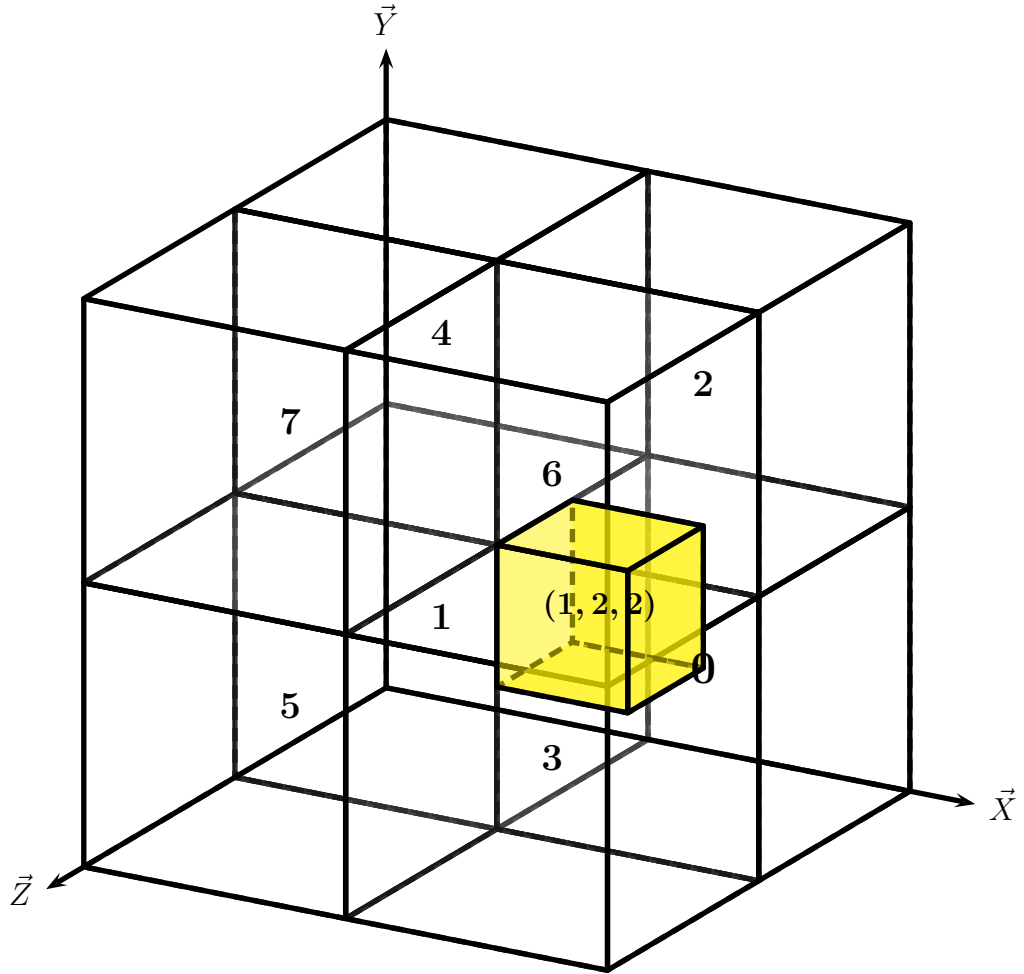


Figure 13: Three dimensional logical neighbor index layout for high order map of refined zone $(1, 2, 2)$. $n_e(1, 2, 2, ib) = ib$, $0 \leq ib \leq 7$.

