

**Documentation of the prognostic
mesoscale model
GRAMM (Graz Mesoscale Model)
Version 22.09**

Author

Mag. Dr. Dietmar Öttl
Ing. Markus Kuntner

Contributions by

CONTENT

1	Changes to previous versions	5
1.1	Changes to the previous version GRAMM 17.1	5
1.2	Changes to the previous version GRAMM 16.1	6
2	General information	7
3	Introduction	8
4	Detailed model description	9
4.1	Preamble	9
4.2	Basic equations	9
4.3	Approximations	11
4.4	Turbulence parameterizations	13
4.4.1	Algebraic turbulence model according to Pandolfo (1969)	14
4.4.2	Turbulent closure of second order	15
4.5	Boundary conditions	17
4.5.1	Lower boundary	17
4.5.2	Lateral boundaries	20
4.5.3	Upper boundary	21
4.6	Surface energy balance	21
4.6.1	Solar radiation	21
4.6.2	Longwave radiation	24
4.6.3	Soil heat flux	25
4.6.4	Snow cover	26
4.7	Numerical schemes	29
4.7.1	Discretization grid	29
4.7.2	Discretization of the conservation equations for momentum	30
4.7.3	Discretization of the conservation equations for pot. Temperature, humidity, turbulent kinetic energy, and dissipation rate	41
4.7.4	Discretization of the non-hydrostatic pressure equation	43
4.7.5	Algorithm to solve the set of conservation equations	49
4.8	Model initialization	49
4.8.1	Single point observation of wind speed and –direction plus stability class	50
4.8.2	Using observed profiles of wind and temperature	53
4.8.3	Initialization of the soil temperatures and sensible heat fluxes	58
5	Compliance with the German Guideline VDI 3783 - 7	59
5.1	General model evaluation	59
5.2	Scientific model evaluation	60
5.3	Model validation	60
5.3.1	Test case E1 (gravity waves)	62
5.3.2	Test case E2 (gravity waves, influence of wind speed)	65
5.3.3	Test case E3 (influence of grid size)	66
5.3.4	Test case E4 (shadowing effects and drainage flows)	70
5.3.5	Test case E5 (homogeneity)	72
5.3.6	Test case E6 (field data Sophienhoehe)	75
5.3.7	Test case E7 (field data Grazer basin)	80
5.3.8	Test case E8 (field data Stuttgart basin)	86
6	Additional validation cases	91

6.1	Test case boundary layer	91
7	References.....	95
8	Acknowledgements.....	97
9	Appendix A.....	98
9.1	Startup parameters	98
9.2	Topographic position and landform analysis.....	98
9.3	List of variables in the GRAMM code	99
9.4	Control files	108
9.4.1	Input files	108
9.4.2	Output files	122

1 Changes to previous versions

1.1 Changes to the previous version GRAMM 20.09

- The application has been transformed to .NET6
- Enable optional user defined settings for the model initialization (chapter 9.4.1.11)
- Improved synchronization of the parallelized solver
- Improved file access for multiple instances in the same project folder
- An error that occurred when the operating system reported 0 free processor cores, resulting in a division by 0, has been fixed

1.2 Changes to the previous version GRAMM 20.01

- Code clean up and performance improvements within the GRAMM radiation module
- Removed the experimental dependencies of Windows-GDAL (introduced in GRAMM 20.01). GRAMM works in LINUX again
- Removed the experimental Topography Index Algorithm (TPI)

1.3 Changes to the previous version GRAMM 19.01

- GRAMM 20.01 has been compiled for the target framework .NETCore 3.1.

1.4 Changes to the previous version GRAMM 17.1

- Stability classes have been previously computed using in a wrong way by using the incoming radiation without the reflected part at the surface
- Changed radiation initialization for stability class 4
- An additional startup parameter allows to set a working directory
- An error in the calculation of the soil heat flux has been fixed
- The sky-view factor has been erroneously used in the computation of the surface longwave radiation balance
- Due to the corrected surface energy balance, some landuse parameters (e.g. forests) have been changed
- In general, the heat conductivities of soils have been set to lower values according to new observations found in literature

Changes to previous versions - Changes to the previous version GRAMM 16.1

- The minimum value for the friction velocity is set to 0.15 m/s instead of 0.05 m/s, and its maximum value is set to 2 m/s instead of 3 m/s
- The influence of snow cover and cloudiness on the surface energy balance has been introduced
- The minimum vertical turbulent exchange coefficient has been changed from 0.5 to 0.05 m²/s²
- The initial potential temperature is computed by using an adiabatic temperature gradient for moist air (instead of dry air)

1.5 Changes to the previous version GRAMM 16.1

- While in the previous version the sun's angle has been set dependent on the stability class only (when using the file meteopgt.all as meteorological input data), in the new version the dependencies have been extended to wind speed and latitude. In this way the natural variability in meteorology is a bit better represented in the model. Moreover, the global radiation fits better to the classification scheme recommended for GRAL. In case of stability class 1 (very convective), the sun has been set 21 June 3pm, which is a good approximation only for the northern hemisphere, but in the worst case, led to downslope winds in the southern hemisphere. This bug could also be solved with the new methodology.
- A new option called "dynamic sun rise" has been implemented, which allows for transient computation of the global radiation in combination with storing intermediate flow-field files for use in the match-to-observation procedure of the graphical user interface.
- GRAMM provides computed stability classes, Obukhov lengths, and friction velocities. Computed stability classes can optionally be used in the match-to-observation procedure of the graphical user interface.
- Specific humidity is computed again. Letting it drop led to rather stable atmospheres effectively damping the generation of e.g. cold-air-drainage flows.
- The sun's position was wrongly computed in all previous version of GRAMM in southern hemisphere (the sun passed over south).
- A new improved time-step scheme leads to lower residuals in the mass conservation equation.

2 General information

Current development team:

Physics, Numerics, Programming, Quality Assurance: Dietmar Oettl (Government of Styria, Air Quality Department, Austria)

Programming, Optimization, Quality Assurance: Markus Kuntner (Office of the Government of Tyrol, Emissions-Security-Facilities Department, Austria)

Training and support: Christian Kurz, Graz University of Technology, Austria

Software requirements: 64 bit Windows operation system for the .Net6 framework, LINUX or macOS system for the .Net6 framework.

GRAMM is delivered together with the Lagrangian Particle Model GRAL for pollutant dispersion and a graphical user interface that can be run on Windows or LINUX operating systems.

Programming language: C# (.NET6)

Availability: Free download from <http://lampx.tugraz.at/~gral/>

Source Code: <https://github.com/GralDispersionModel/GRAMM>

Typical domain sizes: 10 km – 250 km

Typical horizontal grid sizes: 50 m – 1 km

Typical vertical grid sizes: lowest cell height 5 m – 20 m; increasing cell heights with elevation by a factor of 1,05 – 1,40

Typical fields of applications:

- Simulation of quasi-steady-state flow fields in complex terrain for categorized meteorological situations (e.g. channelling effects, local catabatic flows, recirculations).
- Simulation of time-dependent flows in complex terrain for episodes of up to a few weeks (not well tested).

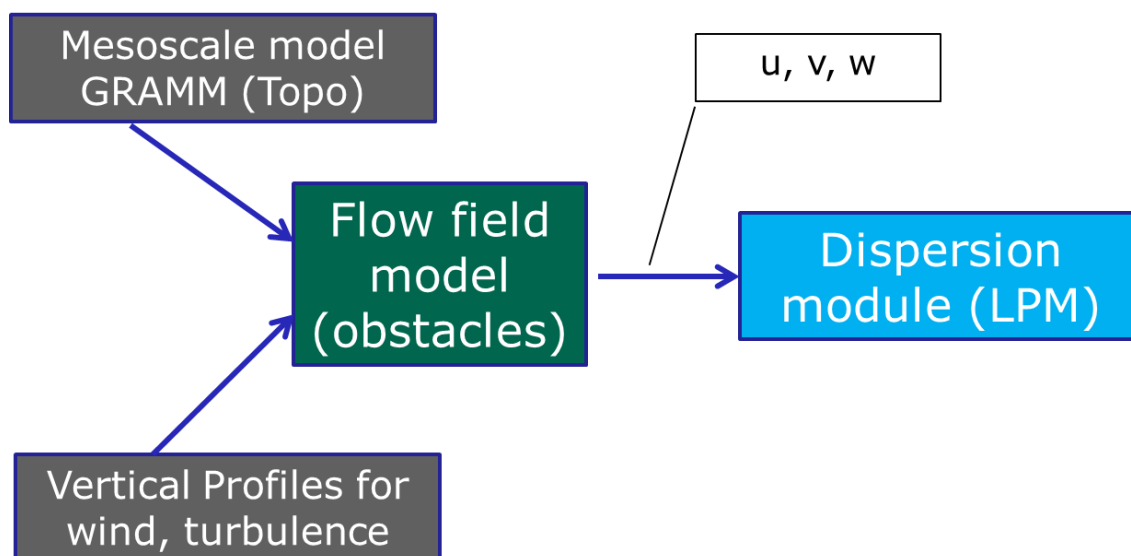
Application limits:

- GRAMM does not account for detailed cloud physics, precipitation.

