

Appendix A

Notation and Mathematical Symbols

A.1 Notation

A.1.1 Indices

Figure A.1 shows a very small sample mesh. This mesh shall now be used to explain the notation. The figure also shows a graph describing the connection between the nodes of the mesh. A lower index is a vector or matrix subscript. Mostly it is used in connection with discrete values on a mesh, however. Then the first lower index of a symbol describes the number of the node (or control volume). Thus \mathbf{x}_4 describes the position vector of the node which has been highlighted in figure A.1. If a lower index pair is set into round parenthesis it refers to a connection. Respectively the term $(\mathbf{n}\Delta A)_{(4,4)}$ describes the connecting interface between nodes 4 and 5. If an index pair is set into curly brackets, the absolute index on the other side of the connection is meant (e.g. $\mathbf{x}_{\{4,4\}} = \mathbf{x}_5$). Symbolic indices are set into a different font. Thus $N_{\text{ngh},4}$ describes the number of neighbors of node number 4, which is equal to 5 in this example. Upper indices denote a time or iteration level.

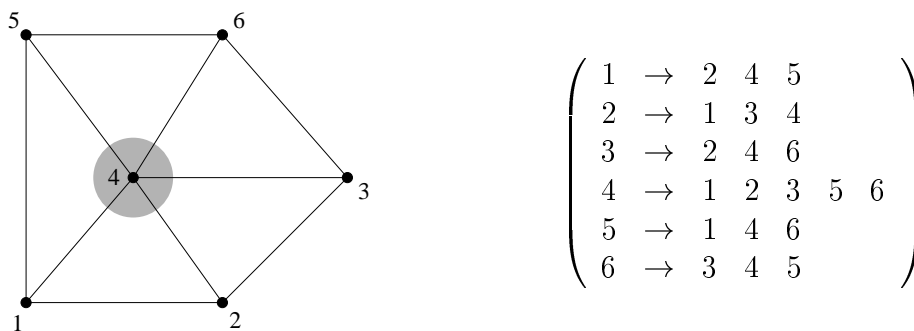


Figure A.1: Indices used for grids.

A.1.2 Unknowns of Partial Differential Equations

Scalar Equations

Generally ϕ has been used as unknown variable for scalar partial differential equations. It is a function of space and time, or just space in case of a steady state problem. It has been used for one-, or multi-dimensional equations. See the following example equations

$$\frac{\partial x}{\partial t} + \frac{\partial^2 \phi}{\partial x^2} = 0 \longrightarrow \phi(x, t) \quad , \quad \frac{\partial \phi}{\partial t} + \nabla \phi = 0 \longrightarrow \phi(\mathbf{x}, t) \quad \text{or} \quad \nabla^2 \phi = 0 \longrightarrow \phi(\mathbf{x}).$$

For discrete equations ϕ becomes a vector of discrete variables:

$$\boldsymbol{\phi} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{N_{\text{nds}}} \end{pmatrix}$$

The number of discrete unknowns is equal to the number of nodes N_{nds} of the grid.

Systems

The unknowns of systems of equations are denoted by the vector \mathbf{q} , just like in the following equation:

$$\int_V \frac{\partial \mathbf{q}}{\partial t} dV + \oint_{\partial V} \mathbf{H} \mathbf{n} dA = 0. \quad (\text{A.1})$$

In the discrete form \mathbf{q} becomes a matrix:

$$\mathbf{Q} = \begin{pmatrix} q_{1,1} & q_{1,2} & q_{1,3} & \cdots & q_{1,N_{\text{nds}}} \\ q_{2,1} & q_{2,2} & q_{2,3} & \cdots & q_{2,N_{\text{nds}}} \\ q_{3,1} & q_{3,2} & q_{3,3} & \cdots & q_{3,N_{\text{nds}}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ q_{N_{\text{eq}},1} & q_{N_{\text{eq}},2} & q_{N_{\text{eq}},3} & \cdots & q_{N_{\text{eq}},N_{\text{nds}}} \end{pmatrix}, \quad \mathbf{q}_i = \begin{pmatrix} q_{1,i} \\ q_{2,i} \\ q_{3,i} \\ \vdots \\ q_{N_{\text{eq}},i} \end{pmatrix}.$$

Please note that for simplicity reasons \mathbf{q}_i has been used as the i th column of \mathbf{Q} . Thus \mathbf{q}_i describes the vector of unknowns on a node i of a mesh.

