THESE ARE NOTES TO THE NUMERICAL SOLUTION OF ADVECTION-CONVECTION-DIFFUSION HEAT- AND MASSFLOWS IN AN ACADEMIC STRATIFIED STORAGE TANK FOR SEASONAL STORAGE OF HEAT.

Nothing in this document can be considered as trustfull. Although I have spent considerable time on checking formulas against the book of Patankar[1] *every line of text and/or formulas in these Notes may contain errors!!.* These Notes were written down for my personal use and were not finished at the time of their release: they were modified at almost a daily rate. Use these Notes at your own risk. General advise is to read thoroughly the book of Patankar and recalculate all formulas before actually continue development of the source code and associated algorithms.

Dr. J.A. (Hans) Piest
Groningen, April 1st 2015

1 Governing equations

To estimate the efficiency loss due to convection in a stratified tank in the temperature range 50 – 100°C the Navier-Stokes equations under simultaneous conservation of heat and mass are to be solved. For Newtonion fluids, such as water, the equations read[1]:

mass conservation/continuity equation:

$$\partial_t \rho + \nabla\cdot(\rho\mathbf{v}) = S_\rho \tag{1.1}$$

momentum conservation/Navier-Stokes:

$$\partial_t(\rho\mathbf{v}) + \nabla\cdot(\rho\mathbf{v}\otimes\mathbf{v}) = -\nabla p + \nabla\cdot\mu\left(\nabla\otimes\mathbf{v} + (\nabla\otimes\mathbf{v})^T\right) + \nabla(\lambda\nabla\cdot\mathbf{v}) + \rho\mathbf{g} \tag{1.2}$$

conservation of energy:

$$\partial_t(\rho C_p T - p) + \nabla\cdot((\rho C_p T - p)\mathbf{v}) = \nabla\cdot(\alpha\rho C_p\nabla T) + \Phi + C_p S_T. \tag{1.3}$$

In the case of a stratified storage tank and working in the temperature range defined a number of approximations are acceptable. The fluid flow and heat transfer phenomena assumed can be summarized in the following hypotheses and restrictions:

i cylindrical symmetry:

$$\begin{cases} \mathbf{x} & = r\hat{\mathbf{r}} + z\hat{\mathbf{z}}, \\ \mathbf{v} & = v_r\hat{\mathbf{r}} + v_z\hat{\mathbf{z}}, \\ \mathbf{g} & = g_z\hat{\mathbf{z}}, \\ r\nabla & = r\hat{\mathbf{r}}\partial_r + \hat{\boldsymbol{\varphi}}\partial_\varphi + r\hat{\mathbf{z}}\partial_z, \\ \partial_\varphi\hat{\mathbf{r}} & = \hat{\boldsymbol{\varphi}}, \\ \partial_\varphi\hat{\boldsymbol{\varphi}} & = -\hat{\mathbf{r}}, \quad \text{furthermore,} \\ \partial_\varphi\xi & = 0 \quad \text{for } \xi = r, z, v_r, v_z, \lambda, \mu, \alpha, p, C_p, T, \rho, \end{cases} \tag{1.4}$$

ii constant physical properties: $\partial_t\xi = 0$ and $\nabla\xi = 0$ except for $\xi = \mathbf{v}, T, p$,

[1] Notation used: $\partial_\xi \equiv \dfrac{\partial}{\partial\xi}$, $\partial^2_\xi \equiv \dfrac{\partial^2}{\partial\xi^2}$, $\partial^2_{\xi\phi} \equiv \dfrac{\partial^2}{\partial\xi\partial\phi}$, and $\partial_{\xi,\epsilon} \equiv \left(\dfrac{\partial}{\partial\xi}\right)_{\epsilon=\text{constant}}$.

iii incompressible fluid: buoyancy modeled via Boussinesq, work to heat[2]: $\rho C_p T - p \approx \rho C_p T$,

iv laminar flow,

v neglected viscous dissipation: $\lambda = 0$ and $\Phi = 0$.

Under these conditions the governing equations take the explicit forms (see Appendix A for derivations):

mass conservation/continuity equation:

$$\rho \left(v_r + r\partial_r v_r + r\partial_z v_z \right) = r S_\rho, \tag{1.5}$$

momentum conservation/Navier-Stokes:

$\hat{\mathbf{r}}$-direction:
$$\left(r\rho(\partial_t + v_r\partial_r + v_z\partial_z) + rS_\rho - \mu(r\partial_r^2 + \partial_r + r\partial_z^2) \right) v_r = -r\partial_r p, \tag{1.6}$$

$\hat{\mathbf{z}}$-direction:
$$\left(r\rho(\partial_t + v_r\partial_r + v_z\partial_z) + rS_\rho - \mu(r\partial_r^2 + \partial_r + r\partial_z^2) \right) v_z = -r(\partial_z p - \rho g_z \beta(T - T_0)), \tag{1.7}$$

conservation of energy:

$$\left(r\rho(\partial_t + v_r\partial_r + v_z\partial_z) + rS_\rho - \alpha\rho(r\partial_r^2 + \partial_r + r\partial_z^2) \right) T = r S_T. \tag{1.8}$$

For definition of coordinate axes etc. see Figure D.1. Prior to the discretization procedure the above equations are brought to a uniform convection-diffusion equation[1]:

$$r\partial_t(\rho\Phi) + r\nabla\cdot(\rho\mathbf{v}\Phi) = r\nabla\cdot(\Gamma\nabla\Phi) + rS, \tag{1.9}$$

which, under the constraints and approximations for this simulation explicitly reads:

$$\left(r\rho(\partial_t + v_r\partial_r + v_z\partial_z) + \rho\left(v_r + r\partial_r v_r + r\partial_z v_z\right) - \Gamma(r\partial_r^2 + \partial_r + r\partial_z^2) \right) \Phi = rS, \tag{1.10}$$

with Φ, Γ, and S defined as presented in Table 1.1:

[2]Generally $pV \sim (C_p - C_v)T \approx 0 \Rightarrow \partial_t p \approx 0$ because for liquids and solids $C_p \approx C_v$ due to very small volume changes ($\delta V = \beta V \delta T$) upon heating, furthermore, $\beta = -\frac{1}{\rho}\left(\frac{\partial\rho}{\partial T}\right)_p$.

Table 1.1: Values of Φ, Γ, and S in the cylindrical coordinate convection-diffusion equation representation of the governing equations.

equation	Φ	Γ	S
continuity	1	0	S_ρ
momentum $\hat{\mathbf{r}}$-direction:	v_r	μ	$-\partial_r p$
momentum $\hat{\mathbf{z}}$-direction:	v_z	μ	$-\partial_z p + \rho g_z \beta (T - T_0)$
energy conservation:	T	$\alpha\rho$	S_T

2 Discretization

2.1 Governing equation

Discretization of the governing equations in cylindrical symmetry is based on [1]. The computational domain is discretized into a finite number of volumes which not necessarily have to be of equivalent sizes. This grid is referred to as the *centered grid*. At the gridcentres, indicated with B(ackward), F(orward), U(p), D(own), and P(osition) in Figures 2.1 and D.1, the scalars p and T are evaluated. In addition two *staggered grids* in the $\hat{\mathbf{r}}$- and $\hat{\mathbf{z}}$-direction are used for evaluation of the components of the velocity vector $\mathbf{v} = v_r\hat{\mathbf{r}} + v_z\hat{\mathbf{z}}$.

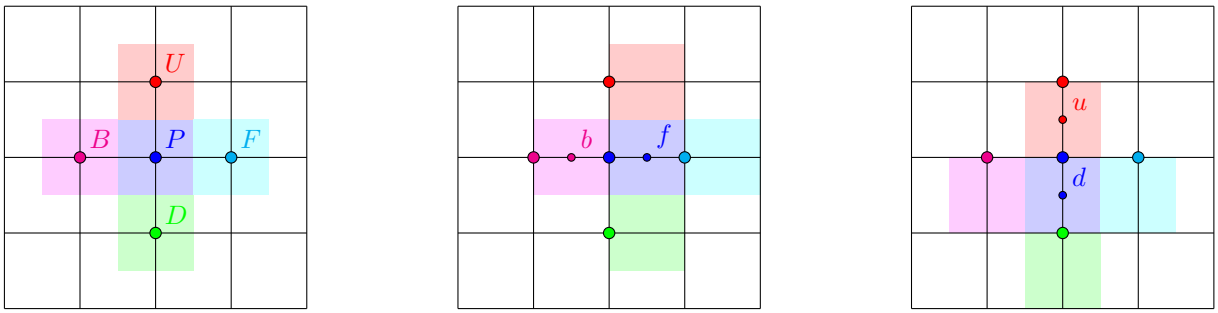


Figure 2.1: Centregrid (A); r-staggered grid (B); z-staggered grid (C).

Without staggering, *e.g.*, the discretized continuity equations demands that the *velocities* at *alternate* gridpoints sum up to zero and not those of adjacent ones. This leads to the – unwanted – result that once a solution obeying the continuity equation has been found, any alternating grid can be superposed without violating continuity. In the staggered grid the velocity components are evaluated at the *faces* of the control volumes, and such that staggering for the $\hat{\mathbf{r}}$ component is only in the $\hat{\mathbf{r}}$-direction and analogous for the $\hat{\mathbf{z}}$ component. This has been depicted in Figure D.1 with the arrows v_b, v_f, and v_d, v_u. Thus, ve-

locities are directed from gridcentre to gridcentre. Now the solution to the continuity equation contains components of *adjacent* gridpoints, thus, preventing for a wavy, alternating, grid. Throughout this document lower case indices u, d, f, b refer to values at faces of centregrid control volumes, *i.e.,* the centres of the staggered grid, whereas uppercase indices U, D, F, B, P indicate the values at the centregrid points. Integration and discretization follows the method of Patankar[1], which is shortly explained in Appendix B.

2.2 Computation of the temperature field on the centrepoint grid

The heat transfer due to diffusion and convection is computed with already known pressure and velocity field. In subsection 2.3 it will be explained how to obtain these fields. The discretized version of Eq. 1.10 is evaluated around centregrid point P:

$$\frac{\rho_P^0}{\Delta t}(T_P - T_P^0)V_P + \sum_{\omega' \in \{U,D,B,F\}} a_{\omega'}(T_P - T_{\omega'}) = ((S_{TP} - S_{\rho P} - S_{\rho C})T_P + S_{TC})V_P. \tag{2.1}$$

Note that with respect to Eq. (1.10) the discretized version has been divided by a factor r_P. Here, V_P is volume of the control volume. The coefficients $a_{\omega'}$ are functions of the the parameters:

$$F_\xi = \rho_P(\mathbf{v}_\xi \cdot \hat{\mathbf{n}}_{A_\xi})A_\xi, \qquad \text{and,} \tag{2.2}$$

$$D_\xi = \frac{\Gamma_\xi}{|\boldsymbol{\delta}_\xi|}A_\xi, \tag{2.3}$$

evaluated at the faces of control volume V_P. For more details read Appendix B.
Eq. (2.1) can be solved at gridpoint P for T_P in terms of its values at neighbouring centregrid points and using its value at the previous timestep T_P^0. The values at faces are evaluated by second or higher order interpolation functions of the Peclet number Pe at face ξ:

$$\mathrm{Pe}_\xi \equiv \frac{F_\xi}{D_\xi}. \tag{2.4}$$

Thus, eq. (2.1) is rewritten for gridpoint P:

$$\mathbf{a}_T \cdot \mathbf{T}^{(P)} \equiv \sum_{\omega \in \{P,B,F,U,D\}} a_{\omega,T} T_\omega = b_T, \tag{2.5}$$

The coefficients $\mathbf{a}_T$ follow:

$$a_{\omega',T} = -(D_\xi A(|\,\mathrm{Pe}_\xi\,|) + \max(-F_\xi, 0)),$$

for: $(\omega', \xi) \in \{(U,u),(D,d),(B,b),(F,f)\}$; furthermore,

$$a_{P,T} = -\left(a_{B,T} + a_{F,T} + a_{D,T} + a_{U,T}\right) + \frac{\rho V_P}{\Delta t} - (S_P - S_{\rho P} - S_{\rho C})V_P, \tag{2.6}$$

and:

$$b_T = S_C V_P + \frac{\rho V_P}{\Delta t} T_P^0. \tag{2.7}$$

The superscript ^{0} denotes the value of a variable in the previous timestep. Because the specific weight ρ is taken constant in time and over the volume of the tank under the Boussinesq approximation its superscript ^{0} and subscript $_P$ are dropped. With $\mathbf{a}_T$ and b_T computed from known values of the previous timestep, $\mathbf{T}^{(P)}$ can be evaluated at the current timestep for centregridpoint P.

2.3 Computation of the flowfield on the staggered r and z grids

Need for a staggered grid

The flowfield is computed on the r- and z-staggered grids from the pressure gradient of the known pressurefield. The staggered grid is necessary to keep the resolution of the computation on the same order as that of the temperature computation. If the flowfield was computed on the centregrid for, *e.g.*, gripdoint Pi and faces b and f on the midpoints in between B and P and P and F, it would yield:

$$v_r \sim \frac{\Delta p}{\Delta r} \sim p_b - p_f = \frac{(p_B + p_P) - (p_P + p_E)}{2} = \frac{p_B - p_E}{2}. \tag{2.8}$$

Thus, the momentum equation contains information of two alternate grid points. Note that the assumption of midpoints in between B and P and P and F is made here for algebraic convenience. The issue of using information only from alternating gridpoints does not go away if chosing faces to controlvolume P that are arbitrary in between B and F. It just becomes more difficult to illustrate the problem.