3 Introduction

In the past, prognostic wind-field models were solely developed and operated by the scientific community. However nowadays they have become more and more applicable in a regulatory environment and are run by administrative staff as well as engineers. There is a demand for standardization, quality assurance, and training of personnel dealing with such complex numerical models. The GRAMM model has been developed at the Graz University of Technology, Inst. f. Internal Combustion Engines since the early 1990's mainly by Prof. Raimund Almbauer. Since 2006, it has been used at the Air Quality Department of the Government of Styria, and it is commonly used in Austria after the model had been provided online in 2012. Since about 2014 the model is being further developed by the Governments of Styria and Tyrol, Austria. Meanwhile, wind-field and air pollution studies have been carried out in various countries such as Switzerland (e.g. Berchet et al., 2015), using GRAMM mostly as wind-field model producing input to the Lagrangian Particle Model GRAL (Fig. 1).

Fig. 1. Coupling scheme showing the relationship between the prognostic mesoscale wind-field model GRAMM and the Lagrangian Particle Model GRAL.



The focus of most GRAMM applications is the computation of quasi steady-state flow fields in complex terrain for categorized meteorological situations using readily available meteorological data from routine networks. This report describes comprehensively the basic equations and approximations of GRAMM, the numerical schemes and validation studies performed so far. It should serve as the main reference for the GRAMM model. Information on how to operate the model for its standard applications is provided in the User Manual of the accompanying Graphical User Interphase, which is included in the freely available download package (<http://lampx.tugraz.at/~gral/>).

4 Detailed model description

4.1 Preamble

Throughout this document numerous equations will be encountered. In some cases tensor notation will be used where applicable and useful. For instance, the three Cartesian coordinates can be denoted by x_i . However, in many other cases equations are written using e.g. x , y , and z to denote the Cartesian coordinates in the horizontal and vertical directions.

4.2 Basic equations

Similar to most prognostic, mesoscale models, GRAMM is based on the following set of Reynolds-averaged conservation equations:

Conservation of momentum (Reynolds-averaged Navier-Stokes or RANS equations):

$$\frac{\partial(\bar{\rho} \bar{u}_i)}{\partial t} + \frac{\partial(\bar{\rho} \bar{u}_i \bar{u}_j)}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} - \bar{\rho} \varepsilon_{ijk} f_j \bar{u}_k - \delta_{i3} \bar{\rho} g - \bar{\rho} \frac{\partial(\overline{u'_i u'_j})}{\partial x_j} \quad (\text{E } 1)$$

Time averaging of variables is indicated by overbars, u_i denote the wind speed components in a Cartesian coordinate system, $\bar{\rho}$ is the density, x_i are the spatial coordinates, p is the pressure, g is gravitational acceleration, u'_i are the turbulent velocity components, and f_j are the Coriolis parameters:

$$f_j = \begin{pmatrix} 0 \\ 2\Omega \cos \varphi \\ 2\Omega \sin \varphi \end{pmatrix}, \quad (\text{E } 2)$$

with Ω being the Earth's rotational velocity, and φ the latitude.

It is necessary to explain the physical meaning of Reynolds-averaging in the frame of this type of model. In principle, any time-averaged quantity $\bar{\Psi}$ represents the non-turbulent part, whereas Ψ' represents the turbulent part. The latter is not resolved by the model, but computed by suitable parameterizations.

The time-scales of atmospheric turbulence are extremely large and have a range between milliseconds and hours. Typical time steps in mesoscale models are in the order of seconds. However, it does not make any sense to compare computed time-series of e.g. wind speeds

Detailed model description - Basic equations

based on the modelling time-steps with observations because of several reasons: (i) especially at the beginning of a simulation the solution is not converged and, thus, does not represent a physical meaningful solution. (ii) depending on the spatial resolution of mesoscale models certain eddies cannot be resolved, although they could in principle be resolved with regard to the time-step (e.g. convective vertical eddies during daytime). (iii) improper boundary conditions prevent model solutions to be physically meaningful for time-scales equal to the modelling time-step. (iv) Reynolds-averaging requires parameterizations of turbulent quantities. These empirical relationships are mostly derived and tested for averaging times between half an hour and one hour.

Therefore, the GRAMM model is suitable to provide wind-, temperature-, humidity-fields for averaging times of about 10 minutes up to 1 hour. It is not suited to compute e.g. wind gusts.

Conservation of mass:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial (\bar{\rho} \bar{u}_i)}{\partial x_i} + \frac{\partial (\bar{\rho}' u'_i)}{\partial x_i} = 0 \quad (\text{E } 3)$$

Conservation equation for the potential temperature θ :

$$\frac{\partial (\bar{\rho} \bar{\theta})}{\partial t} + \frac{\partial (\bar{\rho} \bar{u}_j \bar{\theta})}{\partial x_j} = S_\theta - \frac{\partial (\bar{\rho} u'_j \theta')}{\partial x_j} \quad (\text{E } 4)$$

The potential temperature is defined by:

$$\bar{\theta} = \bar{T} \left(\frac{\bar{p}}{1000 hPa} \right)^{-R/c_p} \quad (\text{E } 5)$$

where R is the universal gas constant and c_p is the specific heat capacity of dry air.

Conservation equation for the specific humidity:

$$\frac{\partial (\bar{\rho} \bar{q})}{\partial t} + \frac{\partial (\bar{\rho} \bar{u}_j \bar{q})}{\partial x_j} = S_q - \frac{\partial (\bar{\rho} u'_j q')}{\partial x_j} \quad (\text{E } 6)$$

S_θ and S_q are source terms for the potential temperature and the specific humidity, respectively. Sources for the potential temperature might be anthropogenic heat, solar radiation, or soil heat flux, etc.

4.3 Approximations

For optimizing the computation of temperature- and pressure fields with regard to numerics, it is sensible to define a reference state Ψ_0 , which is assumed to be steady-state and horizontally homogenous. The deviation Ψ^* between a reference state Ψ_0 and the time-averaged quantity $\bar{\Psi}$ is defined as

$$\Psi^* = \bar{\Psi} - \Psi_0 \quad (\text{E } 7)$$

and will be used subsequently in the conservation equations.