A.1.3 Special Notations

[] Square Brackets

Square-brackets indicate a discretized expression. $[\int_V \mathbf{q} dV]_i$, for example, denotes a discretized volume integral for the i th finite volume.

A.2 Symbols

A.2.1 Indices

Symbolic Lower Indices

H	denotes coarse grid discretization (lower index)
h	denotes a fine grid discretization (if used as a lower index)
corr	corrected
drv	driving discretization
err	error
expl	explicit
inv	inviscid
ltd	limited
max	maximal
min	minimal
tgt	denotes a target-discretization
vis	viscous

Lower Indices

$from, to$	adjacent nodes to a discrete interface
i	counter for nodes/control-volumes
j	counter for connections/neighbors
l	index used for unknowns or equations

Upper Indices

k	iteration index
n	time level

A.2.2 Numbers of Objects

N_{eq}	equations
N_{nds}	nodes
N_{rk}	Runge-Kutta steps
N_{seg}	number of segments of a polygon

A.2.3 Operators

I_H^h	variable transfer operator from coarse to fine
I_h^H	variable transfer operator from fine to coarse
L	linear operator
Φ	non-linear operator
Res	residual operator

∇	gradient operator in Cartesian coordinates ($\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})^T$)
cond	condition number
A	matrix representing a spatial operator
P	pre-conditioning matrix
$I_{r_h}^H$	residual transfer operator from fine to coarse

A.2.4 Physical Symbols

E	energy
R	gas constant
T	temperature
γ	specific heat ratio
$\boldsymbol{\sigma}$	stress tensor
λ	heat conductivity
μ	dynamic viscosity
ρ	density
τ	viscous stresses
c	propagation speed
f	vector of external forces
v	velocity vector
q	vector of conservative properties
a	speed of sound
c_p	specific heat at constant pressure
c_v	specific heat at constant density
e	inner energy
p	pressure
t	physical time
u	velocity in Cartesian x-direction
v	velocity in Cartesian y-direction
w	velocity in Cartesian z-direction

A.2.5 Geometrical Symbols

A	surface of a control-volume
V	a continuous volume
∂V	the boundary of a continuous volume
e	vector connection two neighboring nodes
n	normal vector
x	Cartesian position vector
a	a geometrical length
b	a geometrical length
dim	spatial dimension
e	eccentricity of an ellipse

h	edge length
l	geometrical length
n_x, n_y, n_z	components of a normal vector
r	radius
s	integration variable for surface integrals
x, y, z	Cartesian coordinates
$\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$	Cartesian base vectors
α	angle

A.2.6 Dimensionless Numbers

CFL	CFL number
M	Mach number
Re	Reynolds number
c_D	drag coefficient
c_L	lift coefficient
c_P	pressure coefficient

A.2.7 Numerical Symbols

VA	van Albada function
Δ	a discrete fraction
$\alpha_0, \alpha_1 \dots$	Runge-Kutta coefficients
β	upwind parameter
ω	relaxation factor
ω	weighting factor
\mathbf{f}_H	multi-grid forcing function
\mathbf{r}	residual vector
\mathbf{s}	vector of source terms

A.2.8 Miscellaneous Symbols

C	a constant value
I	imaginary unit
Q	quadrangle
Θ	angle (used for stability analysis)
ϵ	a small value
Λ	eigenvalue matrix
λ	characteristic of a partial differential equation
ϕ	variable used for exemplary scalar equations
$\tilde{}$	denotes a reconstructed value on a discrete interface between two control volumes
\mathbf{H}	flux vector
\mathbf{I}	unit matrix