In fact, the pressure is taken from a coarser grid than the other variables such as temperature. That diminishes the accuracy of the flowfield computed. However, also another problem occurs. An alternating pressurefield, *e.g.*, $\ldots 5 - 10 - 5 - 10 - 5 - \ldots$, would be felt as if it were a *constant* pressurefield, because the alternate values are mutually equal to each other. This means that any kind of checkerboard field

pressure pattern, such as:

$$\begin{array}{ccccccc} \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \iddots \\ \cdots & p_1 & p_2 & p_1 & p_2 & p_1 & \cdots \\ \cdots & p_3 & p_4 & p_3 & p_4 & p_3 & \cdots \\ \cdots & p_1 & p_2 & p_1 & p_2 & p_1 & \cdots \\ \cdots & p_3 & p_4 & p_3 & p_4 & p_3 & \cdots \\ \cdots & p_1 & p_2 & p_1 & p_2 & p_1 & \cdots \\ \iddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array} \tag{2.9}$$

would be seen as a constant pressurefield and not compute any flow. Of course, this is unwanted. The degeneracy is lifted when using staggered r and z grids. *E.g.*, on an r-staggered grid the flows are computed on the forward and backward surfaces of centregrid point P. The gradient at the surfaces scale with the differences $p_P - p_F$ for the forward surface S_f and with $p_B - p_P$ for the backward surface S_b. Thus, the flows contain information of subsequent centregridpoints. The computational accuracy is now again similar to that of the computation of the temperature field. Moreover, a checkerboard pressurefield is recognized as such. For illustrations of the centregrid and staggered grids see Figures D.2 and D.3.

Formulation for v_r and v_h Analogously to Eqs. (2.6) and (2.7) which were defined for the centregrid the discretized flowfield equations for the staggered grid follow the same procedure and yield:

$$\begin{aligned} \mathbf{a}_{v_r,f} \cdot \mathbf{v}_{r,f} &\equiv \sum_{\omega \in \{f \text{ \& neighbours}\}} a_{\omega,v_r} v_{r,\omega} = b_{v_r} + (p_P - p_F) A_f, \\ \mathbf{a}_{v_h,d} \cdot \mathbf{v}_{r,d} &\equiv \sum_{\omega \in \{d \text{ \& neighbours}\}} a_{\omega,v_h} v_{r,\omega} = b_{v_h} + (p_P - p_D) A_d. \end{aligned} \tag{2.10}$$

Here, A_f and A_d are the forward and downward surfaces of the centregrid control volume around P containing the respective staggered gridpoints f and d. The 'neighbours' run over the staggered gridpoints neighbouring to f and d, respectively. The pressure gradient is removed from the source term quantities S_C and S_P. The pressure is an essential variable that has to be computed, therefore, they are explicitly introduced instead of buried in the source terms.
Normally, the pressure field is not known *a priori*. This means it has to be estimated. The flows computed from such an estimate do not generally satisfy the continuity equation. The aim therefore is to start with guessed pressure field p' and try to get this converged to the pressurefield p that corresponds to the

continuity equation. Analogously, the flow components will converge to physically acceptable values:

$$p = p' + \delta p, \tag{2.11}$$

$$v_r = v'_r + \delta v_r, \tag{2.12}$$

$$v_h = v'_h + \delta v_h. \tag{2.13}$$

The parameters $\delta p, \delta v_r, \delta v_h$ are merely corrections to the guessed (primed) values such that the converged values p, v_r, v_h are obtained. These differences fulfill the equation:

$$\mathbf{a}_{v_r,f} \cdot \delta \mathbf{v}_{r,f} = \sum_{\omega \in \{f \text{ \& neighbours}\}} a_{\omega,v_r} \delta v_{r,\omega} = (\delta p_P - \delta p_F) A_f. \tag{2.14}$$

$$\tag{2.15}$$

Because the exact values of v_r at all grid points are not *a priori* known – in fact, this procedure describes how to compute them – we assume that for the neighboring cells they are equal to their guessed values, thus:

$$\sum_{\omega \in \{\text{neighbours}\}} a_{\omega,v_r} \delta v_{r,\omega} = 0 \quad \Rightarrow \quad \mathbf{a}_{v_r,f} \cdot \delta \mathbf{v}_{r,f} = a_{f,v_r} \delta v_{r,f} = (\delta p_P - \delta p_F) A_f, \tag{2.16}$$

$$\Rightarrow \quad \delta v_{r,f} = d_f (\delta p_P - \delta p_F) \qquad \text{with: } d_f \equiv \frac{A_f}{a_{f,v_r}}. \tag{2.17}$$

This brings the advantage that the computation of corrections depends only on the current control volume. The final solution is a set of vanishing corrections, therefore is an element of the set of solutions where all neigbouring corrections vanish. It is in this subspace of corrections that the solution for the flowfield is sought for. For a more rigid motivation of this procedure see, *e.g.*, Patankar[1].
Eq. 2.17 will be referred to as *velocity correction equation*. It relates the pressure corrections δP_P and δP_F with the guessed value $v'_{r,f}$ to obtain the locally correct velocity $v_{r,f}$:

$$v_{r,f} = v'_{r,f} + d_f (\delta p_P - \delta p_F), \tag{2.18}$$

and analogously for the back- down- and upward directions:

$$v_{r,b} = v'_{r,b} + d_b (\delta p_B - \delta p_P), \tag{2.19}$$

$$v_{h,d} = v'_{h,d} + d_d (\delta p_P - \delta p_D), \tag{2.20}$$

$$v_{h,u} = v'_{h,u} + d_u (\delta p_D - \delta p_U). \tag{2.21}$$

Once the pressure corrections have been obtained, the velocity flow field can be computed.

Formulation for p

The aim is to find a flow- and pressure field that is consistent with the continuity equation Eq. (1.1). As indicated in the paragraph before knowledge of the pressure fields facilitates computation of the flowfield. Therefore, a pressure correction equation is derived so that the guessed pressures at centregridpoints can be corrected and finally converge to a pressure field that is a solution of the continuity equation. The discretized equivalent of this equation is given in Eq. (B.4). Under the approximations for cylindrical symmetry, Boussinesq, etc. that are applied in the research presented in this Document it reduces to the following discretized continuity equation:

$$\rho \sum_{\xi \in \{f,b,u,d\}} (\mathbf{v}_{x,\xi} \cdot \hat{\mathbf{n}}_{A_\xi}) A_\xi = \rho(v_{r,f} A_f - v_{r,b} A_b + v_{h,d} A_d - v_{h,u} A_u) = S_\rho V_P, \tag{2.22}$$

This relation is very intuitive to interpret and states that the massflow through the surfaces of the control volume around P must equal the inflowing mass due to the source at P.

Substitution of the velocity correction equations for the respective surfaces yields:

$$\mathbf{a}_{p,P} \cdot \delta \mathbf{p}_P = b_{p,P}, \tag{2.23}$$

with:

$$\begin{aligned} a_{p,\omega} &= -\rho d_\xi A_\xi \qquad \text{for: } (\omega, \xi) \in \{(F,f),(B,b),(U,u),(D,d)\}, \\ a_{p,P} &= -\sum_{\omega \in \{F,B,D,U\}} a_{p,\omega} \\ b_{p,P} &= \rho(v'_{r,b} A_b - v'_{r,f} A_f + v'_{h,u} A_u - v'_{h,d} A_d) + S_\rho V_P. \end{aligned} \tag{2.24}$$

The parameter $b_{p,P}$ can be interpreted as a 'mass source'. If $b_{p,P} = 0$ the continuity equation is fulfilled for the guessed velocities and no pressure correction is needed.

2.4 Sources

Specific weight:

The source of specific weight is handled as follows:

following the standard definition of the derivative:

$$\dot{\rho} = \frac{V\dot{m} - m\dot{V}}{V^2} = \lim_{t \to 0} \frac{V(t)(m(t) - m(0)) - m(t)(V(t) - V(0))}{tV(t)^2}, \tag{2.25}$$

looking at a time independent sized control volume $V(t) = V(0) = V_P$ the above reduces to:

$$\dot{\rho} = \lim_{t \to 0} \frac{m(t) - m(0)}{t V_P}, \tag{2.26}$$

which becomes in terms of the initial specific weight ρ_0:

$$\dot{\rho} = \lim_{t \to 0} \frac{\rho_0 (V_P + \int_0^t \dot{V}_{\text{inflow}}(t) dt - V_P)}{t V_P} = \rho_0 \frac{\dot{V}_{\text{inflow}}}{V_P}, \tag{2.27}$$

where $\dot{V}_{\text{inflow}}$ denotes the rate of inflow into control volume V_P from an external source. This way, the source term to the continuity equation for the control volume at centregridpoint P – with $\rho = \rho_P^0$ and $V_{P,i}$ being the inflowrate in V_P – is written as:

$$S_\rho \big|_P = \dot{\rho} \big|_P = \frac{\rho_P^0 \dot{V}_{P,i}}{V_P}. \tag{2.28}$$

With this term the actual inflow or outflow of the tank is simulated. Each cell can be defined as source or sink this way by setting $\dot{V}_{P,i}$. Note that the rate of water flowing into the tank *must equal* the rate of flow out of the tank at all times:

$$\sum_{P \in \text{ all control volumes}} \dot{V}_{P,i} = \dot{V}_{\text{tank}} = 0. \tag{2.29}$$

Heat

Heat can be delivered either by inflowing water of a certain temperature via sinks and sources or via a heat-exchanger. Generally, the heat flowrate into a control volume V_P is taken constant over the time interval considered. It is derived for the heat source term S_T for a certain control volume V_P that:

$$P = \dot{H} = \rho C_p V_P \dot{T} = C_p V_P S_T \quad \Rightarrow \quad S_T = \frac{P}{C_p V_P}. \tag{2.30}$$

The power P denotes rate of heat flowing into the control volume, either via mass flow in a source or sink volume, or via a heat exchanger. In the first case, the power is computed as:

$$P = \rho \dot{V} C_p T_{\text{s}}. \tag{2.31}$$

Here T_{s} is the temperature of the inflowing water in the case of a source, and the temperature of the outflowing water, in the case of a sink. In the first case the water temperature, thus P, is independent of the temperature of the control volume. In the latter case, however, P is fully dependent on the control volume because the temperature of the outflowing water is always equal to the control volume temperature $T_{\text{s}} = T$.

3 Explicit formulation of the SIMPLE algorithm

The SIMPLE algorithm is the algorithm which has been developed [1] to solve flow problems in the control volume formulation. The acronym stands for *S*emi-*I*mplicit *M*ethod for *P*ressure-*L*inked *E*quations. The sequence of operations in this algorithm reads:

1. guess the pressure field p',
2. solve the momentum equations Eqs. (2.11),
3. solve the pressure corrections Eq. (2.23),
4. compute the new pressure $p = p' + \delta p$,
5. compute the new velocities using Eqs. (2.18) – (2.21),
6. solve the temperature equation,
7. start a new loop with the newly compute pressure as guessed pressure.

1) *Guess of the pressure field*
The first guess of the pressure field is done via the software: $p = 10^5$hPa (1 bar) defined everywhere. Depending on the situation computed a relaxation round can be performed to generate a starting point in time that fulfills already the continuity equation with a realistic pressure profile. This profile is stored and can be read in from a csv file. For different scenarias different initial input files can be used.

2) *Solve the momentum equations*
To solve the momentum equations Eqs. (2.17) these have to be written out explicitly for the staggered grids in the r and the h directions. In Figure 3.1 the definitions for specific parameters appearing in the momentum correction formulas are sketched for the r-staggered grid. The formulas for the h-staggered grid are derived analogously. The formulas for the neighbouring control volumes, where the subscript v_r has been dropped, are given as:

$$\begin{cases} a_{ff} &= -(D_F A(|\,\mathrm{Pe}_F\,|) + \max(-F_F, 0)), \\ a_{fb} &= -(D_P A(|\,\mathrm{Pe}_P\,|) + \max(-F_P, 0)), \\ a_{fu} &= -(D_{ffu} A(|\,\mathrm{Pe}_{ffu}\,|) + \max(-F_{ffu}, 0)), \\ a_{fd} &= -(D_{ffd} A(|\,\mathrm{Pe}_{ffd}\,|) + \max(-F_{ffd}, 0)). \end{cases} \tag{3.1}$$

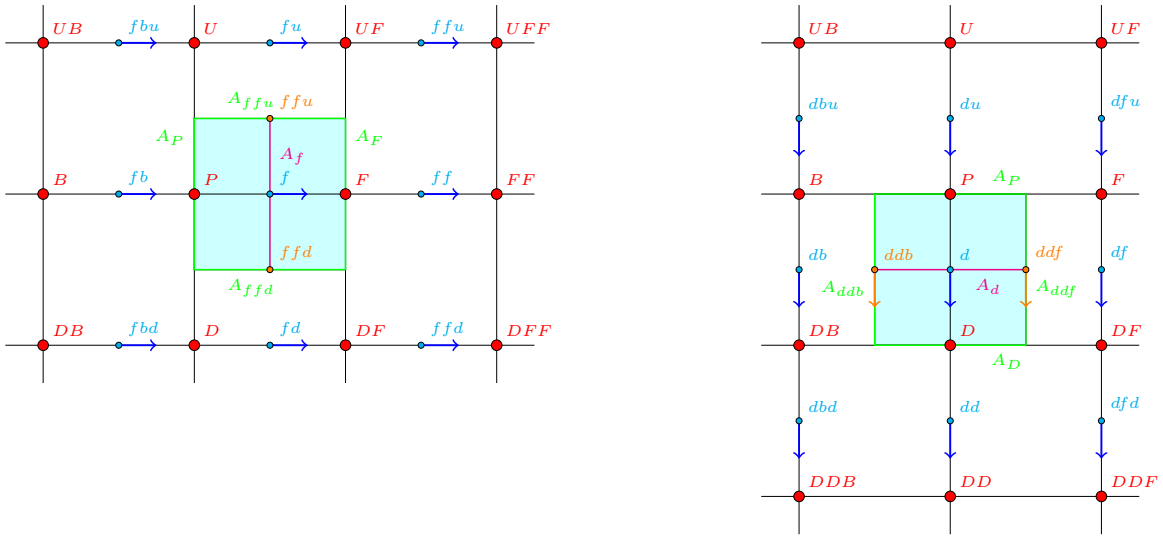


Figure 3.1: Definition of grid and variables for computation of v_r' on the r-staggered grid (left), and for computtion of v_h' on the h-staggered grid (right) .