Hydrostatic assumption:

Mass conservation in the GRAMM model is enforced by correcting the flow field by a pressure field after each iteration as will be outlined later in this document. The gravitational acceleration in the vertical momentum equation in combination with the constraint of zero flux into the surface, would just result in the well-known barometric equation for air pressure. Towards sea level the pressure is in the order of 10^5 Pa, while the pressure field needed to correct the flow field to meet mass conservation is often in the order of only a few Pascal. Thus, it makes sense to compute an initial, steady-state of the barometric pressure distribution at the beginning of the computation by:

$$\frac{\partial p_0}{\partial z} = -g\rho_0 \quad (\text{E } 8)$$

Due to this, the momentum equation in the vertical direction can be rearranged into:

$$\frac{\partial(\bar{\rho}\bar{u}_3)}{\partial t} + \bar{\rho}\frac{\partial(\bar{u}_3\bar{u}_j)}{\partial x_j} = -\frac{\partial p^*}{\partial x_3} - f_j\bar{\rho}\varepsilon_{3jk}\bar{u}_k - \rho^*g - \bar{\rho}\frac{\partial(\bar{u}'_3\bar{u}'_j)}{\partial x_j} \quad (\text{E } 9)$$

In addition to the introduction of a reference state, the pressure is indeed separated into three parts:

Detailed model description - Approximations

$$\bar{p} = p_0 + p_g + p^* \quad (\text{E } 10)$$

p_g is the large-scale horizontal pressure, which is expressed utilizing the geostrophic balance equations

$$\frac{\partial \bar{p}_g}{\partial x_1} = \bar{\rho} f_2 \bar{u}_{2g} \quad \text{and} \quad \frac{\partial \bar{p}_g}{\partial x_2} = \bar{\rho} f_2 \bar{u}_{1g}, \quad (\text{E } 11)$$

where \bar{u}_{ig} are the geostrophic wind components. The large scale forcing of the GRAMM model is carried out using the geostrophic wind, which is much easier to implement than pressure gradients. The horizontal momentum equations can be expressed (here shown for the u_1 -component only):

$$\frac{\partial(\bar{\rho} \bar{u}_1)}{\partial t} + \frac{\partial(\bar{\rho} \bar{u}_1 \bar{u}_j)}{\partial x_j} = -\bar{\rho} f_3 (\bar{u}_{2g} - \bar{u}_2) - \bar{\rho} f_2 \bar{u}_3 - \bar{\rho} \frac{\partial(\bar{u}'_1 \bar{u}'_j)}{\partial x_j} \quad (\text{E } 12)$$

Due to the weakness of the Coriolis-terms in conjunction with the vertical wind speed, the Coriolis terms can be neglected. The pressure gradient due to the non-hydrostatic pressure

$\frac{\partial p^*}{\partial x_1}$ results whenever the flow does not fulfil mass conservation at any location. Neglecting

this term in the momentum equations just leads to the circumstance that the non-hydrostatic pressure does not converge towards zero but approaches a constant value in each grid cell

that will not equal zero in most cases. It was found that excluding $\frac{\partial p^*}{\partial x_1}$ in the momentum

equation leads to better flow convergence in many cases.

Anelastic approximation:

In order to prevent the computation of sound-waves, which do not have a meteorological effect, it is assumed that $\bar{\rho} = \rho_0$ in all conservation equations, except for the momentum equation in the vertical direction. In doing so, the conservation equation for mass simplifies to:

$$\frac{\partial \cdot (\rho_0 \bar{u}_i)}{\partial x_i} = 0 \quad (\text{E } 13)$$

Boussinesq approximation:

Buoyancy effects in the vertical momentum equation are accounted for by using the so-called Boussinesq approximation (e.g. Dutton and Fichtel, 1969). First, the perfect equation of state for air mixed with water vapour \bar{q}

$$p_0 = \rho_0 R T_0 \left[1 + \left(\frac{R_1}{R} - 1 \right) \bar{q} \right] \quad (\text{E } 14)$$

is linearized around the reference state to obtain

$$\frac{\rho^*}{\rho_0} = \frac{p^*}{p_0} - \frac{T^*}{T_0} \quad (\text{E } 15)$$

For shallow convection (<1 km) one can assume $p^*/p_0 \ll T^*/T_0$. Using the definition of the potential temperature gives

$$\frac{\rho^*}{\rho_0} = -\frac{\theta^*}{\theta_0} \quad (\text{E } 16)$$

Non-hydrostatic pressure:

Finally, the non-hydrostatic pressure p^* is computed diagnostically by using

$$\frac{\partial^2 p^*}{\partial x_i^2} = \frac{\partial}{\partial t} \left(\frac{\partial \rho_0 \bar{u}_i}{\partial x_i} \right), \quad (\text{E } 17)$$

which is derived by combining the conservation equations for mass and momentum.

4.4 Turbulence parameterizations

The terms $\overline{u_i' u_j'}$ (= Reynolds stress tensor) in the RANS equations require further modelling. There exists a vast number of methods to compute the Reynolds stresses (e.g. Stull, 1988). In the GRAMM model, an eddy viscosity model is utilized, where it is assumed (Boussinesq, 1897) that the Reynolds stresses can be treated analogous to the viscous stresses in laminar flows:

$$-\overline{u_i' u_j'} = \mu_t \left(\frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial \bar{u}_i}{\partial x_j} \right) - \frac{2}{3} k \delta_{ij} \quad (\text{E } 18)$$

Detailed model description - Turbulence parameterizations

μ_t is the so-called eddy viscosity or turbulent exchange coefficient, and k is the turbulent kinetic energy which ensures that the sum of the normal stresses $\overline{u_1'^2} + \overline{u_2'^2} + \overline{u_3'^2} = 2k$, which can easily be demonstrated:

$$\overline{u_1'^2} + \overline{u_2'^2} + \overline{u_3'^2} = -2\mu_t \left(\frac{\partial \overline{u_1}}{\partial x_1} + \frac{\partial \overline{u_2}}{\partial x_2} + \frac{\partial \overline{u_3}}{\partial x_3} \right) + 2k \quad (\text{E } 19)$$

As incompressibility of the flow is assumed $\frac{\partial \overline{u_1}}{\partial x_1} + \frac{\partial \overline{u_2}}{\partial x_2} + \frac{\partial \overline{u_3}}{\partial x_3} = 0$. The specific relationship between the Reynolds stresses and the mean velocity gradients of the form $-\overline{u_1' u_2'} \sim \frac{\partial \overline{u_2}}{\partial x_1} + \frac{\partial \overline{u_1}}{\partial x_2}$ instead of using a simpler form of e.g. $-\overline{u_1' u_2'} \sim \frac{\partial \overline{u_1}}{\partial x_2}$ ensures that the stress tensor remains symmetric.

The turbulent kinetic energy is defined by:

$$k = \frac{1}{2} \overline{u_i u_i} \quad (\text{E } 20)$$

It should be noted that the term $-\frac{2}{3} k \delta_{ij}$ in (E 18) is not computed explicitly but is implicitly considered in the calculation of the non-hydrostatic pressure.

Two different methods are implemented in GRAMM to compute the turbulent viscosity μ_t :

4.4.1 Algebraic turbulence model according to Pandolfo (1969)

A simple algebraic relation is used based on the mixing length theory. The so-called mixing length l is presumed to represent the typical length over which one can find a high interaction of vortices in a turbulent flow field. The turbulent viscosity μ_t is estimated from

$$\mu_t = l^2 \sqrt{0.5 \left[\left(\frac{\partial \overline{u_1}}{\partial z} \right)^2 + \left(\frac{\partial \overline{u_2}}{\partial z} \right)^2 \right]} \cdot f(Ri_g), \quad (\text{E } 21)$$

where the mixing length l is determined according to Blackadar (1962):

$$l = \left(\frac{1}{l^\infty} + \frac{1}{\kappa[z - z_1]} \right)^{-1} \quad (\text{E } 22)$$

Various methods exist to compute the maximum mixing length l^∞ . Pielke (1984) suggests a fixed value of 90 m, while others suggest a dependence on local scaling parameters, such as geostrophic wind speed or – as used in the GRAMM model – the friction velocity:

$$l^\infty = \frac{0.007u_*}{f_3} \quad (\text{E } 23)$$

The Gradient-Richardson number is defined as:

$$Ri_g = \frac{1}{\left(\frac{\partial \bar{u}_1}{\partial z} \right)^2 + \left(\frac{\partial \bar{u}_2}{\partial z} \right)^2} \cdot \frac{g \partial \theta}{\theta \partial z} \quad (\text{E } 24)$$

$$f(Ri_g) = \begin{cases} \left(\frac{1 - 3Ri_g}{1 + 3Ri_g} \right)^2 & 0 < Ri_g \leq 1/3 \\ C_1 (-Ri_g)^{0.33} & -0.048 < Ri_g \leq 0 \\ C_1 (-Ri_g)^{0.33} & -0.048 \geq Ri_g \end{cases} \quad \text{for} \quad \begin{cases} 0 < Ri_g \leq 1/3 \\ -0.048 < Ri_g \leq 0 \\ -0.048 \geq Ri_g \end{cases} \quad (\text{E } 25)$$

In case of $Ri_g > 1/3$ the turbulent viscosity takes its default minimum value of $0.05 \text{ m}^2 \text{ s}^{-2}$.