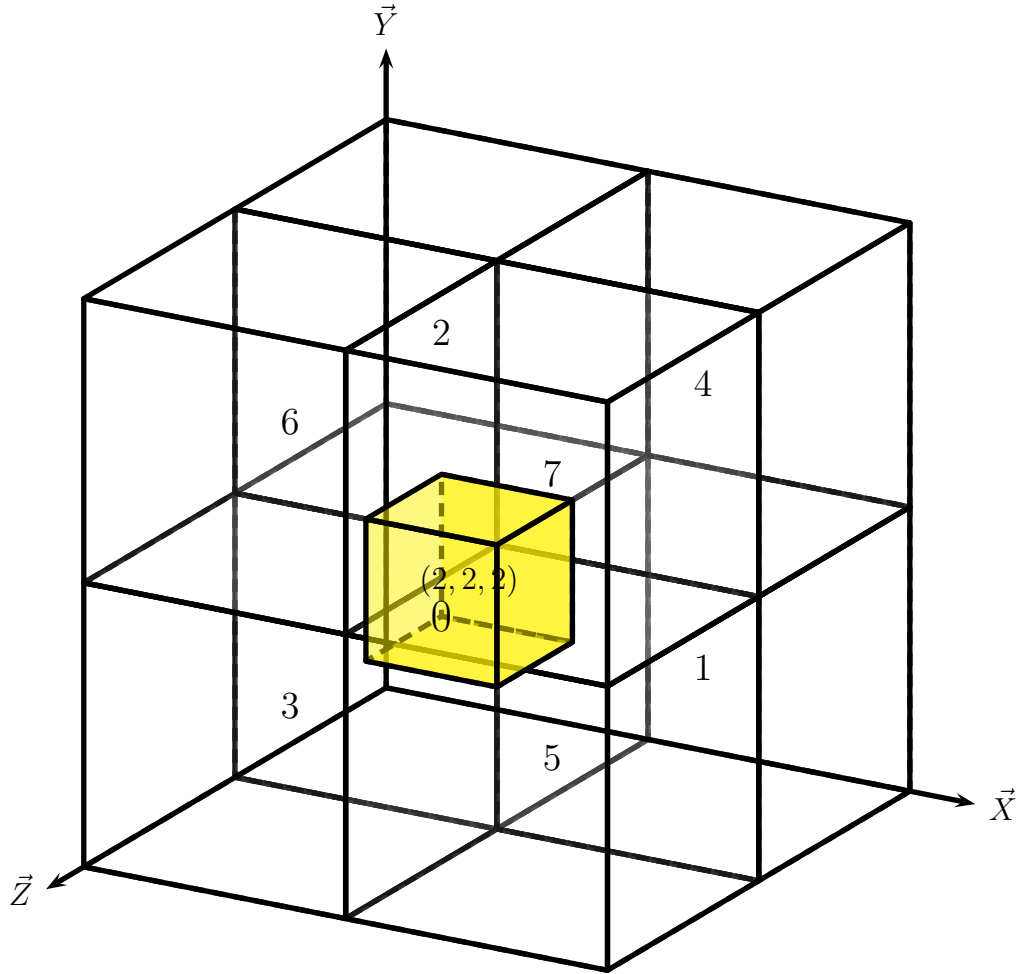


Figure 14: Three dimensional logical neighbor index layout for high order map of refined zone $(2, 2, 2)$. $n_e(2, 2, 2, ib) = ib$, $0 \leq ib \leq 7$.

The refinement map is done using a multi-dimensional Flux-Corrected-Transport algorithm¹ in which a high-order pseudo anti-diffusive flux is calculated for each *new* zone and added to the original state variable of the 0 zone to calculate the *new* state variable in each of the *new* refined zones. The fluxes are limited via the fully multi-dimensional FCT algorithm which, ensures both conservation and monotonicity over the map.