Here Eqs. (2.2) and (2.3) have been applied. Furthermore:

$$\begin{cases} F_F & = A_F \dfrac{\rho}{r_{ff} - r_f} \begin{pmatrix} r_{ff} - r_F \\ r_F - r_f \end{pmatrix} \cdot \begin{pmatrix} v^0_{r,f} \\ v^0_{r,ff} \end{pmatrix} \\ F_P & = -A_P \dfrac{\rho}{r_f - r_{fb}} \begin{pmatrix} r_f - r_P \\ r_P - r_{fb} \end{pmatrix} \cdot \begin{pmatrix} v^0_{r,fb} \\ v^0_{r,f} \end{pmatrix} \\ F_{ffu} = F_{ffd} & = 0 \qquad \text{(due to: } \hat{\mathbf{n}} \perp \mathbf{v}^0_r\text{)}. \end{cases} \tag{3.2}$$

$$\begin{cases} D_F & = \dfrac{\mu^0_F}{r_{ff} - r_f} A_F, \\ D_P & = \dfrac{\mu^0_P}{r_f - r_{fb}} A_P, \\ D_{ffu} & = \dfrac{\mu^0_{ffu}}{h_f - h_{fu}} A_{ffu}, \\ D_{ffd} & = \dfrac{\mu^0_{ffd}}{h_{fd} - h_f} A_{ffd}, \end{cases} \quad \text{and:} \quad \begin{cases} \mathrm{Pe}_F = \dfrac{F_F}{D_F} & = \dfrac{\rho}{\mu^0_F} \begin{pmatrix} r_{ff} - r_F \\ r_F - r_f \end{pmatrix} \cdot \begin{pmatrix} v^0_{r,f} \\ v^0_{r,ff} \end{pmatrix} \\ \mathrm{Pe}_P = \dfrac{F_F}{D_F} & = \dfrac{\rho}{\mu^0_P} \begin{pmatrix} r_f - r_P \\ r_P - r_{fb} \end{pmatrix} \cdot \begin{pmatrix} v^0_{r,fb} \\ v^0_{r,f} \end{pmatrix} \\ \mathrm{Pe}_{ffu} = \dfrac{F_{ffu}}{D_{ffu}} & = 0 \\ \mathrm{Pe}_{ffd} = \dfrac{F_{ffd}}{D_{ffd}} & = 0. \end{cases} \tag{3.3}$$

In the above formulas the superscript ^{0} indicates quantities determined at the previous timestep. The formulation for v_h on the h-staggered grid is basically the same and can be obtained by circular substitution as indicated in the schemes of Figure 3.2 and with reference to Figure 3.1.

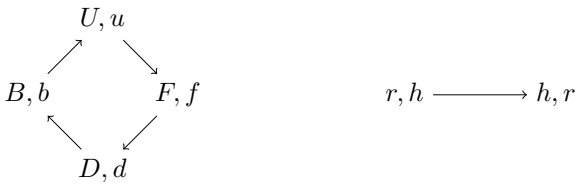


Figure 3.2: Index changes from v_r computation to v_h computation.

Now all the a's are known for the neighbouring points a_f at the center of the control volume can be computed:

$$a_f = -(a_{ff} + a_{fb} + a_{fu} + a_{fd}) + \frac{\rho V_f}{\Delta t} - (S_f - S_{\rho f} - S_{\rho C})V_f. \tag{3.4}$$

The source term for the r direction, except for the pressure gradient that is taken out of $S_f v_r + S_C$, vanishes, furthermore, the source for ρ: $S_{\rho f} - s_{\rho C} = S_\rho$ is also independent of the flow velocities. Therefore, and with the results for F, D, and Pe derived earlier, the above expression for a_f reduces to:

$$a_{f,r} = (D_F A(|\,\mathrm{Pe}_F\,|) + \max(-F_F, 0)) + (D_P A(|\,\mathrm{Pe}_P\,|) + \max(-F_P, 0)) + D_{ffu} + D_{ffd} + \left(\frac{\rho}{\Delta t} + S_\rho\right) V_f. \tag{3.5}$$

After definition of the a coefficients the b coefficient is computed:

$$b = S_C V_f + \frac{\rho V_f}{\Delta t} v^0_{r,f} = \frac{\rho V_f}{\Delta t} v^0_{r,f}. \tag{3.6}$$

Note that S_C doesn't contain the pressure gradient and with $\hat{\mathbf{h}}$ antiparallel to $\hat{\mathbf{z}}$:

$$\begin{cases} S_C = 0, & \hat{\mathbf{r}}\text{-direction}, \\ S_C = -\rho g_z \beta (T - T_0), & \hat{\mathbf{h}}\text{-direction}. \end{cases} \tag{3.7}$$

Using Eqs. 2.11, the previous values of p and $\mathbf{v}$ the coefficients a and b are computed and from that the new velocities $\mathbf{v}$.

3) *&* **4)** *Solve the δp equation*
Using the values of a_f and the new velocities from the previous section the values $a_{p,F} = -\rho d_f A_f$, $a_{p,P}$, $b_{p,P}$, are computed. This yields the new δp values. The new p_P is computed from: $p_P = p'_P + \delta p_P$.

5) *Compute new velocities $v = v' + d\Delta\delta p$*

6) *Solve the temperature equation*
Compute on the centregrid the vector $\mathbf{a}_T$ and scalar b_T and from that the temperature T_P at P from the new values of $\mathbf{v}$ and p and Eqs. (2.6)–(2.7). Treat the source independent: $S_C + S_P T = S_C = S_T$ and as indicated in Section 2.4.

This procedure is schemetically presented in the flowdiagram in Figure D.4.

4 Geometry

The system of water-storage tank is modelled with a cylindrical iron container of inner dimensions $\pi R^2_{\text{water}} H_{\text{water}} = \pi \times 1000^2 \times 2000$ cm^3 and wall thickness $d_{\text{Fe}} = 2$ cm. This tank contains the warm water stored during the season. The metal container is surrounded with isolation material of thickness $d_{\text{iso}} = 20$ cm. The system is completely covered with a layer of wet earth of d_{earth} =100 cm. A sketch of the tank with an indication of its dimensions is given in Figure D.5. Note that there are neither inlets nor outlets. Water inlet and outlet is modeled at the level of control volumes by definition of the source term S_ρ in eqs. (1.1) and (**??**).
The computational domain is divided into $96 \times 192 = 19602$ control volumes around $97 \times 193 = 18721$ centergrid points as shown in Figure D.6. In this figure the centergrid points are on the crossings of vertical and horizontal blue lines. Because the tankheight is taken twice the tankwidth the grid in the vertical direction is constructed by mirroring the grid for the horizontal direction and add it to its unmirrored version.
This grid is non-uniform with an increasing resolution near the water-metal transition and high enough resolution over the thickness of the isolation layer. An exponentially decaying function is used to compute the gridspacings. Ideally, a stretch W is divided in N successive intervals W_n with n running as $n = 1, 2, 3 \ldots N-1, N$, and such that:

$$W_{n+1} = W_n \exp(-B), \tag{4.1}$$

which describes exactly an exponential decaying gridinterval with increasing n. However, to keep control over the computational accuracy is is desired that $W > W_1 > W_N$ are defined beforehand. Thus, an explicit expression for W_n follows:

$$\begin{cases} W_n &= W_1 \exp\left(-B(n-1)\right), \quad \text{and:} \\ W_N &= W_1 \exp\left(-B(N-1)\right). \end{cases} \tag{4.2}$$

With predefined W_1 and W_N eqs. (4.2) hold one degree of freedom. However, also the stretch W is

predefined:

$$W = \sum_{n=1}^{N} W_n. \tag{4.3}$$

This extra constraint removes the last degree of freedom: with predefined W, W_1 and W_N both B and N are uniquely defined. Unfortunately, there is again another constraint: N must be integer. For arbitrary W, W_1 and W_N this is usually not the case. Therefore, the following procedure is practiced to determine the intervals W_n so that they approximate the ideal decaying behaviour of eq. (4.1).
From eqs. (4.2) it is deduced that:

$$\exp(-B) = \left(\frac{W_N}{W_1}\right)^{\frac{1}{N-1}}. \tag{4.4}$$

Furthermore, eq. (4.3) evaluates to:

$$\begin{aligned} W &= W_1 \sum_{n=1}^{N} \exp\left(-B(n-1)\right) \\ &= W_1 \exp(B) \left(\sum_{n=0}^{N} \exp\left(-Bn\right) - 1\right) \\ &= \frac{W_1 - W_N \exp\left(-B\right)}{1 - \exp\left(-B\right)}, \end{aligned} \tag{4.5}$$

where in the last step the equivalence $\sum_{i=1}^{M} x^i = (1 - x^{M+1})(1 - x)^{-1}$ and the result of eq. (4.4) have been exploited. With eqs. (4.4) and (4.5) the gridintervals W_n are approximated by:

Compute $\exp(-B)$ from eq. (4.5):

$$\exp(-B) = \frac{W - W_1}{W - W_N}, \tag{4.6}$$

subsequently, use this result to compute N from eq. (4.4):

$$N = 1 + \frac{\ln\left(\dfrac{W_1}{W_N}\right)}{\ln\left(\dfrac{W - W_N}{W - W_1}\right)}. \tag{4.7}$$

Because N computed this way is generally not an integer round it to its ceiling N_c, *i.e.,* the first integer higher or equal to N:

$$N_c = \text{ceil}(N), \tag{4.8}$$

which value is used to compute the approximated value $\exp(-B_c)$ from eq. (4.4):

$$\exp(-B_c) = \left(\frac{W_N}{W_1}\right)^{\frac{1}{N_c-1}}, \tag{4.9}$$

and from that the value W_c is computed:

$$W_c = \frac{W_1 - W_N \exp(-B_c)}{1 - \exp(-Bc)}. \tag{4.10}$$

This value is used to adjust the intervals W_n:

$$W_{c,n} = W_n \frac{W}{W_c} = W_1 \frac{W}{W_c} \exp(-B_c(n-1)) \\ (n = 1, 2, 3, \ldots, N_c - 1, N_c). \tag{4.11}$$

This yields exponentially decaying intervals with $W_{c,1} \approx W_1$ and $W_{c,N} \approx W_N$, moreover, the stretch W initially chosen is maintained. This is important because it allows for accurate positioning of stretches over region with different material and flow properties in the computational domain.
Note that in the limit for $W_N \to W_1$ eqs. (4.7) and (4.8) and eq. (4.10) yield equidistance spacing for the gridpoints:

eq. (4.7) in the limit for $W_N \to W_1$ yields:

$$\begin{aligned}
\lim_{W_N \to W_1} N &= \lim_{W_N \to W_1} 1 + \frac{\ln\left(\frac{W_1}{W_N}\right)}{\ln\left(\frac{W - W_N}{W - W_1}\right)} \\
&= \lim_{W_N \to W_1} 1 + \frac{\frac{\mathrm{d}}{\mathrm{d}W_N} \ln\left(\frac{W_1}{W_N}\right)}{\frac{\mathrm{d}}{\mathrm{d}W_N} \ln\left(\frac{W - W_N}{W - W_1}\right)} \\
&= \lim_{W_N \to W_1} \frac{W}{W_N} \\
&= \frac{W}{W_1},
\end{aligned}$$

of which result the ceiling is taken to obtain N_c:

$$N_c = \text{ceil}\left(\frac{W}{W_1}\right).$$

Moreover, eq. (4.10) evaluates in this limit to:

$$\begin{aligned}
\lim_{W_N \to W_1} W_c &= \lim_{W_N \to W_1} \frac{W_1 - W_N \left(\frac{W_N}{W_1}\right)^{\frac{1}{N_c-1}}}{1 - \left(\frac{W_N}{W_1}\right)^{\frac{1}{N_c-1}}} \\
&= \lim_{W_N \to W_1} \frac{\frac{\mathrm{d}}{\mathrm{d}W_N} W_N \left(\frac{W_N}{W_1}\right)^{\frac{1}{N_c-1}}}{\frac{\mathrm{d}}{\mathrm{d}W_N} \left(\frac{W_N}{W_1}\right)^{\frac{1}{N_c-1}}} \\
&= \lim_{W_N \to W_1} W_N N_c \\
&= W_1 N_c,
\end{aligned}$$

using eq. (4.11) and in the limiting case $\exp(-B_c) = 1$ it shows that:

$$W_{c,n} = W_n \frac{W}{W_c} = W_1 \frac{W}{W_1 N_c} = \frac{W}{N_c},$$

thus, yielding an equidistant gridpoint spacing. Note that this exercise is necessary once equidistant intervals are desired: direct coding of eqs. (4.7) and (4.10) will result in 'division by zero' errors.
In Figure D.7 the subsequent interval widths W_n are indicated for their index number n. In this case the following values were used and computed following the above computational scheme:

Table 4.1: Values for W, N_c, W_1, and W_N as initially defined and as computed and used for the grid definition.