4.4.2 Turbulent closure of second order

In this case μ_t is computed by applying the standard k - ε model, where ε is the turbulent dissipation rate:

$$\frac{\partial(\rho \bar{k})}{\partial t} + \frac{\partial(\rho \bar{u}_j \bar{k})}{\partial x_j} = \frac{\partial}{\partial x_j} \mu_\phi \rho \left(\frac{\partial \bar{k}}{\partial x_j} \right) + P_m + P_t - \bar{\varepsilon} \quad (\text{E } 26)$$

$$\frac{\partial \rho \bar{\varepsilon}}{\partial t} + \frac{\partial \rho \bar{u}_j \bar{\varepsilon}}{\partial x_j} = \frac{\partial}{\partial x_j} \mu_\phi \rho \left(\frac{\partial \bar{\varepsilon}}{\partial x_j} \right) + \frac{\bar{\varepsilon}}{k} (1.44 \cdot (P_m + P_b) - 1.92 \rho \cdot \bar{\varepsilon}) \quad (\text{E } 27)$$

Detailed model description - Turbulence parameterizations

$$\mu_t = 0.09 \frac{k}{\varepsilon} \quad (\text{E } 28)$$

P_m is the production term for turbulent kinetic energy due to shear stresses, and P_b is the production term for turbulent kinetic energy due to buoyancy given by:

$$P_m = \mu_\phi \rho \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \frac{\partial \bar{u}_i}{\partial x_j} \quad (\text{E } 29)$$

$$P_b = 1.35 \cdot \mu_\phi \rho \cdot \frac{g}{\theta_0} \frac{\partial \bar{\theta}}{\partial x_3} \quad (\text{E } 30)$$

The ratio of the eddy viscosity for momentum and any other scalar is often defined as a constant value represented by the turbulent Schmidt number S_{Ct} for the entire turbulent flow. In GRAMM a ratio of 0.9 was chosen:

$$S_{Ct} = \frac{\mu_\phi}{\mu_t} = 0.9 \quad (\text{E } 31)$$

The standard k - ε model is used throughout the model domain also near the Earth's surface. As outlined e.g. in Apsley (2016), the standard k - ε model is in principle consistent within the log-law region near the surface given that the grid is coarse enough (which in mesoscale models is always the case), and turbulence is in local equilibrium (i.e. $P_m = \varepsilon$), and P_b is negligible. Local equilibrium is often not maintained (e.g. in reattachment zones), thus, the model as implemented in GRAMM could be improved with respect to this. Nevertheless, it was found that the diagnostic equations used by Hurley (2005) for k and ε do not lead to numerically stable results in all circumstances and thus are not used in the current model version.

$$k = 0.09^{-0.5} u_*^2 + 0.5 w_*^2 \quad (\text{E } 32)$$

$$\varepsilon = \frac{u_*^3}{\kappa z} \phi_m - \frac{g}{\theta} u_* \theta_{v*} \quad (\text{E } 33)$$

$$w_* = \left(\frac{-gz_i u_* \theta_{v*}}{\theta_v} \right)^{0.33}, \quad (\text{E } 34)$$

with w_* being the convective velocity scale, θ_{v*} the characteristic potential virtual temperature, z_i the mixing height in convective conditions, and ϕ_m are the flux-profile relationships obtained from surface-layer similarity theory (see chapt. 4.5.1).

4.5 Boundary conditions

4.5.1 Lower boundary

For wind speed no-slip boundary conditions are implemented. Boundary conditions for the turbulent fluxes are determined by Monin-Obukhov surface layer scaling variables (i.e. friction velocity u_* and characteristic potential temperature θ_*) with stability functions from Businger et al. (1971). In the following equations the subscript s refers to surface values, and z_1 is the first vertical model grid point above the surface.

Detailed model description - Boundary conditions

$$u_* = \kappa \left(\bar{u}^{-2} + \bar{v}^{-2} \right)_s^{1/2} \cdot \left[\ln \left(\frac{z_1}{z_0} \right) - \psi_m \right]^{-1} \quad (\text{E } 35)$$

$$\theta_* = \kappa \left[\theta(z_1) - \theta_s \right] \cdot \left[\ln \left(\frac{z_1}{z_0} \right) - \psi_m \left(\frac{z_1}{L} \right) \right]^{-1} \quad (\text{E } 36)$$

$$\psi_m = \begin{cases} 2 \cdot \ln \left[(1 + \phi_m^{-1}) / 2 \right] + \ln \left[(1 + \phi_m^{-2}) / 2 \right] - 2 \arctan(\phi_m^{-1}) + \pi / 2 & \text{for } \frac{z_1}{L} \leq 0 \\ -4.7 \cdot \frac{z_1}{L} & \text{for } \frac{z_1}{L} > 0 \end{cases} \quad (\text{E } 37)$$

$$\psi_h = \begin{cases} 2 \cdot \ln \left[\left(1 + \left(1 - 16 \left(\frac{z_1}{L} \right) \right)^{0.5} \right) / 2 \right] & \text{for } \frac{z_1}{L} \leq 0 \\ -4.7 \cdot \frac{z_1}{L} & \text{for } \frac{z_1}{L} > 0 \end{cases} \quad (\text{E } 38)$$

$$\phi_m = \begin{cases} \left(1 - 16 \frac{z_1}{L} \right)^{-1/4} & \text{for } \frac{z_1}{L} \leq 0 \\ 1 + 4.7 \cdot \frac{z_1}{L} & \text{for } \frac{z_1}{L} > 0 \end{cases} \quad (\text{E } 39)$$

$$\phi_h = \begin{cases} \phi_m & \text{for } \frac{z_1}{L} > 0 \\ \phi_m^2 & \text{for } \frac{z_1}{L} \leq 0 \end{cases}$$

κ is the von Kármán constant (0.35), $L = \frac{u_*^2 \theta}{g \kappa \theta_*}$ the Obukhov length, and z_0 the roughness length. The latter is computed based on land-use data, which is a weighted average according to:

$$z_0 = \frac{1}{A_{tot}} \cdot A_n z_0^n \quad (\text{E } 40)$$

A_{tot} is the total area of a grid cell, and A_n is the share of each land-use-category within a grid cell.

Note that the Obukhov-length L is limited to the range $5m \leq |L| \leq 1000m$ and the friction velocity is limited by $0.15 \text{ ms}^{-1} \leq u_* \leq 2 \text{ ms}^{-1}$.

The sensible Q_s and latent Q_l heat fluxes are computed by

$$Q_s = -\bar{\rho} c_p \frac{(\bar{T}_1 - \bar{T}_s)}{r_H} \quad (\text{E 41})$$

$$Q_l = -\bar{\rho} \lambda \frac{(\bar{q}_1 - \bar{q}_s)}{r_H} \quad (\text{E 42})$$

r_H is the aerodynamic resistance given by:

$$r_H = \frac{\ln\left(\frac{z_1}{z_0} - \psi_h\right)}{\kappa u_*} \quad (\text{E 43})$$

The term $e_*[T_s]$ represents the saturation vapour pressure evaluated at the surface temperature T_s . λ is the evaporation heat of water:

$$\lambda = 2500000 - 2300T_s$$

The specific humidity at the surface is computed by

$$\bar{q}_s = \begin{cases} \bar{q}_* & \text{if } \bar{q}_* \leq \bar{q}_1 \\ f_w \bar{q}_* + (1 - f_w) \bar{q}_1 & \text{if } \bar{q}_* > \bar{q}_1 \end{cases}, \quad (\text{E 43})$$

where f_w is the fractional soil water content. In the GRAMM model, f_w is a constant value depending solely on the landuse-category defined by CORINE classes (e.g. $f_w = 1$ for water bodies, or $f_w = 0.4$ for forests).