The pseudo anti-diffusive fluxes are calculated by first defining $\delta(\varkappa, ic)$ where $\varkappa$ is the number of dimensions and ic is the matrix element.

$$\begin{aligned}
\delta(1, 1) &\equiv x(i, 1, 1, 1) \\
\delta(2, 1) &\equiv x(i, j, 1, 1) \\
\delta(2, 2) &\equiv x(i, j, 1, 2) \\
\delta(2, 3) &\equiv x(i, j, 1, 1)x(i, j, 1, 2) \\
\delta(3, 1) &\equiv x(i, j, k, 1) \\
\delta(3, 2) &\equiv x(i, j, k, 2) \\
\delta(3, 3) &\equiv x(i, j, k, 3) \\
\delta(3, 4) &\equiv x(i, j, k, 1)x(i, j, k, 2) \\
\delta(3, 5) &\equiv x(i, j, k, 1)x(i, j, k, 3) \\
\delta(3, 6) &\equiv x(i, j, k, 2)x(i, j, k, 3) \\
\delta(3, 7) &\equiv x(i, j, k, 1)x(i, j, k, 2)x(i, j, k, 3)
\end{aligned}$$

The pseudo anti-diffusive fluxes $\psi(i, j, k)$ are

$$\psi(i, j, k) = \sum_{ib=1}^{2^\varkappa-1} \sum_{ic=1}^{2^\varkappa-1} \delta(\varkappa, ic) g(i, j, k, ib, ic) (f_x(i, j, k, ib) - f_x(i, j, k, 0))$$

These fluxes are limited to ensure that monotonicity is maintained.

$$\psi(i, j, k) = \{\delta f_{x_{mx}}, \{\delta f_{x_{mn}}, \psi(i, j, k)\}_{max}\}_{min}$$

The final steps ensure that the map is conservative.

$$\begin{aligned}
\psi_{mn}(i, j, k) &= \{0, \psi(i, j, k)\}_{min} \\
\psi_{mx}(i, j, k) &= \{0, \psi(i, j, k)\}_{max}
\end{aligned}$$

¹S. T. Zalesak, J. Comput. Phys. 31(1979), 335

$$\Psi_{mn} = \left| \sum_{i=1}^{i_{mx}} \sum_{j=1}^{j_{mx}} \sum_{k=1}^{k_{mx}} \psi_{mn}(i, j, k) \right|$$

$$\Psi_{mx} = \left| \sum_{i=1}^{i_{mx}} \sum_{j=1}^{j_{mx}} \sum_{k=1}^{k_{mx}} \psi_{mx}(i, j, k) \right|$$

$$\begin{aligned} i_{mx} &= 2 \\ j_{mx} &= 1 & \varkappa < 2 \\ j_{mx} &= 2 & \varkappa \geq 2 \\ k_{mx} &= 1 & \varkappa < 3 \\ k_{mx} &= 2 & \varkappa = 3 \end{aligned}$$

$$\begin{aligned} \alpha_{mn} &= \Psi_{mx} / \{\Psi_{mn}, \Psi_{mx}\}_{max} \\ \alpha_{mx} &= \Psi_{mn} / \{\Psi_{mn}, \Psi_{mx}\}_{max} \end{aligned}$$

Finally the *new* value of the physical variable, $\phi(i, j, k)$ in the *new* zone (i, j, k) is

$$\phi(i, j, k) = f_x(i, j, k, 0) + \alpha_{mx}\psi(i, j, k) \quad \psi(i, j, k) \geq 0 \quad (3)$$

$$\phi(i, j, k) = f_x(i, j, k, 0) + \alpha_{mn}\psi(i, j, k) \quad \psi(i, j, k) < 0 \quad (4)$$

1.2.1 Calculate the Geometrical Contributions

In one, two and three dimensions the geometrical coefficients, $g(i, j, k, ib, ic)$ are calculated and used to map all the *new* zonal extensive variables. The $g(i, j, k, ib, ic)$ variables are solved by inverting a matrix equation that relates the unknown variables d_i , defined below, to the values of the physical variables in each zone surrounding the common vertex.