R	eigenvector matrix
W	vector of characteristic variables
b	right hand side vector of a system of linear equations
f	flux vector in Cartesian x-direction
h_{heat}	vector of conductive heat transfer
a_0, a_1, a_2, \dots	coefficients
e	the number e
f	a function
g	function used for level-sets
g_{line}	level-set defining a grid-line for boundary layer grids
k	wave-number

Appendix B

Gradients

B.1 Calculation of Gradients Using the Divergence Theorem

A vector field \mathbf{f} is given in Cartesian coordinates. Gauss' divergence theorem describes a relation between an integral of the vector field's divergence over any bounded volume V and a surface integral of the vector field itself over the boundary of the volume ∂V .

$$\int_V (\nabla \cdot \mathbf{f}) dV = \oint_{\partial V} \mathbf{f} \cdot \mathbf{n} dA \quad (\text{B.1})$$

Using the three cartesian base vectors

$$\mathbf{e}_x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{e}_y = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{e}_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (\text{B.2})$$

For any scalar function ϕ , three equations (one for each Cartesian direction) can be formulated using equation (B.1) and the trivial vector fields $\phi \mathbf{e}_x$, $\phi \mathbf{e}_y$ and $\phi \mathbf{e}_z$:

$$\int_V \nabla \cdot (\phi \mathbf{e}_x) dV = \oint_{\partial V} \phi \mathbf{e}_x \cdot \mathbf{n} dA \quad \rightarrow \quad \int_V \begin{pmatrix} \frac{\partial \phi}{\partial x} \\ 0 \\ 0 \end{pmatrix} dV = \oint_{\partial V} \begin{pmatrix} \phi n_x \\ 0 \\ 0 \end{pmatrix} dA \quad (\text{B.3})$$

$$\int_V \nabla \cdot (\phi \mathbf{e}_y) dV = \oint_{\partial V} \phi \mathbf{e}_y \cdot \mathbf{n} dA \quad \rightarrow \quad \int_V \begin{pmatrix} 0 \\ \frac{\partial \phi}{\partial y} \\ 0 \end{pmatrix} dV = \oint_{\partial V} \begin{pmatrix} 0 \\ \phi n_y \\ 0 \end{pmatrix} dA \quad (\text{B.4})$$

$$\int_V \nabla \cdot (\phi \mathbf{e}_z) dV = \oint_{\partial V} \phi \mathbf{e}_z \cdot \mathbf{n} dA \quad \rightarrow \quad \int_V \begin{pmatrix} 0 \\ 0 \\ \frac{\partial \phi}{\partial z} \end{pmatrix} dV = \oint_{\partial V} \begin{pmatrix} 0 \\ 0 \\ \phi n_z \end{pmatrix} dA \quad (\text{B.5})$$

Adding (B.3), (B.4), (B.5) yields a formulation for the gradient $\nabla \phi$:

$$\int_V (\nabla \phi) dV = \oint_{\partial V} \phi \mathbf{n} dA \quad (\text{B.6})$$

B.2 Gradient Using the Least Squares Approach

ϕ is a scalar function in (x, y, z) which is given on the discrete nodes of an unstructured grid. To compute a discrete gradient $[\nabla\phi]_i$ on a node i , a Taylor expansion of ϕ around the vertex i is used. It will then be tried to minimize the sum of the square of the errors on all neighboring nodes. The error shall be the difference between ϕ , computed using the Taylor-expansion and the discretely stored value of ϕ on the neighboring vertex. The following paragraph will explain this method in detail for a Taylor-series, which has been cut after the linear contributions.

$$\phi_{i,j} = \phi_i + \Delta\mathbf{x}_{i,j} \cdot [\nabla\phi]_i + O(\Delta\mathbf{x}_{i,j}^2) \quad , \quad \Delta\mathbf{x}_{i,j} = \begin{pmatrix} (\Delta x)_{i,j} \\ (\Delta y)_{i,j} \\ (\Delta z)_{i,j} \end{pmatrix} = \begin{pmatrix} x_{i,j} - x_i \\ y_{i,j} - y_i \\ z_{i,j} - z_i \end{pmatrix} \quad (\text{B.7})$$