	W / cm		N_c	W_1 / cm	W_N / cm
water	1000	initial		75.00	.30
		computed	73	74.09	.30
metal	2	initial		.30	.30
		computed	7	.29	.29
isolation	20	initial		.30	5.00
		computed	12	.28	4.73
earth	100	initial		5	75
		computed	4	4.07	61.1

References

[1] Suhas V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation/McGraw-Hill Book Company, 1980.

A Derivation of explicit governing equations in cylindrical coordinates

Use:

$$r\nabla = r\hat{\mathbf{r}}\partial_r + \hat{\boldsymbol{\varphi}}\partial_\varphi + r\hat{\mathbf{z}}\partial_z, \tag{A.1}$$

$$r^2\nabla^2 = r\partial_r(r\partial_r) + \partial_\varphi^2 + r^2\partial_z^2, \tag{A.2}$$

and for the divergence of a rank 2 tensor $\boldsymbol{\tau}$:

$$\begin{aligned} r^2\nabla\cdot(&\tau_{rr}\hat{\mathbf{r}}\otimes\hat{\mathbf{r}} + \tau_{r\varphi}\hat{\mathbf{r}}\otimes\hat{\boldsymbol{\varphi}} + \tau_{rz}\hat{\mathbf{r}}\otimes\hat{\mathbf{z}} + \\ &\tau_{\varphi r}\hat{\boldsymbol{\varphi}}\otimes\hat{\mathbf{r}} + \tau_{\varphi\varphi}\hat{\boldsymbol{\varphi}}\otimes\hat{\boldsymbol{\varphi}} + \tau_{\varphi z}\hat{\boldsymbol{\varphi}}\otimes\hat{\mathbf{z}} + \\ &\tau_{zr}\hat{\mathbf{z}}\otimes\hat{\mathbf{r}} + \tau_{z\varphi}\hat{\mathbf{z}}\otimes\hat{\boldsymbol{\varphi}} + \tau_{zz}\hat{\mathbf{z}}\otimes\hat{\mathbf{z}}) = r(&\partial_r(r\tau_{rr}) + \partial_\varphi\tau_{\varphi r} + r\partial_z\tau_{zr} - \tau_{\varphi\varphi})\hat{\mathbf{r}} + \\ &(\partial_r(r^2\tau_{r\varphi}) + r\partial_\varphi\tau_{\varphi\varphi} + r^2\partial_z\tau_{z\varphi} - r(\tau_{\varphi r} - \tau_{r\varphi}))\hat{\boldsymbol{\varphi}} + \\ &r(\partial_r(r\tau_{rz}) + \partial_\varphi\tau_{\varphi z} + r\partial_z\tau_{zz})\hat{\mathbf{z}}. \end{aligned} \tag{A.3}$$

furthermore, use the constraints as listed on page 1.

A.1 Continuity equation

$$\partial_t\rho + \nabla\cdot(\rho\mathbf{v}) = S_\rho \tag{A.4}$$

$$\Rightarrow$$

$$(r\hat{\mathbf{r}}\partial_r + \hat{\boldsymbol{\varphi}}\partial_\varphi + r\hat{\mathbf{z}}\partial_z)\cdot(\rho v_r\hat{\mathbf{r}} + \rho v_z\hat{\mathbf{z}}) = rS_\rho \tag{A.5}$$

$$\Rightarrow$$

$$\rho(r\partial_r v_r + r\partial_z v_z + v_r) = rS_\rho. \tag{A.6}$$

Remind that by virtue of the product rule for derivation terms have to be evaluated such as:

$$\begin{aligned} \hat{\boldsymbol{\varphi}}\cdot\partial_\varphi\rho v_r\hat{\mathbf{r}} &= v_r\hat{\boldsymbol{\varphi}}\cdot\hat{\mathbf{r}}\partial_\varphi\rho + \rho\hat{\boldsymbol{\varphi}}\cdot\hat{\mathbf{r}}\partial_\varphi v_r + \rho v_r\hat{\boldsymbol{\varphi}}\cdot\partial_\varphi\hat{\mathbf{r}} \\ &= 0 + 0 + \rho v_r, \end{aligned}$$

Because $\hat{\boldsymbol{\varphi}}\cdot\hat{\mathbf{r}} = 0$ and $\hat{\boldsymbol{\varphi}}\cdot\partial_\varphi\hat{\mathbf{r}} = \hat{\boldsymbol{\varphi}}\cdot\hat{\boldsymbol{\varphi}} = 1$.

A.2 Momentum equation

$$\partial_t(\rho\mathbf{v}) + \nabla\cdot(\rho\mathbf{v}\otimes\mathbf{v}) = -\nabla p + \nabla\cdot\mu\left(\nabla\otimes\mathbf{v} + (\nabla\otimes\mathbf{v})^T\right) + \nabla(\lambda\nabla\cdot\mathbf{v}) + \rho\mathbf{g} \tag{A.7}$$

$$\Rightarrow \qquad (\rho, \mu \text{ constant, } \lambda \text{ vanishing, Boussinesq})$$

$$r^2\rho(\partial_t\mathbf{v} + \nabla\cdot(\mathbf{v}\otimes\mathbf{v})) = r^2(-\nabla p + \mu\nabla^2\mathbf{v} + \rho\beta(T-T_0)\mathbf{g}) \tag{A.8}$$

$$\Rightarrow$$

$$\begin{aligned} r^2\rho(\partial_t(v_r\hat{\mathbf{r}} + v_z\hat{\mathbf{z}}) + \nabla\cdot(v_r^2\hat{\mathbf{r}}\otimes\hat{\mathbf{r}} + \\ v_rv_z(\hat{\mathbf{r}}\otimes\hat{\mathbf{z}} + \hat{\mathbf{z}}\otimes\hat{\mathbf{r}}) + v_z^2\hat{\mathbf{z}}\otimes\hat{\mathbf{z}})) &= r^2(-\nabla p + \mu\nabla^2(v_r\hat{\mathbf{r}} + v_z\hat{\mathbf{z}}) + \rho g_z\beta(T-T_0)\hat{\mathbf{z}}) \end{aligned} \tag{A.9}$$

$$\Rightarrow$$

$$\begin{cases} \hat{\mathbf{r}}: & \rho(r\partial_t v_r + \partial_r(rv_r^2) + r\partial_z(v_rv_z)) \\ \hat{\mathbf{z}}: & \rho(r\partial_t v_z + \partial_r(rv_rv_z) + r\partial_z(v_z^2)) \end{cases} = \begin{cases} -r\partial_r p + \mu(r\partial_r^2 + r\partial_z^2 + \partial_r)v_r \\ -r\partial_z p + \mu(r\partial_r^2 + r\partial_z^2 + \partial_r)v_z + r\rho g_z\beta(T-T_0) \end{cases} \tag{A.10}$$

$$\Rightarrow$$

$$\begin{cases} \hat{\mathbf{r}}: & \rho(r\partial_t v_r + v_r^2 + 2rv_r\partial_r v_r + rv_r\partial_z v_z + rv_z\partial_z v_r) \\ \hat{\mathbf{z}}: & \rho(r\partial_t v_z + rv_r\partial_r v_z + rv_z\partial_r v_r + v_rv_z + 2rv_z\partial_z v_z) \end{cases} = \begin{cases} -r\partial_r p + \mu(r\partial_r^2 + r\partial_z^2 + \partial_r)v_r \\ -r\partial_z p + \mu(r\partial_r^2 + r\partial_z^2 + \partial_r)v_z + r\rho g_z\beta(T-T_0) \end{cases} \tag{A.11}$$

$$\Rightarrow$$

$$\begin{cases} \hat{\mathbf{r}}: & \rho(r\partial_t + rv_r\partial_r + rv_z\partial_z + v_r + r\partial_r v_r + r\partial_z v_z)v_r \\ \hat{\mathbf{z}}: & \rho(r\partial_t + rv_r\partial_r + rv_z\partial_z + v_r + r\partial_r v_r + r\partial_z v_z)v_z \end{cases} = \begin{cases} -r\partial_r p + \mu(r\partial_r^2 + r\partial_z^2 + \partial_r)v_r \\ -r\partial_z p + \mu(r\partial_r^2 + r\partial_z^2 + \partial_r)v_z + r\rho g_z\beta(T-T_0) \end{cases} \tag{A.12}$$

$$\Rightarrow \qquad (\text{use result of A.6})$$

$$\begin{cases} \hat{\mathbf{r}}: & (r\rho(\partial_t + v_r\partial_r + v_z\partial_z) + rS_\rho)v_r \\ \hat{\mathbf{z}}: & (r\rho(\partial_t + v_r\partial_r + v_z\partial_z) + rS_\rho)v_z \end{cases} = \begin{cases} -r\partial_r p + \mu(r\partial_r^2 + r\partial_z^2 + \partial_r)v_r \\ -r\partial_z p + \mu(r\partial_r^2 + r\partial_z^2 + \partial_r)v_z + r\rho g_z\beta(T-T_0) \end{cases} \tag{A.13}$$

A.3 energy equation

First realize that:

$$\begin{aligned} H &\equiv U + pV_m \\ &\Rightarrow \\ \delta H &= \delta U + p\delta V_m + V_m\delta p \qquad\qquad \Rightarrow \\ \delta H &\approx \delta U. \end{aligned}$$

For solids and liquids the volume change δV_m is very small and can be neglected. Furthermore, with the molar volume V_m very small also the term $V_m\delta p$ approximately vanishes. For the heat capacities at constant pressure and at constant volume it follows that:

$$\begin{aligned} C_p = \partial_{T,p}H &\approx \partial_{T,V}U = C_V \\ &\Rightarrow \\ p &\cong (C_p - C_V) \approx 0. \end{aligned}$$

Applying the constraints listed on page 1 it follows that:

$$\partial_t(\rho C_p T - p) + \nabla\cdot((\rho C_p T - p)\mathbf{v}) = \nabla\cdot(\alpha\rho C_p \nabla T) + \Phi + C_p S_T \tag{A.14}$$

$$\Rightarrow$$

$$\rho r^2(\partial_t T + \nabla\cdot(T\mathbf{v})) = \rho\alpha r^2\nabla^2 T + r^2 S_T \tag{A.15}$$

$$\Rightarrow$$

$$r\rho\partial_t T + \rho(rv_r\partial_r + rv_z\partial_z + r\partial_r v_r + r\partial_z v_z + v_r)T = \alpha\rho(r\partial_r^2 + \partial_r + r\partial_z^2)T + rS_T \tag{A.16}$$

$$\Rightarrow \qquad \text{(use result of A.6)}$$

$$(r\rho(\partial_t + v_r\partial_r + v_z\partial_z) + rS_\rho)T = \alpha\rho(r\partial_r^2 + \partial_r + r\partial_z^2)T + rS_T. \tag{A.17}$$

B Discretization of the general convection-diffusion equation

Start with the abstract general convection-diffusion equation Eq. (1.9):

$$\partial_t(\rho\Phi) + \nabla\cdot(\rho\mathbf{v}\Phi - \Gamma\nabla\Phi) = S, \tag{B.1}$$

defining the flux $\mathbf{J}$ due to convection and diffusion:

$$\mathbf{J} \equiv \rho\mathbf{v}\Phi - \Gamma\nabla\Phi, \tag{B.2}$$

Integration over the control volume V_P yields:

$$\frac{\rho_P^t\Phi_P^t - \rho_P^{t_o}\Phi_P^{t_o}}{t - t_o}V_P + \sum_{\xi\in\{f,b,u,d\}}\int_{A_\xi}\mathbf{J}_\xi\cdot d\mathbf{A} = (S_C + S_P\Phi_P^t)V_P. \tag{B.3}$$

The indices ξ refer to the surfaces A_ξ where the fluxes are evaluated. The flux *out* of control volume V_P has a positive sign given by the definition of the direction vector of $d\mathbf{A}$ which is pointed outward of the control volume. The time derivative is evaluated over the time interval $\Delta t \equiv t - t_o$, the difference of the current time t and the previous time t_o. In general, the source S can be dependent on the dependent variable Φ. Therefore, this term is linearized where S_C denotes the constant part for all control volumes and $S_P\Phi_P$ denotes the control volume specific linear addition. Because the discretized equations will be solved by linear techniques there is no use for introduction of higher order dependencies. First, Eq. (B.3) is rewritten: for $\Gamma = 0$ and $\Phi = 1$ (Table 1.1) the continuity equation (1.1) reads:

$$\frac{\rho_P^t - \rho_P^{t_o}}{\Delta t}V_P + \sum_{\xi\in\{f,b,u,d\}}\int_{A_\xi}\rho_P^t\mathbf{v}_\xi\cdot d\mathbf{A} = (S_{\rho C} + S_{\rho P})V_P. \tag{B.4}$$

Multiplication of this equation by a factor Φ_P^t – which now represents either $1, v_r, v_z$, or T – and subsequently subtract it from Eq. (B.3) yields the integrated and discretized equivalent of Eq. (1.10):

$$\frac{(\Phi_P^t - \Phi_P^{t_o})\rho_P^{t_o}}{\Delta t}V_P + \sum_{\xi\in\{f,b,u,d\}}(J_\xi - F_\xi\Phi_P^t) = ((S_C + S_P\Phi_P^t) - (S_{\rho C} + S_{\rho P})\Phi_P^t)V_P, \tag{B.5}$$

with:

$$\begin{cases} J_\xi \equiv \int_{A_\xi} \mathbf{J}_\xi \cdot d\mathbf{A}, \\ \\ F_\xi \equiv \int_{A_\xi} \rho_P^t \mathbf{v}_\xi \cdot d\mathbf{A}. \end{cases}$$

Note that both $\mathbf{J}$ and $\rho\mathbf{v}$ are assumed uniform over control surface A_ξ.