In one dimension the map of the spatial derivative d_1 is linear and two independent values of the *old* variable are needed to create the geometrical coefficients.

$$\begin{aligned} f &= (a_x + b_x(x - x_0)) \\ f &= a_x + b_x(x - x_0) \\ f &= f_0 + f_x(x - x_0) \end{aligned} \quad (5)$$

$$\begin{aligned}
s_1 &\equiv (f_1 - f_0) \\
d_1 &\equiv f_x \\
a_{11} &\equiv (x_1 - x_0) \\
a_{11}d_1 &= s_1
\end{aligned} \tag{6}$$

In two dimensions the map of the spatial derivatives d_i is bi-linear and four independent values of the *old* variable are needed to create the geometrical coefficients.

$$\begin{aligned}
f &= (a_x + b_x(x - x_0))(a_y + b_y(y - y_0)) \\
f &= a_x a_y + b_x a_y(x - x_0) + a_x b_y(y - y_0) + b_x b_y(x - x_0)(y - y_0) \\
f &= f_0 + f_x(x - x_0) + f_y(y - y_0) + f_{xy}(x - x_0)(y - y_0)
\end{aligned} \tag{7}$$

$$\begin{aligned}
s_1 &\equiv (f_1 - f_0) \\
s_2 &\equiv (f_2 - f_0) \\
s_3 &\equiv (f_3 - f_0)
\end{aligned}$$

$$\begin{aligned}
d_1 &\equiv f_x \\
d_2 &\equiv f_y \\
d_3 &\equiv f_{xy}
\end{aligned}$$

$$\begin{aligned}
a_{11} &\equiv (x_1 - x_0), \quad a_{12} \equiv (y_1 - y_0), \quad a_{13} \equiv (x_1 - x_0)(y_1 - y_0) \\
a_{21} &\equiv (x_2 - x_0), \quad a_{22} \equiv (y_2 - y_0), \quad a_{23} \equiv (x_2 - x_0)(y_2 - y_0) \\
a_{31} &\equiv (x_3 - x_0), \quad a_{32} \equiv (y_3 - y_0), \quad a_{33} \equiv (x_3 - x_0)(y_3 - y_0)
\end{aligned}$$

$$\begin{aligned}
a_{11}d_1 + a_{12}d_2 + a_{13}d_3 &= s_1 \\
a_{21}d_1 + a_{22}d_2 + a_{23}d_3 &= s_2 \\
a_{31}d_1 + a_{32}d_2 + a_{33}d_3 &= s_3
\end{aligned} \tag{8}$$

In three dimensions the map of the spatial derivatives, d_i is tri-linear and eight independent values of the *old* variable are needed to create the geometrical coefficients.

$$\begin{aligned}
f &= (a_x + b_x(x - x_0))(a_y + b_y(y - y_0))(a_z + b_z(z - z_0)) \\
f &= f_0 + f_x(x - x_0) + f_y(y - y_0) + f_z(z - z_0) + \\
&\quad f_{xy}(x - x_0)(y - y_0) + f_{xz}(x - x_0)(z - z_0) + f_{yz}(y - y_0)(z - z_0) + \\
&\quad f_{xyz}(x - x_0)(y - y_0)(z - z_0)
\end{aligned} \tag{9}$$