The index j denotes the neighbor of node i . Thus i, j describes the j th neighbor of vertex i . An error function f_{err} is defined. It is a weighted sum of the error squares on all neighbor-nodes.

$$f_{err}([\nabla\phi]_i) = \sum_{j=1}^{n_i} \left(w_{i,j} (\phi_{i,j} - \phi_i - \Delta\mathbf{x}_{i,j} \cdot [\nabla\phi]_i)^2 \right) \quad (\text{B.8})$$

$w_{i,j}$ is a geometrical weighting factor. Good experiences have been made by using $w_{i,j} = 1/\Delta\mathbf{x}_{i,j}^2$. Even $w_{i,j} = 1$ gives quite good results. Please note that this approach will produce a consistent discretization for any value of $w_{i,j}$. f_{err} is a function in $[\nabla\phi]_i$ and to minimize f_{err} its gradient in $[\nabla\phi]_i$ shall now be computed and set to $(0, 0, 0)^T$.

$$\begin{aligned} \begin{pmatrix} \frac{\partial f_{err}}{\partial([\nabla\phi]_i)_x} \\ \frac{\partial f_{err}}{\partial([\nabla\phi]_i)_y} \\ \frac{\partial f_{err}}{\partial([\nabla\phi]_i)_z} \end{pmatrix} &= -2 \sum_{j=1}^{n_i} (w_{i,j} (\phi_{i,j} - \phi_i - \Delta\mathbf{x}_{i,j} \cdot [\nabla\phi]_i) \Delta\mathbf{x}_{i,j}) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\ &= \mathbf{A} \cdot [\nabla\phi]_i + \mathbf{b} , \\ \mathbf{A} &= -2 \sum_{j=1}^{n_i} w_{i,j} \begin{pmatrix} (\Delta x)_{i,j}^2 & (\Delta x)_{i,j}(\Delta y)_{i,j} & (\Delta x)_{i,j}(\Delta z)_{i,j} \\ (\Delta x)_{i,j}(\Delta y)_{i,j} & (\Delta y)_{i,j}^2 & (\Delta y)_{i,j}(\Delta z)_{i,j} \\ (\Delta x)_{i,j}(\Delta z)_{i,j} & (\Delta y)_{i,j}(\Delta z)_{i,j} & (\Delta z)_{i,j}^2 \end{pmatrix} , \\ \mathbf{b} &= -2 \sum_{j=1}^{n_i} (w_{i,j} (\phi_{i,j} - \phi_i) \Delta\mathbf{x}_{i,j}) \end{aligned} \quad (\text{B.9})$$

$$\Leftrightarrow [\nabla\phi]_i = -\mathbf{A}^{-1} \cdot \mathbf{b} \quad (\text{B.10})$$

The gradient of the discrete function ϕ can be computed using (B.10). Please note that the matrix \mathbf{A} is only dependent on the mesh geometry, not on ϕ . Thus it could be computed and inverted and then be stored for later use. The vector \mathbf{b} , however, depends on ϕ and has to be recomputed every time a gradient is needed. For two dimensional

problems (B.10) can be used as well, but the matrix \mathbf{A} will of course be different.

$$\mathbf{A}_{2d} = \begin{pmatrix} (\Delta x)_{i,j}^2 & (\Delta x)_{i,j}(\Delta y)_{i,j} \\ (\Delta x)_{i,j}(\Delta y)_{i,j} & (\Delta y)_{i,j}^2 \end{pmatrix} \quad (\text{B.11})$$