Specific humidity is related to the saturation vapour pressure $e_*[T_s]$ as

$$\bar{q}_* = \frac{0.622 e_*[T_s]}{p - 0.378 e_*[T_s]}, \quad (\text{E 44})$$

where the saturation vapour pressure $e_*[T_s]$ is computed by

Detailed model description - Boundary conditions

$$e_*[T_s] = 610.78 e^{\frac{17.269(T_s - 273.16)}{T_s - 35.86}} \quad (\text{E } 45)$$

For numerical reasons (i.e. the diffusion terms $\bar{\rho} \partial / \partial x_j$ are computed once at the beginning of each iteration and are equal for all conserved quantities) the no-slip condition for wind speed is implemented by using:

$$\begin{aligned} \bar{\rho} \frac{\partial (\overline{u'_1 u'_3})}{\partial x_3} &= \bar{\rho} \frac{\partial}{\partial x_3} (u_* u_{*1}), \text{ and} \\ \bar{\rho} \frac{\partial (\overline{u'_2 u'_3})}{\partial x_3} &= \bar{\rho} \frac{\partial}{\partial x_3} (u_* u_{*2}) \end{aligned} \quad (\text{E } 46)$$

$$\begin{aligned} u_{*1} &= \kappa u_{1s} \cdot \left[\ln \left(\frac{z_1}{z_0} \right) - \psi_m \right]^{-1}, \text{ and} \\ u_{*2} &= \kappa u_{2s} \cdot \left[\ln \left(\frac{z_1}{z_0} \right) - \psi_m \right]^{-1} \end{aligned} \quad (\text{E } 47)$$

Thus, the diffusion terms $\bar{\rho} \frac{\partial (\overline{u'_1 u'_3})}{\partial x_3}$ otherwise computed directly from the spatial gradients of the mean wind components and the turbulent viscosity are modelled using the equations listed above.

4.5.2 Lateral boundaries

In the GRAMM model two different lateral boundary conditions are provided for the horizontal wind components:

Homogenous Neumann condition:

$$\frac{\partial u_{1,2}}{\partial x_{1,2}} = 0 \quad (\text{E } 48)$$

Nudging towards large-scale values:

Here the boundary values are forced towards their prescribed large-scale values \tilde{u} :

$$u_{1,2} = u_{1,2} - \alpha(u_{1,2} - \tilde{u}_{1,2}) \quad (\text{E } 49)$$

$$\alpha = e^{-2\Delta n} \quad (\text{E } 50)$$

Δn is the number of grid points between the grid point of interest and the grid point at the boundary. Currently, Δn is limited to six grid points. Note that first the values at the boundaries are computed using homogenous Neumann conditions.

For all quantities other than the horizontal wind components, homogenous Neumann conditions are applied.

4.5.3 Upper boundary

The wind components are nudged towards their prescribed large-scale values using a slightly different expression than for the lateral boundary conditions:

$$\phi = \phi - \alpha(\phi - \tilde{\phi}) \quad (\text{E } 51)$$

$$\alpha = 1 - \tanh(1.5\Delta n) \quad (\text{E } 52)$$

Δn is limited to three grid points only. In case of quantities other than the wind components, homogenous Neumann conditions are used.

4.6 Surface energy balance

4.6.1 Solar radiation

The model used in GRAMM is based on the work of Somieski (1988). It represents a simplified tool to compute radiation densities at the surface and atmospheric heating rates. Short- and long-wave radiation are treated separately and are influenced by the inclination of orography in x - and y -directions, the slope line, and the azimuth of the slope line, which is defined as the deviation from geographic north. Shadowing effects of surrounding orography is taken into account.

Solar radiation is determined from the sun's angle and the time of the day (so-called hour angle). The average radiation density outside the Earth's atmosphere is set to 1367 W m^{-2} . Direct and diffusive radiation are distinguished; both of them become zero when the sun is below the horizon or when the point of interest is inside shaded areas.

Detailed model description - Surface energy balance

In order to determine the diffusive and absorbing properties of the Earth's atmosphere, the vertical integrated particle-extinction coefficients (for an aerosol size of 0.55 μm) and the gross water-vapour content, as well as those for liquid water and ice are calculated. For this, the governing parameters are the aerosol optical depth of a cloud-free atmosphere, the density of air, the concentrations of water vapour, liquid water, rain, ice and snow. The computed integrated values at a certain height are made for the column above and below the actual point of interest.

Further, the atmosphere's albedo at each altitude is required to model the direct and diffusive part of solar radiation. To do so, the Linke turbidity coefficient Y , which describes the number of clean atmospheres causing the turbidity, is utilized. Typically, it ranges between 1 – 8 (Ineichen, 2008). The albedo is a function of the modified surface albedo A_{Gm} , the cloud albedo A_C , and the transmission functions T^* :

$$T^* = 0,3 + (0,1r_r)^{0,2}(0,75 + \mu) + 8\frac{P_0}{P}\left(Y + 0,1\frac{P}{P_0} + 0,02\right) \quad (\text{E } 53)$$

The total radiation density E , also consisting of a diffusive D and a direct S part, is modelled based on a scheme that is mainly applicable for thin cloud layers. The main governing parameter is the cloud optical thickness. The influence of clouds is taken into account by using an empirical relationship according to (Greuell et al., 1997), whereby cl is the cloudiness ($0 \leq cl \leq 1$).

$$E = S + D$$

$$S = I\mu_e n_s f_{cl} \quad (\text{E } 54)$$

$$D = I\mu(n_D + j)f_A f_{cl}$$

$$f_{cl} = 1 - 0.233cl - 0.415cl^2$$

n_s , n_D , and j are transmission functions, while f_A is a correction function to take into account the influence of albedo on the diffusive radiation density. μ_e is the sinus of the sun's angle, and $\mu = \text{Max}(\mu_e, 0.02)$.

The total solar radiation R_{SolG} at the surface is computed as a function of its direct and diffusive parts S and D , the sky-view factor $(1 - Q)$, the angle of incidence i_e , and the surface albedo A_G .

$$R_{SolG} = (1 - A_G)(S_G + D_{SG} + D_{DG} + B_G) \quad (E 55)$$

$$S_G = \frac{S}{\mu_e} \sin i_e \quad (E 56)$$

$$D_{SG} = x_h \frac{\sin i_e}{\mu} D \quad (E 57)$$

$$D_{DG} = (1 - x_h)(1 - Q)D \quad (E 58)$$

$$B_G = A_G Q(S + D) \quad (E 59)$$

$$x_h = 1.5\mu s \left(0.65 + 0.04Y \frac{p}{p_0} - 0.6s \right) \quad (E 60)$$

The subscript G in general denotes surface values, thus, S_G is the direct solar radiation hitting the surface, D_{SG} is the direct diffusive radiation, D_{DG} is the isotropic diffusive share, and B_G is the radiation originating from the surrounding terrain. In case that i_e is below zero or if the surface lies in a shaded area, S_G and D_{SG} are equal to zero.

Atmospheric heating $\left(\frac{\partial T}{\partial t}\right)_{sol}$ due to solar radiation outside and inside clouds is modelled using different approaches, where the outside region is defined by liquid water contents below 10^{-2} g m^{-3} and ice fractions below 10^{-3} g m^{-3} . In a cloud-free atmosphere, heating rates in the grid volume with vertical extension Δx_3 are computed via:

$$\left(\frac{\partial T}{\partial t}\right)_{sol} = \frac{0.7}{\rho c_p \Delta x_3} \left(E \left(x_3 + \frac{\Delta x_3}{2} \right) - E \left(x_3 - \frac{\Delta x_3}{2} \right) \right) \left(1 + A \left(x_3 - \frac{\Delta x_3}{2} \right) \right) \quad (E 61)$$

c_p is the specific heat capacity of air for constant pressure. The empirical factor of 0.7 is based on the assumption that 70 % of the extinction in a clear atmosphere results from absorption. Further, it is assumed that the albedo at the vertical boundary surfaces of each grid cell are equal:

$$A\left(x_3 - \frac{\Delta x_3}{2}\right) \cong A\left(x_3 + \frac{\Delta x_3}{2}\right) \quad (\text{E } 62)$$

4.6.2 Longwave radiation

In a first step, emissivities are computed based on vertically integrated amounts of water vapour, carbon dioxide, liquid water, and ice. These are combined to the emissivity for clear air ε_{clear} and the total atmospheric emissivity ε_{tot} .