$$s_1 \equiv (f_1 - f_0)$$

$$s_2 \equiv (f_2 - f_0)$$

$$s_3 \equiv (f_3 - f_0)$$

$$s_4 \equiv (f_4 - f_0)$$

$$s_5 \equiv (f_5 - f_0)$$

$$s_6 \equiv (f_6 - f_0)$$

$$s_7 \equiv (f_7 - f_0)$$

$$d_1 \equiv f_x$$

$$d_2 \equiv f_y$$

$$d_3 \equiv f_z$$

$$d_4 \equiv f_{xy}$$

$$d_5 \equiv f_{xz}$$

$$d_6 \equiv f_{yz}$$

$$d_7 \equiv f_{xyz}$$

$$\begin{aligned}
a_{11} &\equiv (x_1 - x_0), \quad a_{12} \equiv (y_1 - y_0), \quad a_{13} \equiv (z_1 - z_0), \\
a_{14} &\equiv (x_1 - x_0)(y_1 - y_0), \quad a_{15} \equiv (x_1 - x_0)(z_1 - z_0), \quad a_{16} \equiv (y_1 - y_0)(z_1 - z_0), \\
a_{17} &\equiv (x_1 - x_0)(y_1 - y_0)(z_1 - z_0), \\
a_{21} &\equiv (x_2 - x_0), \quad a_{22} \equiv (y_2 - y_0), \quad a_{23} \equiv (z_2 - z_0), \\
a_{24} &\equiv (x_2 - x_0)(y_2 - y_0), \quad a_{25} \equiv (x_2 - x_0)(z_2 - z_0), \quad a_{26} \equiv (y_2 - y_0)(z_2 - z_0), \\
a_{27} &\equiv (x_2 - x_0)(y_2 - y_0)(z_2 - z_0), \\
a_{31} &\equiv (x_3 - x_0), \quad a_{32} \equiv (y_3 - y_0), \quad a_{33} \equiv (z_3 - z_0), \\
a_{34} &\equiv (x_3 - x_0)(y_3 - y_0), \quad a_{35} \equiv (x_3 - x_0)(z_3 - z_0), \quad a_{36} \equiv (y_3 - y_0)(z_3 - z_0), \\
a_{37} &\equiv (x_3 - x_0)(y_3 - y_0)(z_3 - z_0), \\
a_{41} &\equiv (x_4 - x_0), \quad a_{42} \equiv (y_4 - y_0), \quad a_{43} \equiv (z_4 - z_0), \\
a_{44} &\equiv (x_4 - x_0)(y_4 - y_0), \quad a_{45} \equiv (x_4 - x_0)(z_4 - z_0), \quad a_{46} \equiv (y_4 - y_0)(z_4 - z_0), \\
a_{47} &\equiv (x_4 - x_0)(y_4 - y_0)(z_4 - z_0), \\
a_{51} &\equiv (x_5 - x_0), \quad a_{52} \equiv (y_5 - y_0), \quad a_{53} \equiv (z_5 - z_0), \\
a_{54} &\equiv (x_5 - x_0)(y_5 - y_0), \quad a_{55} \equiv (x_5 - x_0)(z_5 - z_0), \quad a_{56} \equiv (y_5 - y_0)(z_5 - z_0), \\
a_{57} &\equiv (x_5 - x_0)(y_5 - y_0)(z_5 - z_0), \\
a_{61} &\equiv (x_6 - x_0), \quad a_{62} \equiv (y_6 - y_0), \quad a_{63} \equiv (z_6 - z_0), \\
a_{64} &\equiv (x_6 - x_0)(y_6 - y_0), \quad a_{65} \equiv (x_6 - x_0)(z_6 - z_0), \quad a_{66} \equiv (y_6 - y_0)(z_6 - z_0), \\
a_{67} &\equiv (x_6 - x_0)(y_6 - y_0)(z_6 - z_0), \\
a_{71} &\equiv (x_7 - x_0), \quad a_{72} \equiv (y_7 - y_0), \quad a_{73} \equiv (z_7 - z_0), \\
a_{74} &\equiv (x_7 - x_0)(y_7 - y_0), \quad a_{75} \equiv (x_7 - x_0)(z_7 - z_0), \quad a_{76} \equiv (y_7 - y_0)(z_7 - z_0), \\
a_{77} &\equiv (x_7 - x_0)(y_7 - y_0)(z_7 - z_0)
\end{aligned}$$