Appendix C

Consistency on Triangular Grids

For an arbitrarily triangulated grid it can be shown, that a consistent discretization of first derivatives can be achieved, as long as the variables are linearly reconstructed along the edges of the mesh. Figure C.1 a) shows a typical control volume of a triangular mesh. The demonstration shall be done in two dimensions and hence the control volume is in fact an area. The grey shaded area (named A_{cell}) in figure C.1/a) represents this control area. To demonstrate the consistency it shall be shown that the discretized surface integral over ∂A_{cell} will be the exact surface integral, as long as the integrand is a linear function in x and y . The following has to be demonstrated:

$$\oint_{\partial A_{cell}} \phi(x, y) ds = \left[\oint_{\partial A_{cell}} \phi(x, y) ds \right] = \int_{A_{cell}} (\nabla \phi(x, y)) dA, \phi(x, y) = a_0 + a_1 x + a_2 y \quad (C.1)$$

The control volume is constructed by connecting the centers of the edges and the centers of the triangles. Normally surface integrals are evaluated along edges. The values are linearly reconstructed on the middle of an edge. Both normal vectors are combined to a single one, which will then be multiplied by the reconstructed values (see figure C.1/b). To demonstrate consistence adjacent triangles to a node shall be investigated, instead of adjacent nodes (see figure C.1/c).

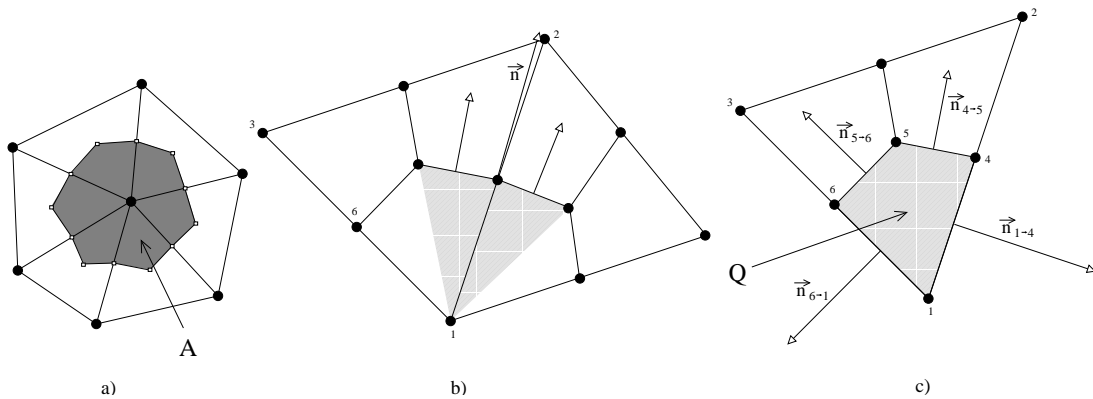


Figure C.1: A node centered control volume

The nodes of a single triangle are given in Cartesian coordinates:

$$\mathbf{r}_{1j} = \begin{pmatrix} x_{1j} \\ y_{1j} \end{pmatrix}, \quad \mathbf{r}_{2j} = \begin{pmatrix} x_{2j} \\ y_{2j} \end{pmatrix}, \quad \mathbf{r}_{3j} = \begin{pmatrix} x_{3j} \\ y_{3j} \end{pmatrix} \quad (\text{C.2})$$

The auxiliary points (4, 5, 6) can be computed by simple averaging.

$$\begin{aligned} \mathbf{r}_{4j} &= \frac{1}{2}(\mathbf{r}_{1j} + \mathbf{r}_{2j}) \\ \mathbf{r}_{5j} &= \frac{1}{3}(\mathbf{r}_{1j} + \mathbf{r}_{2j} + \mathbf{r}_{3j}) \\ \mathbf{r}_{6j} &= \frac{1}{2}(\mathbf{r}_{1j} + \mathbf{r}_{3j}) \end{aligned} \quad (\text{C.3})$$

Values for ϕ on the nodes 1, 2, 3 are directly available ($\phi_1, \phi_{2j}, \phi_{3j}$). The index j denotes the triangle in question (there are n adjacent triangles to node 1). Everything on the node 1 does not have the index j , since the node is the same for all triangles. Values on the auxiliary points can once again be computed by averaging the values on the nodes.