Furthermore, the terms $J_\xi - F_\xi \Phi_P^t$ can be expressed as follows:

first rewrite the flux:

$$J_\xi^* = \frac{\mathbf{J}_\xi \cdot \boldsymbol{\delta}_\xi}{\Gamma} = \overbrace{\frac{\rho \mathbf{v}_\xi \cdot \boldsymbol{\delta}_\xi}{\Gamma}}^{\mathrm{Pe}} \Phi_\xi - \frac{1}{\mathbf{J}_\xi \cdot \boldsymbol{\delta}_\xi} \nabla \Phi_\xi, \tag{B.6}$$

where $\boldsymbol{\delta}_\xi$ is the displacement vector between two successive centregridpoints on both sides of surface A_ξ see, *e.g.,* Figure D.1.
The value Φ_ξ at this surface will be an averaged value of both Φ_B (*B*efore) and Φ_A (*A*head) of the surface while the gradient $\nabla \Phi_\xi$ at the surface essentially scales with the difference of these gridpoint Φ's, thus:

$$\begin{aligned} &J^* = \mathrm{Pe}((1-\alpha)\Phi_A + \alpha \Phi_B) - \beta(\Phi_A - \Phi_B), && \Rightarrow \\ &J^* = B(\mathrm{Pe})\Phi_B - A(\mathrm{Pe})\Phi_A. && \end{aligned} \tag{B.7}$$

Here A and B are dimensionless coefficients, both functions of the Peclet number Pe. Properties of A and B are:

$$B(\mathrm{Pe}) = A(\mathrm{Pe}) + \mathrm{Pe}, \tag{B.8}$$

furthermore, by symmetry, on reversing coordinate axes:

$$B(\mathrm{Pe}) = A(-\mathrm{Pe}). \tag{B.9}$$

These properties are fulfilled if:

$$\begin{cases} A(\mathrm{Pe}) & = \dfrac{\mathrm{Pe}}{\exp(\mathrm{Pe}) - 1}, \\ B(\mathrm{Pe}) & = A(\mathrm{Pe}) \exp(\mathrm{Pe}). \end{cases} \tag{B.10}$$

By virtue of the symmetry relations above if only $A(\mathrm{Pe})$ for $\mathrm{Pe} > 0$ is known both $A(\mathrm{Pe})$ and $B(\mathrm{Pe})$ are *fully* determined determined, because:

for $\mathrm{Pe} \geq 0$:

$$A(\mathrm{Pe}) = A(|\,\mathrm{Pe}\,|), \tag{B.11}$$

and for $\mathrm{Pe} < 0$:

$$\begin{aligned} A(\mathrm{Pe}) &= B(\mathrm{Pe}) - \mathrm{Pe} \\ &= A(-\mathrm{Pe}) - \mathrm{Pe} \\ &= A(|\,\mathrm{Pe}\,|) - \mathrm{Pe}\,. \end{aligned} \tag{B.12}$$

Thus, for $\mathrm{Pe} \in \mathbb{R}$:

$$\begin{cases} A(\mathrm{Pe}) & = A(|\,\mathrm{Pe}\,|) + \max(-\mathrm{Pe}, 0), \\ B(\mathrm{Pe}) & = A(|\,\mathrm{Pe}\,|) + \max(\mathrm{Pe}, 0). \end{cases} \tag{B.13}$$

Commonly used functions approximating $A(|\,\mathrm{Pe}_i\,|)$ for numerical interpolation are listed in Table B.1.

Table B.1: The function $A(|\,\mathrm{Pe}_i\,|)$ for different schemes.

Scheme	$A(\lvert\,\mathrm{Pe}_i\,\rvert)$
Central difference	$1 - \dfrac{\lvert\,\mathrm{Pe}_i\,\rvert}{2}$
Upwind	1
Hybrid	$\max\left(0, 1 - \dfrac{\lvert\,\mathrm{Pe}_i\,\rvert}{2}\right)$
Power law	$\max\left(0, 1 - \dfrac{\lvert\,\mathrm{Pe}_i\,\rvert}{10}\right)^n$
Exponential (exact)	$\dfrac{\lvert\,\mathrm{Pe}_i\,\rvert}{\exp\lvert\mathrm{Pe}_i\rvert - 1}$

For the flux J^* it now follows:

using:

$$J^* - \mathrm{Pe}\,\Phi_B = B\Phi_B - A\Phi_A - \mathrm{Pe}\,\Phi_B \tag{B.14}$$

$$= B\Phi_B - A\Phi_A - (B-A)\Phi_B \tag{B.15}$$

that:

$$\begin{cases} J^* - \mathrm{Pe}\,\Phi_B &= A(\mathrm{Pe})(\Phi_B - \Phi_A), \\ J^* - \mathrm{Pe}\,\Phi_A &= B(\mathrm{Pe})(\Phi_B - \Phi_A), \end{cases} \Rightarrow \begin{cases} J - F_\xi\Phi_B &= a_A(\Phi_B - \Phi_A), \\ J - F_\xi\Phi_A &= a_B(\Phi_B - \Phi_A). \end{cases} \tag{B.16}$$

With this result, and using a cylindrically symmetric grid defined by $\hat{\mathbf{r}}$ and $\hat{\mathbf{z}}$, Eq. (B.5) is written as:

$$\sum_{\omega\in\{P,U,D,B,F\}} a_\omega\Phi_\omega = \mathbf{a}_\Phi\cdot\mathbf{\Phi} = b, \tag{B.17}$$

where:

$$a_{\omega'} = -(D_\xi A(|\,\mathrm{Pe}_\xi\,|) + \max(-F_\xi, 0)) \tag{B.18}$$

for combinations of:

$$(\omega', \xi) \in \{(B,b),(U,u),(F,f),(D,d)\}.$$

Furthermore,

$$a_P = \sum_{\omega'\in\{U,D,B,F\}} a_{\omega'} + a_P^0 - (S_P - S_{\rho P} - S_{\rho C})V_P, \tag{B.19}$$

$$b = S_C V_P + a_P^0\Phi_P^0, \tag{B.20}$$

$$a_P^0 = \frac{\rho_P^0 V_P}{\Delta t}, \tag{B.21}$$

$$F_\xi = \rho_P(\mathbf{v}_\xi\cdot\hat{\mathbf{n}}_{A_\xi})A_\xi, \tag{B.22}$$

$$D_\xi = \frac{F_\xi}{\mathrm{Pe}_\xi} = \frac{\Gamma_\xi}{|\boldsymbol{\delta}_\xi|}A_\xi. \tag{B.23}$$

C GNU OCTAVE/MATLAB-code

C.1 Coding of a_{fb}, a_{ff}, a_{fu}, and a_{fd} for v_r computation on the r-staggered grid.

Computation of D values

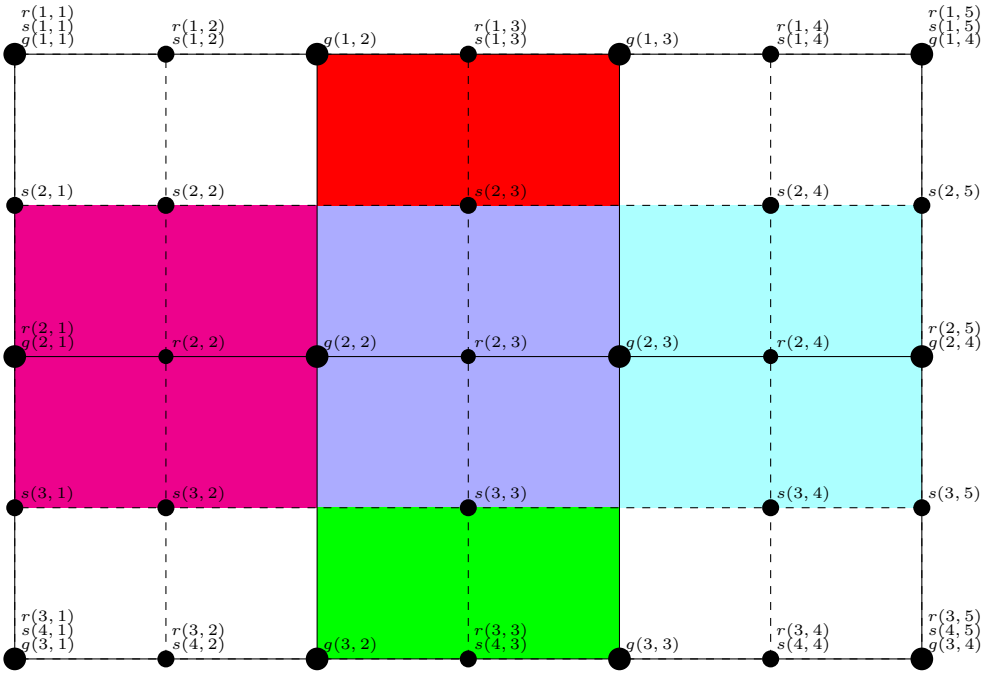


Figure C.1: Gridpoint numbering with respect to $M \times N = 3 \times 4$ centered grid g, with the associated $(M+1) \times (N+1) = 4 \times 5$ double staggered grid s, and the associated $M \times (N+1) = 3 \times 5$ r-staggered grid r.

In Figure C.1 a small grid is sketched with centre gridpoints $g(i_g, j_g)$ for i_g running from 1 through 3 and j_g running from 1 through 4 with i_g the rowcounter and j_g the columncounter. Also the double staggered gridpoints $s(i_s, j_s)$ are indicated for $(i_s, j_s)(\in 1 \ldots 4, 1 \ldots 5)$ and the r-staggered gridpoints $r(i_r, j_r)$ for $(i_r, j_r) \in (1 \ldots 3, 1 \ldots 5)$.
If computing the radial velocity components v_r on, *e.g.*, the r-staggered gridpoint $f = r(2,3)$ a staggered controlvolume around this point is considered which is drawn in blue. The computed value of v_r depends on the quantities a_f around gridpoint f, a_{ff} around gridpoint $ff = r(2,4)$ (cyan), a_{fb} around gridpoint $fb = r(2,2)$ (magenta), a_{fu} around gridpoint $fu = r(1,3)$ (red), and a_{fd} around gridpoint $fd = r(3,3)$ (green) as indicated in Eq. (3.1). It turns out that the F and D values, and with that the Peclet numbers appearing in Eq. (3.1), can be computed in one run, which is outlined in Eqs. (C.1) – (C.4)

for points $P = g(2,2)$, $F = g(2,3)$, $ffu = s(2,3)$, and $ffd = s(3,3)$ using the notation as in Eq. (3.3).

$$\left.\begin{aligned} D_P \quad &: D(g(2,2)) = \frac{\mu(g(2,2))A_{\rightarrow}(r(2,2))}{R(r(2,3)) - R(r(2,2))} \\ D_F \quad &: D(g(2,3)) = \frac{\mu(g(2,3))A_{\rightarrow}(r(2,3))}{R(r(2,4)) - R(r(2,3))} \end{aligned}\right\} \quad D_g(i,j) = \frac{\mu_g(i,j)A_{r\rightarrow}(i,j)}{\underbrace{R_r(i,j+1) - R_r(i,j)}_{\delta R_r(i,j)}} \tag{C.1}$$

$$\left.\begin{aligned} D_{ffu} \quad &: D(s(2,3)) = \frac{\mu(s(2,3))A_{\downarrow}(r(1,3))}{H(r(2,3)) - H(r(1,3))} \\ D_{ffd} \quad &: D(s(3,3)) = \frac{\mu(s(3,3))A_{\downarrow}(r(2,3))}{H(r(3,3)) - H(r(2,3))} \end{aligned}\right\} \quad D_s(i+1,j) = \frac{\mu_s(i+1,j)A_{r\downarrow}(i,j)}{\underbrace{H_r(i+1,j) - H_r(i,j)}_{\delta H_r(i,j)}}$$