$$\begin{aligned}
a_{11}d_1 + a_{12}d_2 + a_{13}d_3 + a_{14}d_4 + a_{15}d_5 + a_{16}d_6 + a_{17}d_7 &= s_1 \\
a_{21}d_1 + a_{22}d_2 + a_{23}d_3 + a_{24}d_4 + a_{25}d_5 + a_{26}d_6 + a_{27}d_7 &= s_2 \\
a_{31}d_1 + a_{32}d_2 + a_{33}d_3 + a_{34}d_4 + a_{35}d_5 + a_{36}d_6 + a_{37}d_7 &= s_3 \\
a_{41}d_1 + a_{42}d_2 + a_{43}d_3 + a_{44}d_4 + a_{45}d_5 + a_{46}d_6 + a_{47}d_7 &= s_4 \\
a_{51}d_1 + a_{52}d_2 + a_{53}d_3 + a_{54}d_4 + a_{55}d_5 + a_{56}d_6 + a_{57}d_7 &= s_5 \\
a_{61}d_1 + a_{62}d_2 + a_{63}d_3 + a_{64}d_4 + a_{65}d_5 + a_{66}d_6 + a_{67}d_7 &= s_6 \\
a_{71}d_1 + a_{72}d_2 + a_{73}d_3 + a_{74}d_4 + a_{75}d_5 + a_{76}d_6 + a_{77}d_7 &= s_7
\end{aligned} \tag{10}$$

It is not always the case that there will be 2^z zones that define a refinement map. Consider the two-dimensional case shown in Figure 15 in which both zones 0 and $0'$ are scheduled to be refined. In this case vertex 1 and vertex 3 are the same.

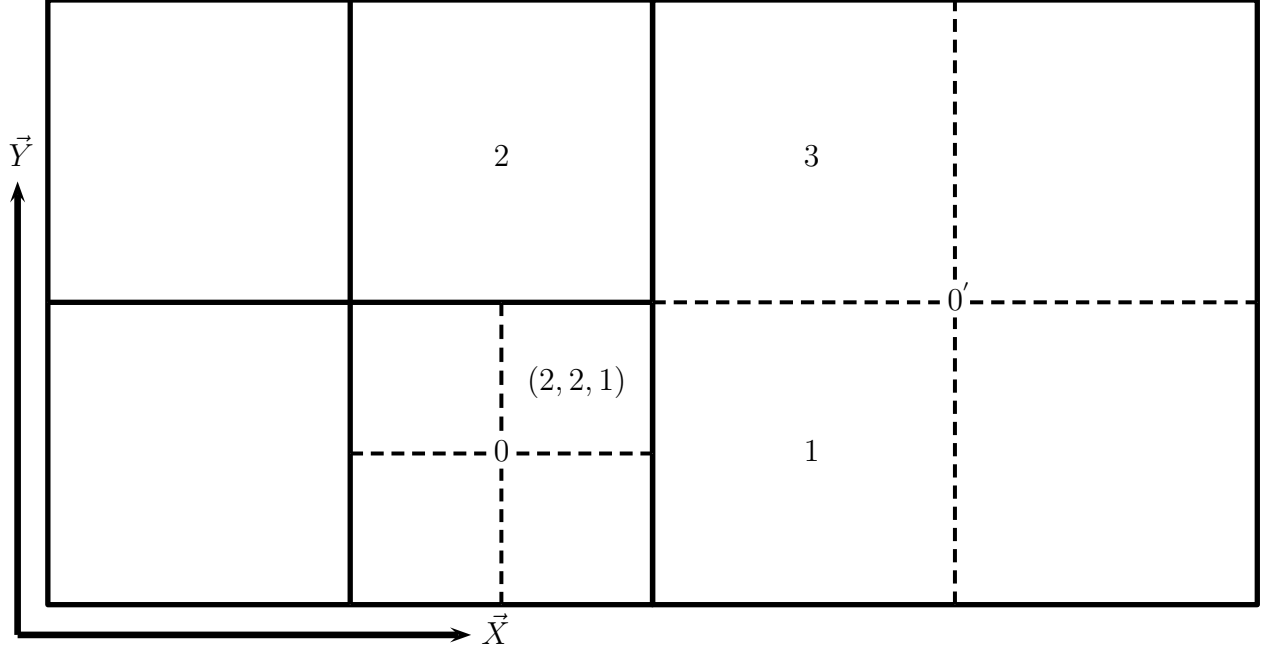


Figure 15: Degenerate case of a two dimensional logical neighbor index layout for high order map of the refined zone $(2, 2, 1)$. $n_e(2, 2, 1, ib) = ib$, $0 \leq ib \leq 2$