$$\varepsilon_{clear} = 1.03(\varepsilon_r + \varepsilon_{CO_2}) \quad (\text{E } 63)$$

$$\varepsilon_{tot} = 1 - (1 - \varepsilon_{clear})(1 - \varepsilon_c) \quad (\text{E } 64)$$

The downward terrestrial radiation flux L' at the surface is obtained via summation from the surface up to a black-cloud ($\Delta W > W_{crit}$) representing the top boundary.

$$L' = \sum \sigma T^4 \left(\varepsilon_{tot} \left(x_3 + \frac{\Delta x_3}{2}, G \right) - \varepsilon_{tot} \left(x_3 - \frac{\Delta x_3}{2}, G \right) \right) \quad (\text{E } 65)$$

σ is the Stefan-Boltzmann constant and $\varepsilon_{tot} \left(x_3 + \frac{\Delta x_3}{2}, G \right)$ is the total atmospheric emissivity between the top of a layer with vertical extension Δx_3 , temperature T , and the surface. Considering surface properties such as the sky-view factor $(1 - Q)$, one obtains for the downward terrestrial flux

$$L = \hat{L}(Q + 0.03 \sin \alpha) + \varepsilon_a \sigma T_1^4 (1 - Q) \quad (\text{E } 66)$$

where T_1 is the air temperature in the lowest atmospheric layer, and ε_a is the emissivity of the atmosphere. The term $0.03 \sin \alpha$ is introduced to include anisotropic effects of the downward radiation. The atmospheric emissivity depends on the cloudiness cl and the emissivity under clear sky ε_{cs} and overcast sky ε_{oc} , respectively.

$$\varepsilon_a = \varepsilon_{cs}(1 - cl^b) + \varepsilon_{oc}cl^b \quad (\text{E } 67)$$

Parameter b is set equal to 2.0 (Greuell et al., 1997). The total longwave radiative flux at the surface is given by:

$$R_{terr,G} = L - \varepsilon_G \sigma T_G^4 \quad (E\ 68)$$

ε_G is the surface emissivity and T_G is the surface temperature.

Atmospheric heating rates due to longwave terrestrial radiation flux are calculated on the assumption that a clear atmosphere has the same temperature as the layer for which the heating rate is computed. Deviations from this assumption are permitted in the following cases:

Top boundary of the atmosphere: $\varepsilon_a = \varepsilon_{top}$ $T_a = T_{top}$

Lower boundary of the atmosphere: $\varepsilon_b = \varepsilon_G$ $T_b = T_G$

The subscripts a and b denote target layers above and below the actual target layer. ε_{top} and T_{top} are the emissivity and temperature at the top boundary of the atmosphere.

4.6.3 Soil heat flux

The soil heat flux G is obtained by

$$G = -\lambda \frac{\partial T(x_3)}{\partial x_3} \quad (E\ 69)$$

λ is the soil conductivity and $\frac{\partial T(x_3)}{\partial x_3}$ is the temperature gradient within the soil close to the surface. It is assumed that in 1 m depth no significant daily or seasonal temperature variations take place. This so-called deep soil temperature is adjusted for terrain height using the relationship

$$T_{1m} = T_{1m,init} - 0.005x_3 \quad (E\ 70)$$

$T_{1m,init}$ is either user specified or estimated as an empirical function of the latitude:

$$T_{1m,init} = 293 - 0.005\varphi^2 + 0.006\varphi \quad (E\ 71)$$

As shown in Fig. 2, at sea level temperatures range between 20°C and -20°C at the Equator and the North Pole, respectively. Permafrost is found at around 70 N. In Alpine regions (~47

Detailed model description - Surface energy balance

deg. North) permafrost is found at around 2.000 m above sea level, whereas at the Equator the permafrost zone rises up to around 4.000 m.

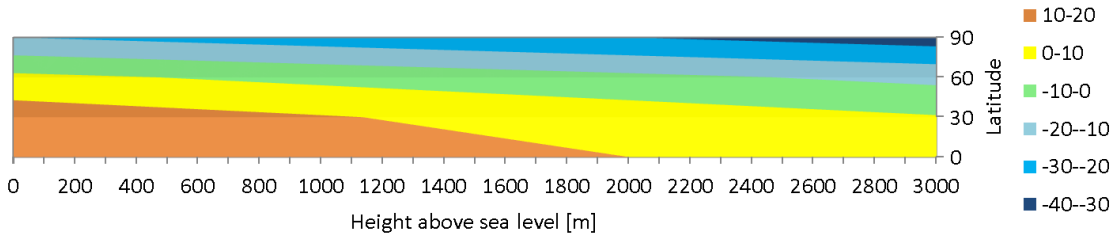


Fig. 2. Dependency of the deep soil temperature on altitude and latitude

Soil temperatures above the constant deep soil temperature level are modelled by

$$\frac{\partial T_G}{\partial t} = \frac{\partial}{\partial x_3} \left(\frac{\lambda}{\rho_B c_B} \frac{\partial T(x_3)}{\partial x_3} \right), \quad (\text{E } 72)$$

with ρ_B being soil density and c_B the specific heat capacity of soil.

Land-use characteristics are determined on the basis of CORINE land-use categories as listed in Table 1. Note, that a user has the possibility to amend each of these values if necessary. It should be noted that the surface albedo is not taken as constant value in the radiation model, but calculated as a function of the sun's zenith angle.

4.6.4 Snow cover

The influence of snow cover on the surface energy balance is taken into when the snow depth is larger than 0.15 m, and larger than the roughness length. In urban areas and over water bodies snow cover is generally not considered.

In case of snow cover, the surface albedo is set to 0.6, the surface maximum temperature is limited to 273 K, and the soil heat flux in the upper layer is set zero.

Table 1: Land-use characteristics based on CORINE land-use categories

Code	description	albedo	emissivity	Soil moisture	roughness length [m]	Heat conductivity [W/m/K]	Thermal conductivity [m ² /s]
111	Continuous urban fabric	0.25	0.95	0.03	1.5000	4.0	2.00E-06
112	Discontinuous urban fabric	0.25	0.95	0.03	0.5000	4.0	1.30E-06
121	Industrial or commercial units	0.25	0.95	0.03	0.5000	4.0	1.30E-06
122	Road and rail networks and associated land	0.25	0.95	0.03	0.3000	4.0	1.30E-06
123	Port areas	0.25	0.95	0.03	1.0000	4.0	1.30E-06
124	Airports	0.25	0.95	0.03	0.2000	4.0	1.30E-06
131	Mineral extraction sites	0.25	0.95	0.03	0.2000	2.0	1.30E-06
132	Dump sites	0.25	0.95	0.03	0.2000	2.0	1.30E-06
133	Construction sites	0.25	0.95	0.03	0.3000	2.0	1.30E-06
141	Green urban areas	0.19	0.92	0.10	0.3000	1.0	7.00E-07
142	Sport and leisure facilities	0.19	0.92	0.10	0.3000	1.0	7.00E-07
211	Non-irrigated arable land	0.19	0.92	0.10	0.1000	0.5	7.00E-07
212	Permanently irrigated land	0.19	0.92	0.50	0.1000	1.5	7.00E-07
213	Rice fields	0.19	0.92	0.50	0.1000	2.0	7.00E-07
221	Vineyards	0.19	0.92	0.10	0.1500	0.5	7.00E-07
222	Fruit trees and berry plantations	0.19	0.92	0.10	0.2500	0.5	7.00E-07
223	Olive groves	0.19	0.92	0.05	0.3000	0.5	7.00E-07
231	Pastures	0.19	0.92	0.10	0.1000	0.5	7.00E-07
241	Annual crops associated with permanent crops	0.19	0.92	0.10	0.1000	0.5	7.00E-07
242	Complex cultivation patterns	0.19	0.92	0.10	0.2000	0.5	7.00E-07
243	Land principally occupied by agriculture, with significant areas of natural vegetation	0.19	0.92	0.10	0.2000	0.5	7.00E-07
244	Agro-forestry areas	0.17	0.95	0.40	1.0000	0.5	8.00E-07
311	Broad-leaved forest	0.16	0.95	0.40	1.0000	0.5	8.00E-07
312	Coniferous forest	0.12	0.95	0.40	1.0000	0.5	8.00E-07
313	Mixed forest	0.14	0.95	0.40	1.0000	0.5	8.00E-07
321	Natural grasslands	0.15	0.92	0.10	0.0200	0.5	1.00E-06
322	Moors and heathland	0.15	0.92	0.10	0.0200	2.7	1.00E-06
323	Sclerophyllous vegetation	0.15	0.92	0.02	0.0500	0.5	1.00E-06
324	Transitional woodland-shrub	0.15	0.92	0.10	0.0200	0.5	1.00E-06