The subscript s denotes a value of D or μ at a doublestaggered grid point. The subscript g denotes that one of these variables is taken at the centre grid. All other variables are taken at the r-staggered grid with $A_{\rightarrow}$ the forward surface, $A_{\downarrow}$ the downward surface, R the radial coordinate, and H the axial coordinate. The expressions in Eq. (C.1) are cast in OCTAVE/MATLAB code.

quantity		original size		reduced to $M \times N$ centered grid size
D_g	$\rightarrow$	$M \times N$	$\rightarrow$	`Drg`
μ_g	$\rightarrow$	$M \times N$	$\rightarrow$	`Mug` $=$ `GpointProps(:,:,MuTlayer)`
$A_{r\rightarrow}$	$\rightarrow$	$M \times (N+1)$	$\rightarrow$	`AFr` $=$ `RpointProps(:,1:end-1,FWlayer)`
δR_r	$\rightarrow$	$M \times (N+1)$	$\rightarrow$	`dRr` $=$ `real(SpointProps(:,2:end,1) - SpointProps(:,1:end-1,1))`
				induced to $(M+1) \times (N+1)$ double staggered grid size
D_s	$\rightarrow$	$(M+1) \times (N+1)$	$\rightarrow$	`Drs`
μ_s	$\rightarrow$	$(M+1) \times (N+1)$	$\rightarrow$	`Mus` $=$ `SpointProps(:,:,MuTlayer)`
$A_{r\downarrow}$	$\rightarrow$	$M \times (N+1)$	$\rightarrow$	`ADr` $=$ `[zeros(1,size(RpointProps(:,:,DWlayer))(2));` $\cdots$ `RpointProps(:,:,DWlayer)]`
δH_r	$\rightarrow$	$M \times (N+1)$	$\rightarrow$	`dHr` $=$ `[GRpointH;zeros(1,size(GRpointH)(2))]` $-\cdots-$ `[zeros(1,size(GRpointH)(2));GRpointH]`

(C.2)

Such that, *e.g.,* the product $\mu_s .* A_{r\downarrow}$ in the numerator of D_s yields:

$$\begin{pmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} & \mu_{15} \\ \mu_{21} & \mu_{22} & \mu_{23} & \mu_{24} & \mu_{25} \\ \mu_{31} & \mu_{32} & \mu_{33} & \mu_{34} & \mu_{35} \\ \mu_{41} & \mu_{42} & \mu_{43} & \mu_{44} & \mu_{45} \end{pmatrix} .* \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ A_{r\downarrow 11} & A_{r\downarrow 12} & A_{r\downarrow 13} & A_{r\downarrow 14} & A_{r\downarrow 15} \\ A_{r\downarrow 21} & A_{r\downarrow 22} & A_{r\downarrow 23} & A_{r\downarrow 24} & A_{r\downarrow 25} \\ A_{r\downarrow 31} & A_{r\downarrow 32} & A_{r\downarrow 33} & A_{r\downarrow 34} & A_{r\downarrow 35} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \mu_{21}A_{r\downarrow 11} & \mu_{22}A_{r\downarrow 12} & \mu_{23}A_{r\downarrow 13} & \mu_{24}A_{r\downarrow 14} & \mu_{25}A_{r\downarrow 15} \\ \mu_{31}A_{r\downarrow 21} & \mu_{32}A_{r\downarrow 22} & \mu_{33}A_{r\downarrow 23} & \mu_{34}A_{r\downarrow 24} & \mu_{35}A_{r\downarrow 25} \\ \mu_{41}A_{r\downarrow 31} & \mu_{42}A_{r\downarrow 32} & \mu_{43}A_{r\downarrow 33} & \mu_{44}A_{r\downarrow 34} & \mu_{45}A_{r\downarrow 35} \end{pmatrix} \tag{C.3}$$

thereby fulfilling the relationship of Eq. (C.1) on the double staggered grid.

The resulting code in the procedure `SolveMomentum.m` in the OCTAVE/MATLAB code yields:

$$\begin{cases} \texttt{Drg} & = \texttt{Mug.} * \texttt{AFr./dRr;} \\ \\ \texttt{Drs} & = \texttt{Mus.} * \texttt{ADr./dHr;} \end{cases} \tag{C.4}$$

with `Drg` and `Drs` of dimensions $M \times N$ and $(M+1) \times (N+1)$, respectively, containing the D values for every centregrid and associated staggered grid point for radial velocity computation on the r-staggered grid.

Computation of F values

The computation of F values essentially follows the same procedure and constructions as that of the computation of D values. In particular, for values of F_g, on the centred grid, the expression in Eq. (C.1) for $D_g(i,j)$ is valid with $\mu_g(i,j)$ replaced with $\rho(i,j) = \rho$ and multiplication the linear velocity interpolation factor $\boldsymbol{\delta}_{r,f,ff} \cdot \mathbf{v}_{r,f,ff}$ used to compute v_{rg} on the centregrid point in between r-staggered grid points f and ff. Here:

$$v_{rg}\delta R_r(i,j) = \boldsymbol{\delta}_{r,f,ff} \cdot \mathbf{v}_{r,f,ff} = \begin{pmatrix} R_r(i,j+1) - R_g(i,j) \\ R_g(i,j) - R_r(i,j) \end{pmatrix} \cdot \begin{pmatrix} v_r^0(i,j) \\ v_r^0(i,j+1) \end{pmatrix} \tag{C.5}$$

The F_s values on the staggered gridpoints vanish by virtue of orthogonality between velocity vector and surface normal. Also, for all elements with vanishing velocity, *i.e.,* steel, isolation, and earth, the F values

vanish again due to a vanishing innerproduct $\boldsymbol{\delta}\cdot\mathbf{v}$. Therefore, the values for F_r and F_s yield:

$$\begin{cases} \texttt{Frg} & = \texttt{deltaRinVr}.*\texttt{Drg}.*\texttt{rho}./\texttt{Mug}; \\ \\ \texttt{Frs} & = 0; \end{cases} \tag{C.6}$$

The scalar `deltaRinVr` representing the innerproduct $\boldsymbol{\delta}\cdot\mathbf{v}$ is represented in OCTAVE-code:

$$\begin{aligned} \boldsymbol{\delta}_{r,f,ff}\cdot\mathbf{v}_{r,f,ff} &= \big(R_r(i,j+1)-R_g(i,j)\big)v_r(i,j)+\big(R_g(i,j)-R_r(i,j)\big)v_r(i,j+1) \\ &\Rightarrow \\ \texttt{deltaRinVr} &= \texttt{real}(\texttt{RpointProps}(:,2:\texttt{end},1)-\texttt{GpointProps}(:,:,1)).*. \end{aligned} \tag{C.7}$$

C.2 Coding of a_{du}, a_{dd}, a_{db}, and a_{df} for v_h computation on the h-staggered grid.

Computation of D values

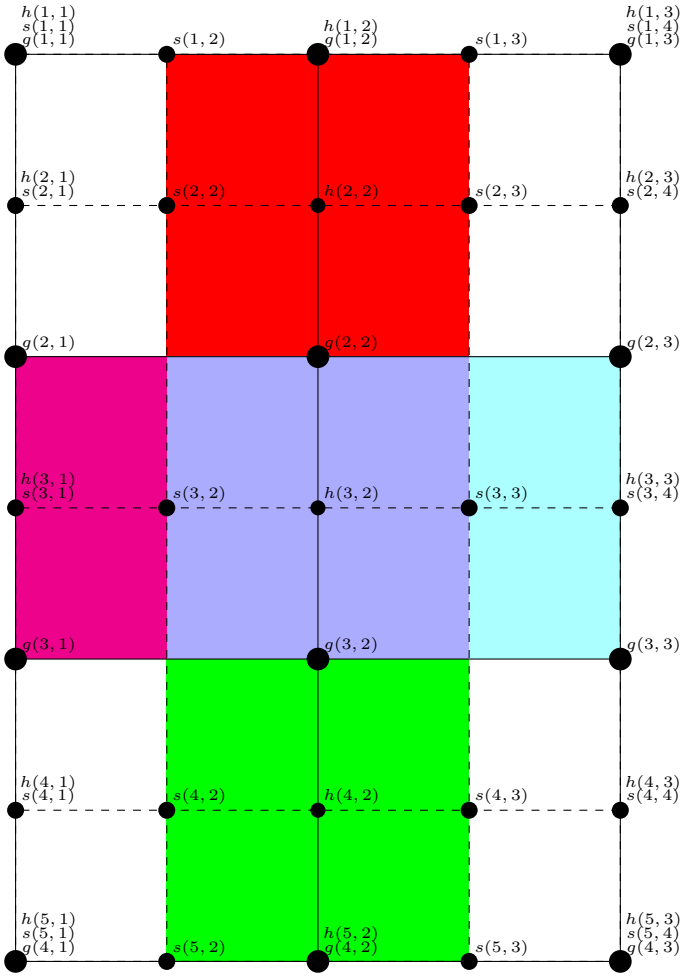


Figure C.2: Gridpoint numbering with respect to $M \times N = 4 \times 3$ centered grid g, with the associated $(M+1) \times (N+1) = 5 \times 4$ double staggered grid s, and the associated $(M+1) \times N = 5 \times 3$ h-staggered grid h.

In Figure C.2 a small grid is sketched with centre gridpoints $g(i_g, j_g)$ for i_g running from 1 through 4 and j_g running from 1 through 3 with i_g the rowcounter and j_g the columncounter. Also the double staggered gridpoints $s(i_s, j_s)$ are indicated for $(i_s, j_s)(\in 1 \ldots 5, 1 \ldots 4)$ and the h-staggered gridpoints $r(i_r, j_r)$ for $(i_r, j_r) \in (1 \ldots 5, 1 \ldots 3)$.
If computing the axial velocity components v_h on, *e.g.*, the h-staggered gridpoint $d = h(3, 2)$ a staggered controlvolume around this point is considered which is drawn in blue.The computed value of v_h depends on the quantities a_d, a_{dd} around gridpoint $dd = h(4, 2)$ (green), a_{du} around gridpoint $du = h(2, 2)$ (red), a_{db} around gridpoint $db = h(3, 1)$ (magenta), and a_{df} around gridpoint $df = h(3, 3)$ (cyan) analagous to

a_{ff}, a_{fb}, a_{fu}, and a_{fd} in Eq. (3.1). It turns out that the values F_ζ, D_ζ, and Pe_ζ for $\zeta \in \{P, D, ddf, ddb\}$ (see Fig. 3.1) can be computed in one run, which is outlined in Eqs. (C.8) – (C.11) for points $P = g(2,2)$, $D = g(3,2)$, $ddf = s(3,3)$, and $ddb = s(3,2)$ using the notation analogous to Eq. (3.3).

$$\left.\begin{aligned} D_P &: D(g(2,2)) = \frac{\mu(g(2,2))A_\downarrow(h(2,2))}{H(h(3,2)) - H(h(2,2))} \\ D_D &: D(g(3,2)) = \frac{\mu(g(3,2))A_\downarrow(h(3,2))}{H(h(4,2)) - H(h(3,2))} \end{aligned}\right\} \quad D_g(i,j) = \frac{\mu_g(i,j)A_{h\downarrow}(i,j)}{\underbrace{H_h(i+1,j) - H_h(i,j)}_{\delta H_h(i,j)}} \tag{C.8}$$

$$\left.\begin{aligned} D_{ddb} &: D(s(3,2)) = \frac{\mu(s(3,2))A_\rightarrow(h(3,1))}{R(h(3,2)) - R(h(3,1))} \\ D_{ddf} &: D(s(3,3)) = \frac{\mu(s(3,3))A_\rightarrow(h(3,2))}{R(h(3,3)) - R(h(3,2))} \end{aligned}\right\} \quad D_s(i,j+1) = \frac{\mu_s(i,j+1)A_{h\rightarrow}(i,j)}{\underbrace{R_h(i,j+1) - R_h(i,j)}_{\delta R_h(i,j)}}$$

The subscript s denotes a value of D or μ at a doublestaggered grid point. The subscript g denotes that one of these variables is taken at the centre grid. All other variables are taken at the h-staggered grid with $A_\rightarrow$ the forward surface, $A_\downarrow$ the downward surface, R the radial coordinate, and H the axial coordinate. The expressions in Eq. (C.8) are cast in OCTAVE/MATLAB code.

quantity		original size		reduced to $M \times N$ centered grid size
D_g	$\rightarrow$	$M \times N$	$\rightarrow$	`Dhg`
μ_g	$\rightarrow$	$M \times N$	$\rightarrow$	`Mug` $=$ `GpointProps(:,:,MuTlayer)`
$A_{h\downarrow}$	$\rightarrow$	$(M+1) \times N$	$\rightarrow$	`ADh` $=$ `HpointProps(1:end-1,:,DWlayer)`
δH_h	$\rightarrow$	$(M+1) \times N$	$\rightarrow$	`dHh` $=$ `SHpointH(2:end,:)` $-$ `SHpointH(1:end-1,:)`
				induced to $(M+1) \times (N+1)$ double staggered grid size
D_s	$\rightarrow$	$(M+1) \times (N+1)$	$\rightarrow$	`Dhs`
μ_s	$\rightarrow$	$(M+1) \times (N+1)$	$\rightarrow$	`Mus` $=$ `SpointProps(:,:,MuTlayer)`
$A_{h\rightarrow}$	$\rightarrow$	$(M+1) \times N$	$\rightarrow$	`AFh` $=$ `[zeros(size(HpointProps(:,:,FWlayer)(2))),` $\cdots$ `HpointProps(:,:,FWlayer)]`
δR_h	$\rightarrow$	$(M+1) \times N$	$\rightarrow$	`dRh` $=$ `[SHpointR, zeros(size(SHpointR)(2))]` $-\cdots=$ `[zeros(size(SHpointR)(2)), SHpointR]`