$$\phi_{4j} = \frac{1}{2}(\phi_1 + \phi_{2j}), \quad \phi_{5j} = \frac{1}{3}(\phi_1 + \phi_{2j} + \phi_{3j}), \quad \phi_{6j} = \frac{1}{2}(\phi_1 + \phi_{3j}) \quad (\text{C.4})$$

The surface integral for all quads Q_j can be exactly computed:

$$\begin{aligned} \oint_{\partial Q_j} \phi \, ds &= \frac{\phi_1 + \phi_{4j}}{2} (\mathbf{n}_{1 \rightarrow 2})_j + \frac{\phi_{4j} + \phi_{5j}}{2} (\mathbf{n}_{4 \rightarrow 5})_j \\ &\quad + \frac{\phi_{5j} + \phi_{6j}}{2} (\mathbf{n}_{5 \rightarrow 6})_j + \frac{\phi_{6j} + \phi_1}{2} (\mathbf{n}_{6 \rightarrow 1})_j \\ &= \int_Q (\nabla \phi) \, dA = Q_j \cdot (\nabla \phi) \end{aligned} \quad (\text{C.5})$$

Since the gradient of a linear function is constant the integral over the area of the quad from (C.5) reduces to a simple product. The sum over all quads Q_j will, in case of a constant gradient, leads to the following:

$$\sum_{j=1}^n \left(\oint_{\partial Q_j} \phi \, dA \right) = \sum_{j=1}^n (Q_j \cdot (\nabla \phi)) = A \cdot (\nabla \phi) \quad (\text{C.6})$$

The integral over the quad Q_j can be split into two parts. Firstly the contribution, as it would result from normal edgewise contributions (\mathbf{res}_j) and secondly the remainder to the exact integral (\mathbf{err}_j). With the normal vectors from figure C.1

$$\begin{aligned} (\mathbf{n}_{1 \rightarrow 2})_j &= \begin{pmatrix} \frac{y_{2j} - y_{1j}}{2} \\ \frac{x_{1j} - x_{2j}}{2} \end{pmatrix} & (\mathbf{n}_{4 \rightarrow 5})_j &= \begin{pmatrix} \frac{y_{3j}}{6} - \frac{y_{1j} + y_{2j}}{3} \\ \frac{x_{1j} + x_{2j}}{6} - \frac{x_{3j}}{3} \end{pmatrix} \\ (\mathbf{n}_{5 \rightarrow 6})_j &= \begin{pmatrix} \frac{y_{1j} + y_{3j}}{6} - \frac{y_{2j}}{3} \\ \frac{x_{2j}}{3} - \frac{x_{1j} + x_{3j}}{6} \end{pmatrix} & (\mathbf{n}_{6 \rightarrow 1})_j &= \begin{pmatrix} \frac{y_{1j} - y_{3j}}{2} \\ \frac{x_{3j} - x_{1j}}{2} \end{pmatrix} \end{aligned} \quad (\text{C.7})$$

the following can be written for \mathbf{res}_j and \mathbf{err}_j .

$$\begin{aligned}
\oint_{\partial Q_j} \phi dA &= \mathbf{res}_j + \mathbf{err}_j \\
\mathbf{res}_j &= \frac{\phi_1 + \phi_{2_j}}{2} (\mathbf{n}_{4 \rightarrow 5})_j + \frac{\phi_1 + \phi_{3_j}}{2} (\mathbf{n}_{5 \rightarrow 6})_j \\
\mathbf{err}_j &= \left(\begin{array}{c} \frac{\phi_{6_j}(y_1+y_{2_j})-\phi_{4_j}(y_1+y_{3_j})}{6} + \frac{\phi_{5_j}(y_{3_j}-y_{2_j})+\phi_1(y_{2_j}-y_{3_j})}{4} + \frac{\phi_{4_j}y_{2_j}-\phi_{6_j}y_{3_j}}{3} \\ \frac{\phi_{4_j}(x_1+x_{3_j})-\phi_{6_j}(x_{2_j}+x_1)}{6} + \frac{\phi_{5_j}(x_{2_j}-x_{3_j})+\phi_1(x_{3_j}-x_{2_j})}{4} + \frac{\phi_{6_j}x_{3_j}-\phi_{4_j}x_{2_j}}{3} \end{array} \right) \\
&= \frac{1}{12} \left(\begin{array}{c} 5\phi_1(y_{2_j} - y_{3_j}) + \phi_{3_j}y_1 + \phi_{2_j}y_{2_j} - \phi_{2_j}y_1 - \phi_{3_j}y_{3_j} \\ 5\phi_1(x_{3_j} - x_{2_j}) + \phi_{2_j}x_1 + \phi_{3_j}x_{3_j} - \phi_{3_j}x_1 - \phi_{2_j}x_{2_j} \end{array} \right) \quad (\text{C.8})
\end{aligned}$$