Detailed model description - Surface energy balance

331	Beaches, dunes, sands	0.25	0.95	0.60	0.0500	0.3	1.00E-06
332	Bare rocks	0.15	0.92	0.01	0.1000	1.5	1.00E-06
333	Sparsely vegetated areas	0.15	0.92	0.01	0.0100	0.5	1.00E-06
334	Burnt areas	0.15	0.92	0.05	0.1000	0.3	1.00E-06
335	Glaciers and perpetual snow	0.6	0.95	0.10	0.0100	1.0	5.00E-07
411	Inland marshes	0.14	0.95	0.70	0.0100	20.0	1.00E-06
412	Peat bogs	0.14	0.95	0.70	0.0100	20.0	1.00E-06
421	Salt marshes	0.5	0.95	0.70	0.0100	20.0	1.00E-06
422	Salines	0.5	0.95	0.70	0.0100	20.0	1.00E-06
423	Intertidal flats	0.14	0.95	0.70	0.0100	20.0	1.00E-06
511	Water courses	0.08	0.98	1.00	0.0001	100.0	1.00E-06
512	Water bodies	0.08	0.98	1.00	0.0001	100.0	1.00E-06
521	Coastal lagoons	0.08	0.98	1.00	0.0001	100.0	1.00E-06
522	Estuaries	0.08	0.98	1.00	0.0001	100.0	1.00E-06
523	Sea and ocean	0.08	0.98	1.00	0.0001	100.0	1.00E-06

4.7 Numerical schemes

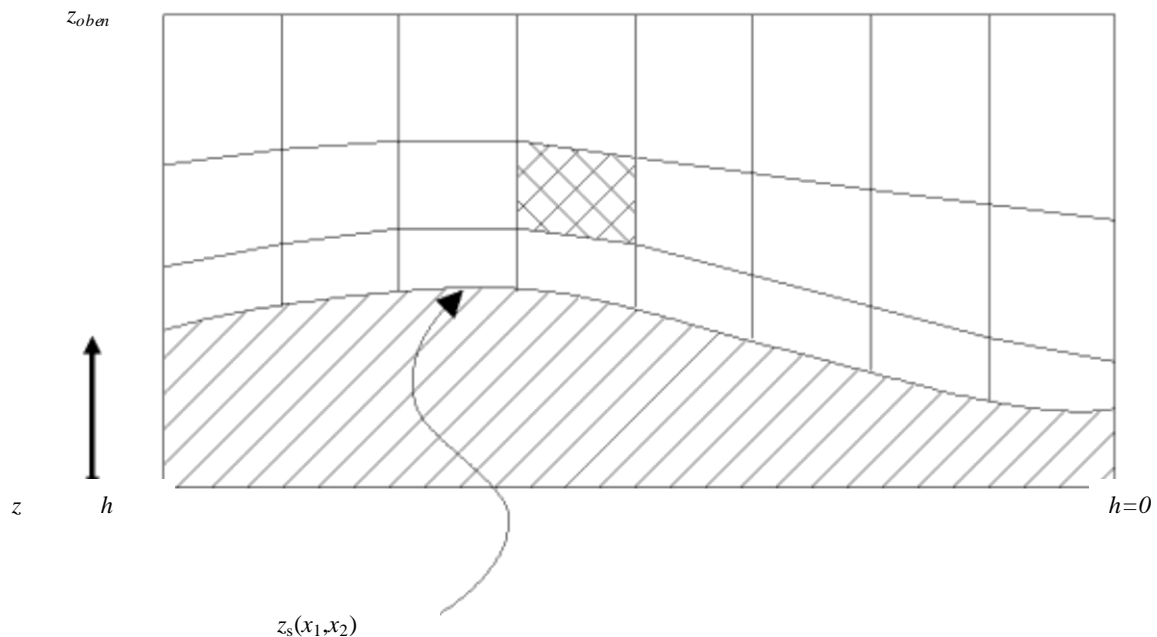
4.7.1 Discretization grid

While in the horizontal directions equidistant grid sizes are used in the entire model domain, in the vertical a terrain-following coordinate with increasing grid sizes from the surface up to the top model boundary is utilized:

$$h(x_3) = z_{top} \frac{x_3 - z_s(x_1, x_2)}{z_{top} - z_s(x_1, x_2)} \quad (\text{E } 73)$$

$z_s(x_1, x_2)$ is the height of orography, x_3 is the height of the undistorted model grid, and z_{top} is the top model boundary ($=z_{oben}$ in Fig. 3).

Fig. 3. Terrain-following grid in the GRAMM model



The user may specify a zone at the lateral model-domain boundaries, where orography is smoothed. Typically, such a zone extends a few kilometres into the model domain, depending on the orography. Smoothing is carried out by setting all outmost cell heights equal to the lowest orographic height encountered at the lateral boundaries. In a further step, orography is linear interpolated between the grid point located at the boundary and the actual orography at a certain point, located inside the model domain at a perpendicular distance to the lateral boundary, which is specified by the user via the number of grid-cells to be considered in the smoothing process.

4.7.2 Discretization of the conservation equations for momentum

In complex orography, the shape of grid cells is usually a distorted hexahedron, hereafter referred to as primary grid (Fig. 4). In order to discretize the conservation equations for specific humidity, potential temperature, turbulent kinetic energy, and dissipation rate, each of these quantities is defined in the centre of a grid cell. In most atmospheric flow field models the problem of decoupling the momentum and pressure-correction equations is overcome by using staggered grids, e.g. the Arakawa-C grid (Arakawa and Lamb, 1977). In the GRAMM model a different method as suggested by Almbauer (1995) has been implemented, hereafter referred to as Almbauer grid. A secondary grid is established by dividing grid cells diagonally in two parts along the space which generates two tetrahedrons (Fig. 5). While the non-hydrostatic pressure is defined at the surfaces of every tetrahedron, the velocity components are defined in the centre of the tetrahedrons. In contrast to the widely used Arakawa-C grid, there is the advantage that all velocity components are defined at the very same point in space. It will be shown later that it is not necessary to know the exact shape of the area separating both tetrahedrons to discretize the governing equations. By definition both tetrahedrons are equal in volume.

Fig. 4. Primary grid cell used for discretizing the conservation equations of specific humidity, potential temperature, turbulent kinetic energy, and dissipation rate

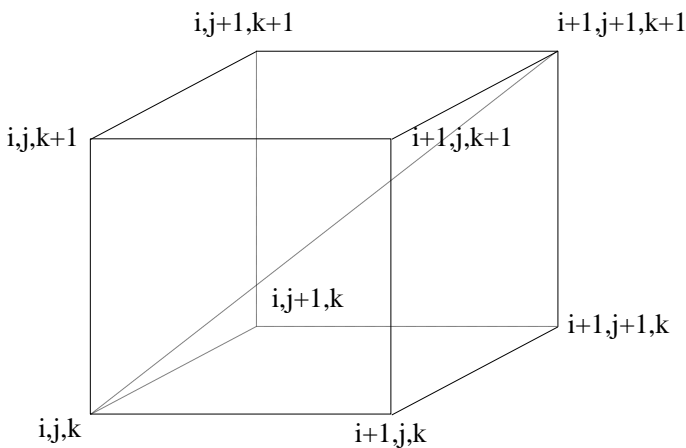
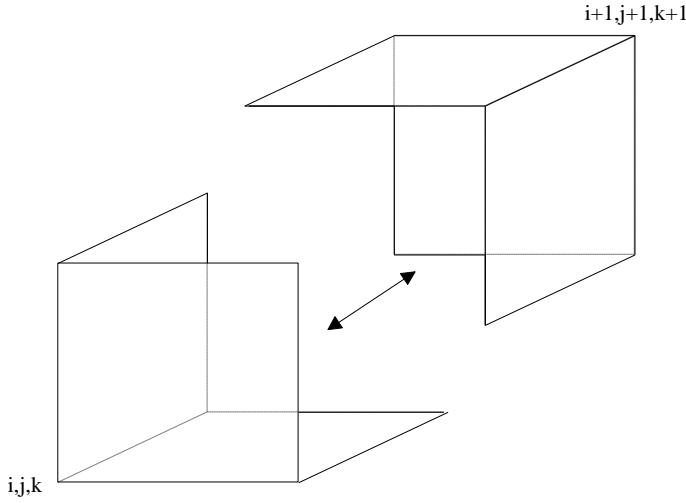