(C.9)

Such that, *e.g.*, the product $\mu_s. * A_{h\rightarrow}$ in the numerator of D_s yields:

$$\begin{pmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} \\ \mu_{21} & \mu_{22} & \mu_{23} & \mu_{24} \\ \mu_{31} & \mu_{32} & \mu_{33} & \mu_{34} \\ \mu_{41} & \mu_{42} & \mu_{43} & \mu_{44} \\ \mu_{51} & \mu_{52} & \mu_{53} & \mu_{54} \end{pmatrix} . * \begin{pmatrix} 0 & A_{h\rightarrow 11} & A_{h\rightarrow 12} & A_{h\rightarrow 13} \\ 0 & A_{h\rightarrow 21} & A_{h\rightarrow 22} & A_{h\rightarrow 23} \\ 0 & A_{h\rightarrow 31} & A_{h\rightarrow 32} & A_{h\rightarrow 33} \\ 0 & A_{h\rightarrow 41} & A_{h\rightarrow 42} & A_{h\rightarrow 43} \\ 0 & A_{h\rightarrow 51} & A_{h\rightarrow 52} & A_{h\rightarrow 53} \end{pmatrix} = \begin{pmatrix} 0 & \mu_{12}A_{h\rightarrow 11} & \mu_{13}A_{h\rightarrow 12} & \mu_{14}A_{h\rightarrow 13} \\ 0 & \mu_{22}A_{h\rightarrow 21} & \mu_{23}A_{h\rightarrow 22} & \mu_{24}A_{h\rightarrow 23} \\ 0 & \mu_{32}A_{h\rightarrow 31} & \mu_{33}A_{h\rightarrow 32} & \mu_{34}A_{h\rightarrow 33} \\ 0 & \mu_{42}A_{h\rightarrow 41} & \mu_{43}A_{h\rightarrow 42} & \mu_{44}A_{h\rightarrow 43} \\ 0 & \mu_{52}A_{h\rightarrow 51} & \mu_{53}A_{h\rightarrow 52} & \mu_{54}A_{h\rightarrow 53} \end{pmatrix} \tag{C.10}$$

thereby fulfilling the relationship of Eq. (C.8) on the double staggered grid.

The resulting code in the procedure `SolveMomentum.m` in the OCTAVE/MATLAB code yields:

$$\begin{cases} \texttt{Dhg} & = \texttt{Mug.} * \texttt{ADh./dHh;} \\ \\ \texttt{Dhs} & = \texttt{Mus.} * \texttt{AFh./dRh;} \end{cases} \tag{C.11}$$

with `Dhg` and `Dhs` of dimensions $M \times N$ and $(M+1) \times (N+1)$, respectively, containing the D values for every centregrid and associated staggered grid point for axial velocity computation on the h-staggered grid.

D Figures

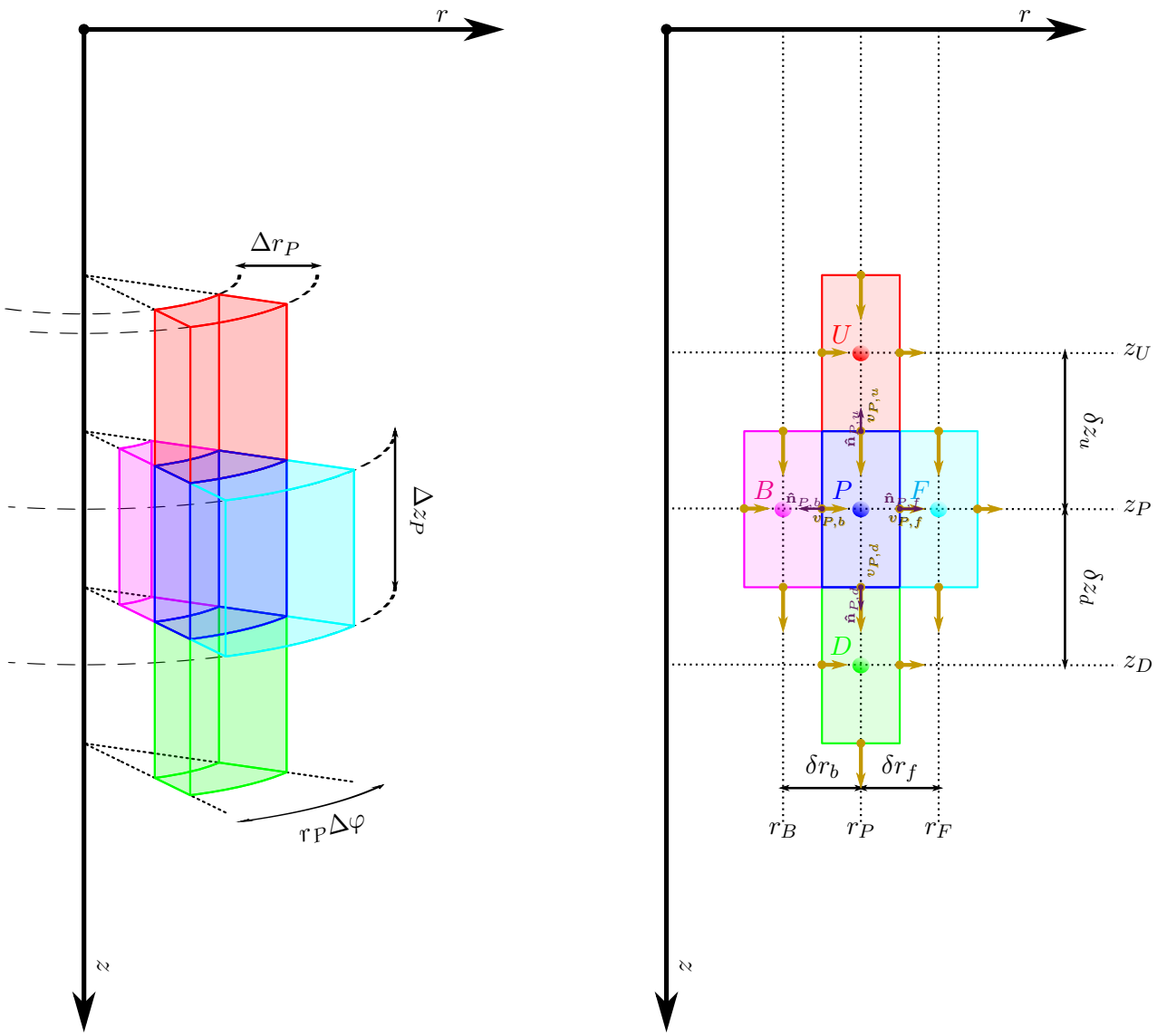
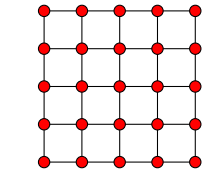


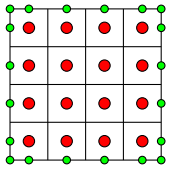
Figure D.1: Schematic view of grid- and elementary volume definition in cylindrical coordinates.



$$\begin{pmatrix} g_{1,1} & \cdots & g_{1,M} \\ \vdots & \ddots & \vdots \\ g_{N,1} & \cdots & g_{N,M} \end{pmatrix}$$

GpointProps

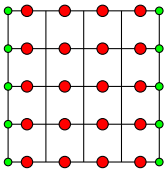
(A)



$$\begin{pmatrix} s_{1,1} & & \cdots & & s_{1,M+1} \\ & s_{2,2} & \cdots & s_{N,2} & \\ \vdots & & \ddots & & \vdots \\ & s_{N,2} & \cdots & s_{N,M} & \\ s_{N+1,1} & & \cdots & & s_{N+1,M+1} \end{pmatrix}$$

SpointProps

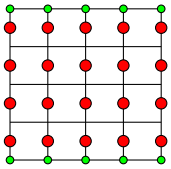
(B)



$$\begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,M} & r_{1,M+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{N,1} & r_{N,2} & \cdots & r_{N,M} & r_{N,M+1} \end{pmatrix}$$

RpointProps

(C)



$$\begin{pmatrix} h_{1,1} & \cdots & h_{1,M} \\ h_{2,1} & \cdots & h_{2,M} \\ \vdots & \ddots & \vdots \\ h_{N,1} & \cdots & h_{N,M} \\ h_{N+1,1} & \cdots & h_{N+1,M} \end{pmatrix}$$

HpointProps

(D)

Figure D.2: Definitions of the matrices `GpointProps`, `SpointProps`, `RpointProps`, and `HpointProps` as used in the octave-code `RunSimulation.m`.

Centregrid (A); double staggered grid (B); r-staggered grid (C); z-staggered grid (D).

Pure gridpoints of the respective grids are indicated with red dots; auxiliary gridpoints, not belonging to the respective grids but included for programming ease, are indicated with green dots.

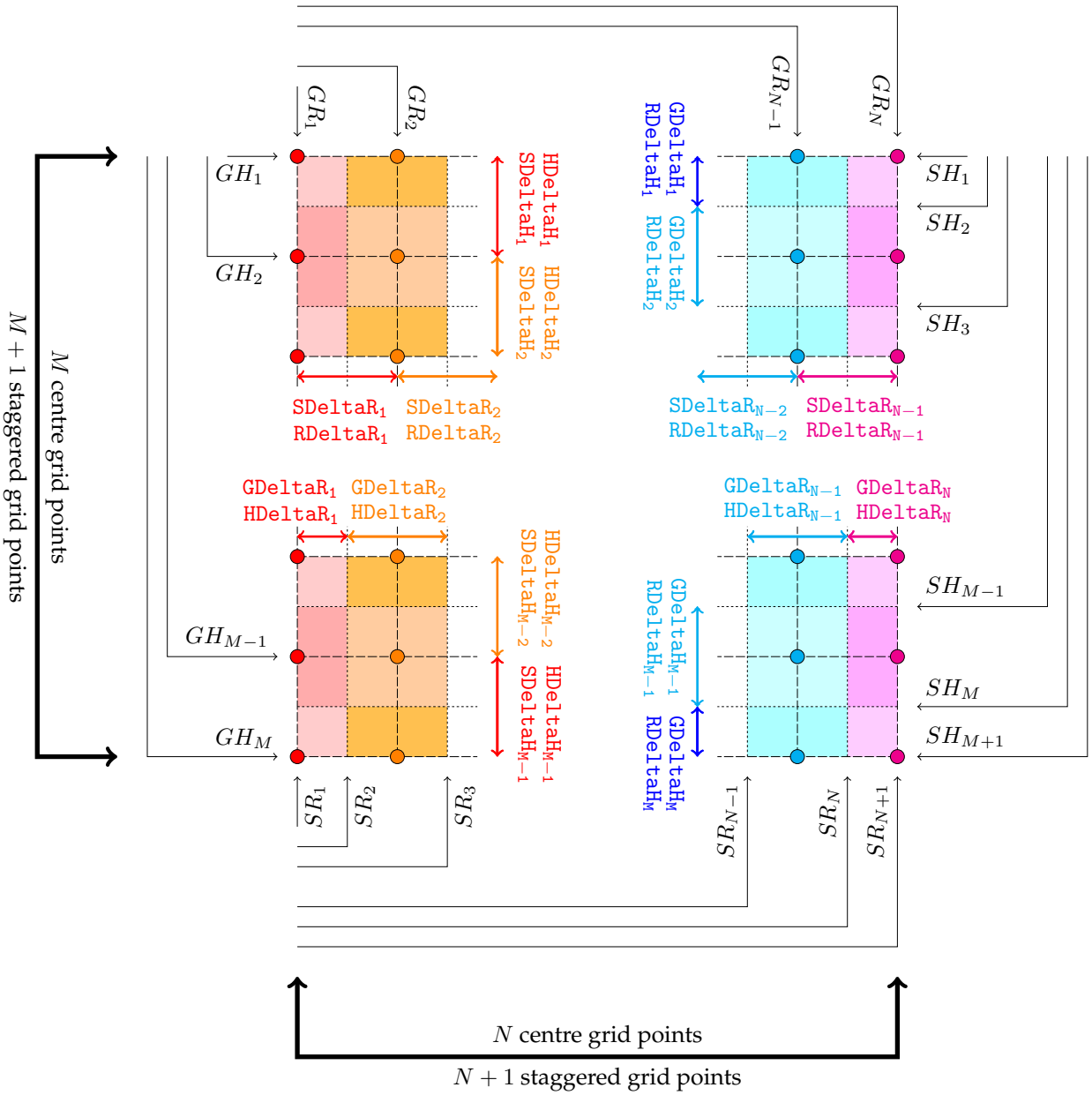


Figure D.3: Definitions of controlvolume surfaces and volumes of the centered grid in terms of the array elements of GR, GH, SR, and SH.