If it is possible to show that the sum of \mathbf{err}_j is zero, the consistency of the discretization has been demonstrated.

$$\begin{aligned}
&\sum_{j=1}^n \mathbf{err}_j &&= 0 \\
\Leftrightarrow &\sum_{j=1}^n \left(\begin{array}{c} 5\phi_1(y_{2_j} - y_{3_j}) + \phi_{3_j}y_1 + \phi_{2_j}y_{2_j} - \phi_{2_j}y_1 - \phi_{3_j}y_{3_j} \\ 5\phi_1(x_{3_j} - x_{2_j}) + \phi_{2_j}x_1 + \phi_{3_j}x_{3_j} - \phi_{3_j}x_1 - \phi_{2_j}x_{2_j} \end{array} \right) &&= 0 \\
\Leftrightarrow &5\phi_1 \left(\sum_{j=1}^N y_{2_j} - \sum_{j=1}^N y_{3_j} \right) + y_1 \left(\sum_{j=1}^N \phi_{3_j} - \sum_{j=1}^N \phi_{2_j} \right) + \sum_{j=1}^N \phi_{2_j}y_{2_j} - \sum_{j=1}^N \phi_{3_j}y_{3_j} &&= 0 \\
\wedge &5\phi_1 \left(\sum_{j=1}^N x_{3_j} - \sum_{j=1}^N x_{2_j} \right) + x_1 \left(\sum_{j=1}^N \phi_{2_j} - \sum_{j=1}^N \phi_{3_j} \right) + \sum_{j=1}^N \phi_{3_j}x_{3_j} - \sum_{j=1}^N \phi_{2_j}x_{2_j} &&= 0 \\
\Leftarrow &\sum_{j=1}^N y_{2_j} = \sum_{j=1}^N y_{3_j} \quad \wedge \quad \sum_{j=1}^N x_{3_j} = \sum_{j=1}^N x_{2_j} \quad \wedge \quad \sum_{j=1}^N \phi_{3_j} = \sum_{j=1}^N \phi_{2_j} \\
&\wedge \quad \sum_{j=1}^N \phi_{2_j}y_{2_j} = \sum_{j=1}^N \phi_{3_j}y_{3_j} \quad \wedge \quad \sum_{j=1}^N \phi_{3_j}x_{3_j} = \sum_{j=1}^N \phi_{2_j}x_{2_j} && \quad (\text{C.9})
\end{aligned}$$

Due to the cyclic nature of the sum it is possible to write:

$$x_3 = \begin{pmatrix} x_{2_2} \\ x_{2_3} \\ \vdots \\ x_{2_n} \\ x_{2_1} \end{pmatrix}, \quad y_3 = \begin{pmatrix} y_{2_2} \\ y_{2_3} \\ \vdots \\ y_{2_n} \\ y_{2_1} \end{pmatrix}, \quad \phi_3 = \begin{pmatrix} \phi_{2_2} \\ \phi_{2_3} \\ \vdots \\ \phi_{2_n} \\ \phi_{2_1} \end{pmatrix} \quad (\text{C.10})$$

Using (C.10) it is obvious, that (C.9) is true. Thus the discretization is valid for an arbitrary triangulation.