The coordinates of zones 1 and 3 are equal to the coordinates of the center of zone $0'$. The map for this degenerate case is no longer bi-linear but linear.

$$f = f_0 + f_x(x - x_0) + f_y(y - y_0) \quad (11)$$

$$s_1 \equiv (f_1 - f_0)$$

$$s_2 \equiv (f_2 - f_0)$$

$$d_1 \equiv f_x$$

$$d_2 \equiv f_y$$

$$\begin{aligned} a_{11} &\equiv (x_1 - x_0), \quad a_{12} \equiv (y_1 - y_0), \\ a_{21} &\equiv (x_2 - x_0), \quad a_{22} \equiv (y_2 - y_0) \end{aligned}$$

$$\begin{aligned} a_{11}d_1 + a_{12}d_2 &= s_1 \\ a_{21}d_1 + a_{22}d_2 &= s_2 \end{aligned} \tag{12}$$

In the above equations, for all dimensions:

$$\begin{aligned} x_p &= x(i, j, k, p, 1) \\ y_p &= x(i, j, k, p, 2) \\ z_p &= x(i, j, k, p, 3) \\ f_p &= f(ne(i, j, k, p)) \quad 0 \leq p \leq 2^\varkappa - 1 \end{aligned}$$

where $\varkappa$ is the problem dimension.

In each case the matrix equation is inverted by solving for the inverse of the matrix a_{ij} . The matrix equations in one, two, and three dimensions, written as

$$a_{ij}d_j = s_i \tag{13}$$

can be solved using cofactors. The cofactor of the matrix a_{ki} is

$$c_{ki} = (-1)^{k+i} \det((a_{p,q})_{p \neq k, q \neq i})$$

The inverse of the matrix a_{ki} is

$$a_{ki}^{-1} = c_{ik} / \det(a_{ki})$$

where c_{ik} is the transpose of the cofactor c_{ki} . The solution of the matrix equations is given as:

$$\begin{aligned} a_{ki}^{-1} a_{ij} d_j &= a_{ki}^{-1} s_i \\ a_{ki}^{-1} a_{ij} &= \delta_{kj} \end{aligned}$$

where δ_{kj} is the Kronecker delta. Therefore

$$\delta_{kj} d_j = a_{ki}^{-1} s_i$$

and

$$d_k = a_{ki}^{-1} s_i \tag{14}$$

The geometrical coefficients $g(i, j, k, ib, ic)$ are than defined as

$$g(i, j, k, ib, ic) \equiv a_{ic,ib}^{-1} \tag{15}$$

where the $a_{ic,ib}^{-1}$ are calculated for each (i, j, k) *new* zone.

2 Conclusion

The mapping of the thermodynamic physical variables has been presented when coarsening and/or refinement of the *new* mesh is required. The maps are conservative, monotonic high order pseudo FCT maps. It is important to note that when the kinematic quantities reside on the thermodynamic mesh or when they are defined at panels of the thermodynamic mesh, the mesh description above applies as stated. If however, the kinematic variables reside at the vertices of the thermodynamic mesh than a separate map of the kinematic variables is required on a regenerated kinematic mesh. In that case the *duel* kinematic mesh has to be regenerated and the kinematic variables mapped unto that *new* kinematic mesh. The map on the *new* duel mesh proceeds exactly as on the thermodynamic mesh. The map described here has been used in a wide variety of one, two and three dimensional problems in a multi-physics code.