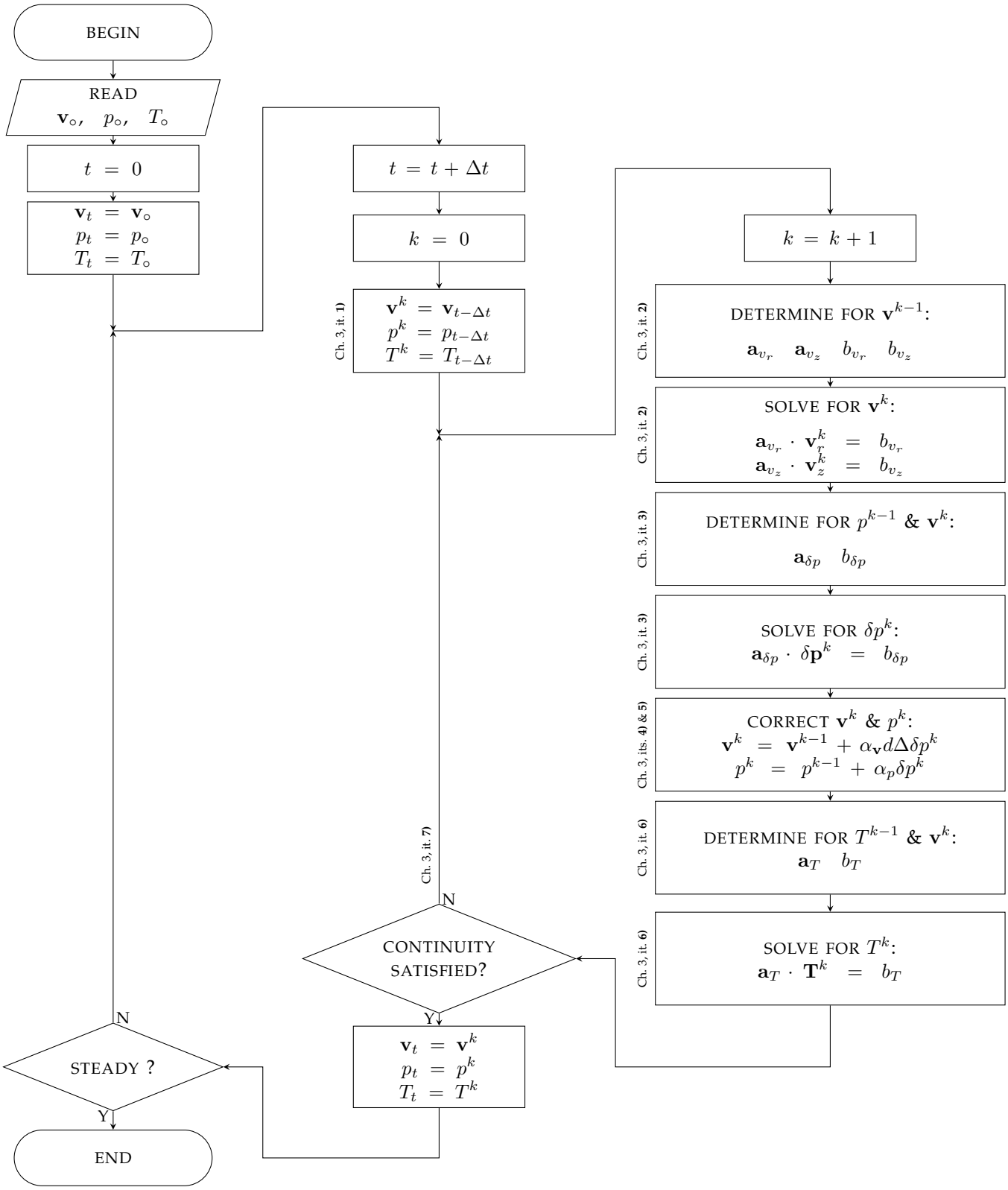


Figure D.4: Flowdiagram for the SIMPLE algorithm.

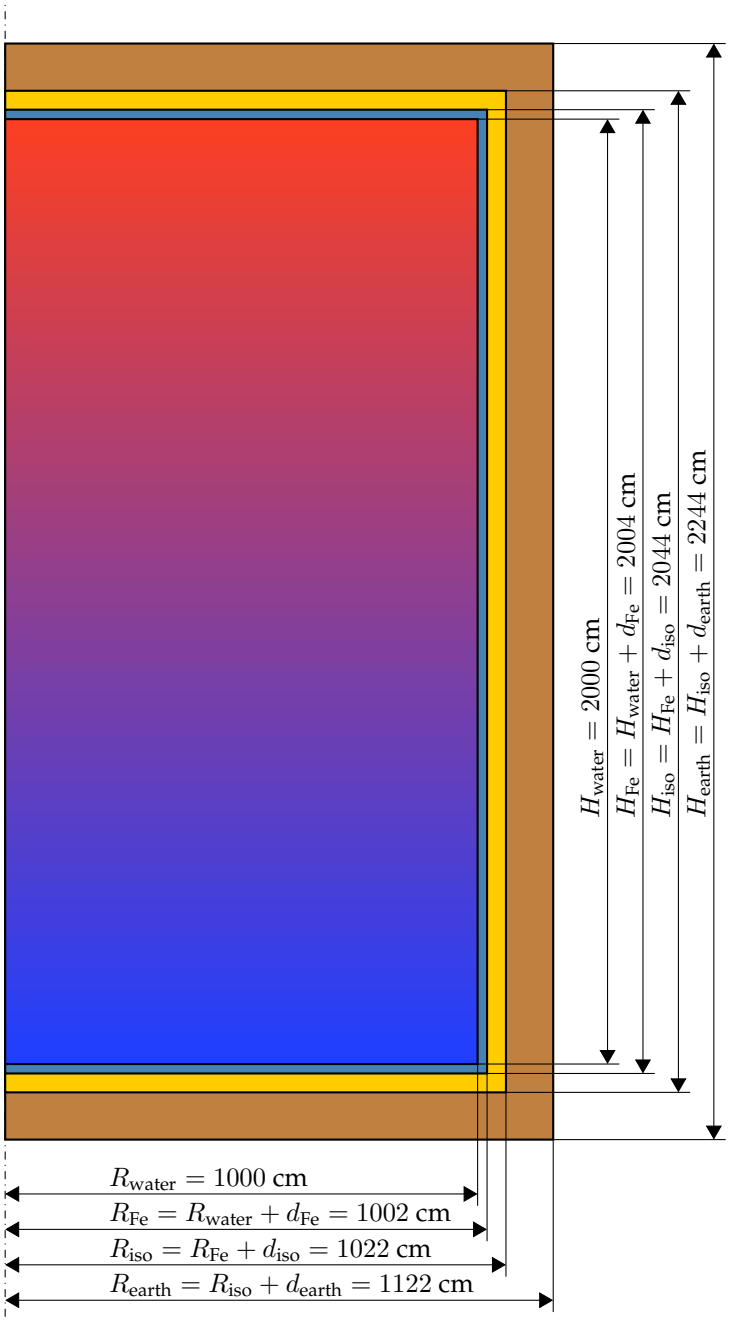


Figure D.5: Geometry of the storage tank under study.

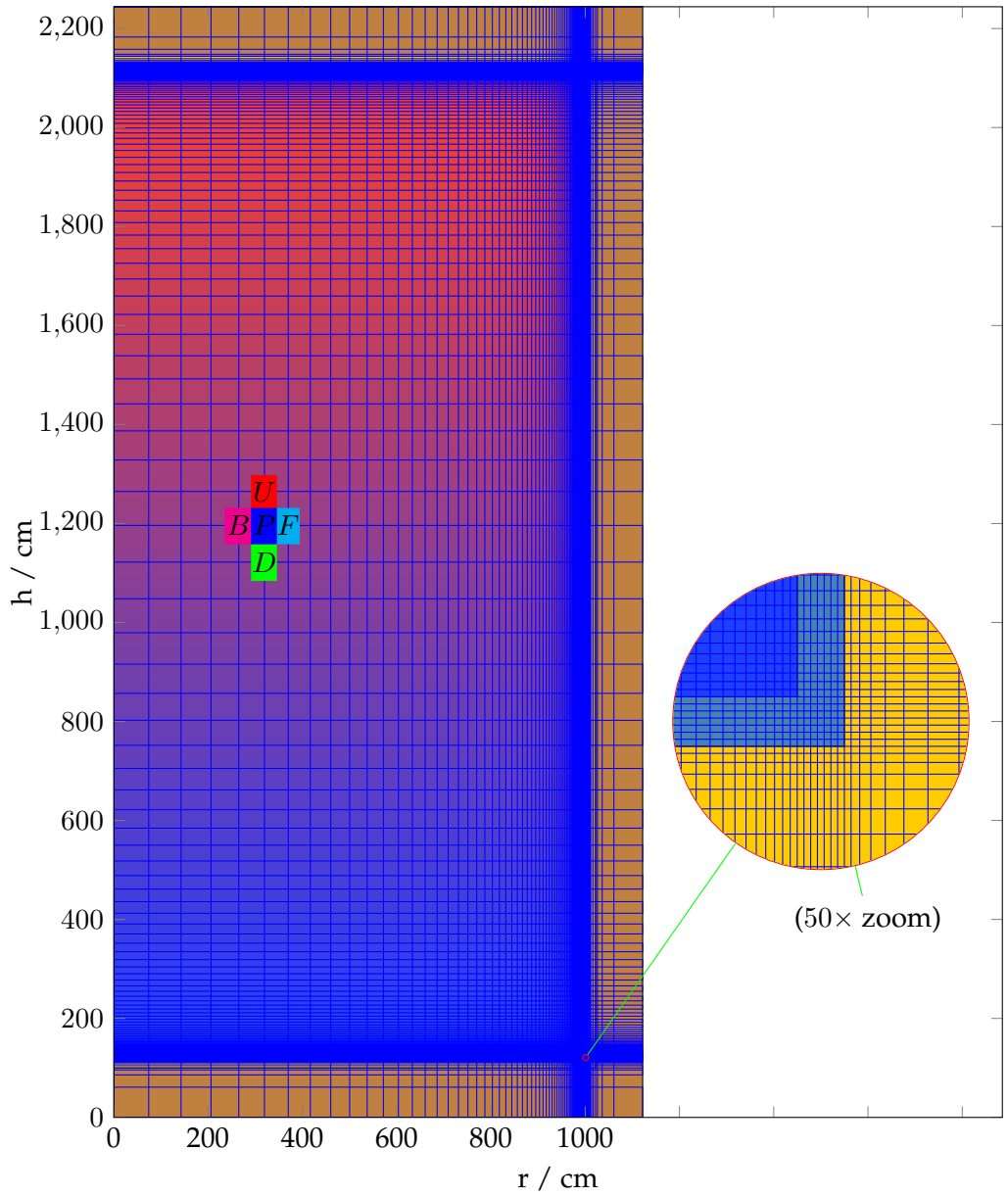


Figure D.6: Computational grid used with the SIMPLE-algorithm. An indication of a set of controlvolumes D, U, F, B, P for a gridpoint is given. Zoom: example of grid structure at water-metal-isolation transition.

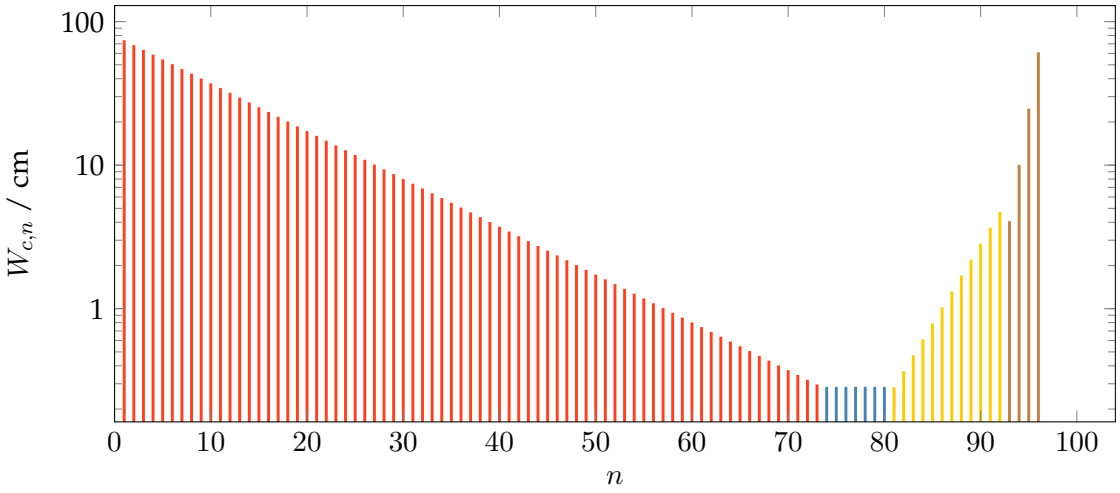


Figure D.7: Intervalwidths $W_{c,n}$ for the radial direction as used for the computational grid in Figure D.6 on a logarithmic scale as function of n: water (red), metal (blue), isolation (yellow), and earth (brown).