Fig. 5. Secondary grid used for discretizing the conservation equations of momentum and the pressure-correction (non-hydrostatic pressure equation)



Below the conservation equations for a quantity ψ consisting of a time-dependent term, an advection and diffusion term, and source- and sink terms, is written in its general form:

$$\frac{\partial(\bar{\rho}\phi)}{\partial t} + \frac{\partial(\bar{\rho}u_j\phi)}{\partial x_j} = \bar{\rho} \frac{\partial}{\partial x_j} \left(\mu_\phi \frac{\partial \bar{\phi}}{\partial x_j} \right) + S_\phi \quad (\text{E } 74)$$

As previously explained, the non-hydrostatic pressure field established to correct the velocity field to meet mass-conservation is defined at the surfaces of the tetrahedrons. Therefore, mass balance is calculated for every surface. To do so, volume integrals need to be transformed applying the Gauss's divergence (Gauss, 1813) theorem into surface integrals. V is the considered closed volume, δO the surface enclosing V , and n a vector of unit length perpendicular to δO :

$$\iiint_V \frac{\partial \vec{u}(x_1, x_2, x_3)}{\partial x_1} dx_1 dx_2 dx_3 = \oiint_{\delta O} \vec{u} \cos(n, x_1) dO \quad (\text{E } 75)$$

$\cos(n, x_1)$ is the cosine of the angle between the n and x_1 -axis. Using $dV = dx_1 dx_2 dx_3$ gives

$$\iiint_V \left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3} \right) dV = \oiint_{\delta O} [u_1 \cos(n, x_1) + u_2 \cos(n, x_2) + u_3 \cos(n, x_3)] dO \quad (\text{E } 76)$$

It is easy to see that $\cos(n, x_i) dO$ expresses the projection of each surface in the x_i - planes. By using

Detailed model description - Numerical schemes

$$\cos(n, x_1) dO = A_{x_2 x_3} = dx_2 dx_3 \quad (\text{E } 77)$$

the divergence theorem can be rearranged into:

$$\iiint_V \left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3} \right) dV = \oiint_{\partial O} [u_1 dx_2 dx_3 + u_2 dx_3 dx_1 + u_3 dx_1 dx_2] \quad (\text{E } 78)$$

Assuming constant values for u_1 , u_2 and u_3 for each surface gives

$$\oiint_{\partial O} [u_1 dx_2 dx_3 + u_2 dx_3 dx_1 + u_3 dx_1 dx_2] = \sum_{i=1}^4 (u_1 A_{x_2 x_3} + u_2 A_{x_3 x_1} + u_3 A_{x_1 x_2})_i. \quad (\text{E } 79)$$

The discretized equation is carried out for all four surfaces of each tetrahedron, which is indicated by the subscript i .

In a similar manner, the equations for the pressure gradients can be obtained. One disadvantage of the Almbauer grid is the number of discretization points for the non-hydrostatic pressure, which is four times the one of the Arakawa-C grid.

Another topic that needs to be addressed is sort of an anisotropic effect introduced by the grid: In case of a plane surface one would expect the flow field to behave equally regardless the direction of the flow. A closer look on the discretisation of the diffusion terms in the momentum equations reveals that this view is violated when these are computed based on the tetrahedrons for two reasons: in a horizontally homogenous flow velocity gradients appear only in the vertical direction due to surface friction. A logarithmic wind-profile develops in the surface layer, which means in case of the specific grid used in the GRAMM model that in case of a westerly flow higher wind speeds in the top tetrahedron feed in the bottom tetrahedron of the neighbouring grid cell. In case of easterly flows the situation is vice versa: lower wind speeds calculated for the bottom tetrahedron feed in the top tetrahedron of the neighbouring cell (Fig. 6). The second problem that arises is that the mass balance at the interface between the two tetrahedrons is not fulfilled, which leads to the development of unrealistic vertical winds (Fig. 7). The whole problem can easily be overcome by expressing the vertical diffusion terms (spatial velocity gradients) in the momentum equations based on the primary grid (hexahedrons) instead of the tetrahedrons. In this way wind speeds will be the same in both tetrahedrons.

Fig. 6. Asymmetric behaviour of horizontally homogenous flows due to the Almbauer grid

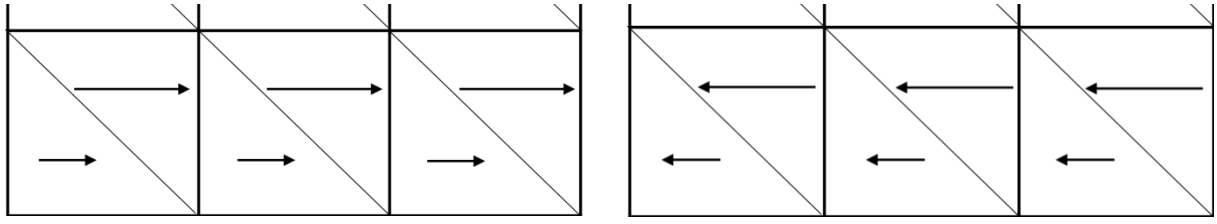
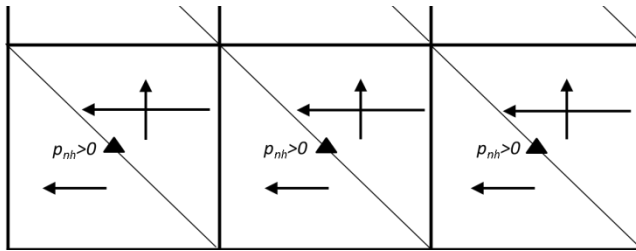


Fig. 7. Unrealistic vertical wind generation in horizontally homogenous flows induced by the Almbauer grid



Below the derivation of the momentum equation for the u -component is outlined in more detail. The basic equation is

$$\frac{\partial(\bar{\rho}\bar{u}_1)}{\partial t} + \frac{\partial(\bar{\rho}\bar{u}_1\bar{u}_j)}{\partial x_j} = -\bar{\rho}f_3(\bar{u}_{2g} - \bar{u}_2) - \bar{\rho}f_2\bar{u}_3 - \bar{\rho}\frac{\partial}{\partial x_j}\left(\mu_t\left(\frac{\partial\bar{u}_j}{\partial x_1} + \frac{\partial\bar{u}_1}{\partial x_j}\right) - \frac{2}{3}k\delta_{1j}\right) \quad (\text{E } 80)$$

As already explained the term $\frac{2}{3}k\delta_{1j}$ is not computed explicitly. In addition, the term $\bar{\rho}f_2\bar{u}_3$ is neglected. The vertical gradients in the diffusion terms are computed based on the primary grid. The u -components of the primary grid are termed U and those of the secondary grid U_1 and U_2 (see Fig. 8). Applying the divergence theorem for the diffusion terms in (E 80) gives:

$$\iiint_V \bar{\rho} \frac{\partial}{\partial x_j} \left(\mu_t \left(\frac{\partial\bar{u}_j}{\partial x_1} + \frac{\partial\bar{u}_1}{\partial x_j} \right) \right) dV = \iint_O \bar{\rho} \mu_t \left(\frac{\partial\bar{u}_j}{\partial x_1} + \frac{\partial\bar{u}_1}{\partial x_j} \right) dO \quad (\text{E } 81)$$

The vertical gradient $\frac{\partial\bar{u}_1}{\partial x_3}$ based on the primary grid is therefore obtained by:

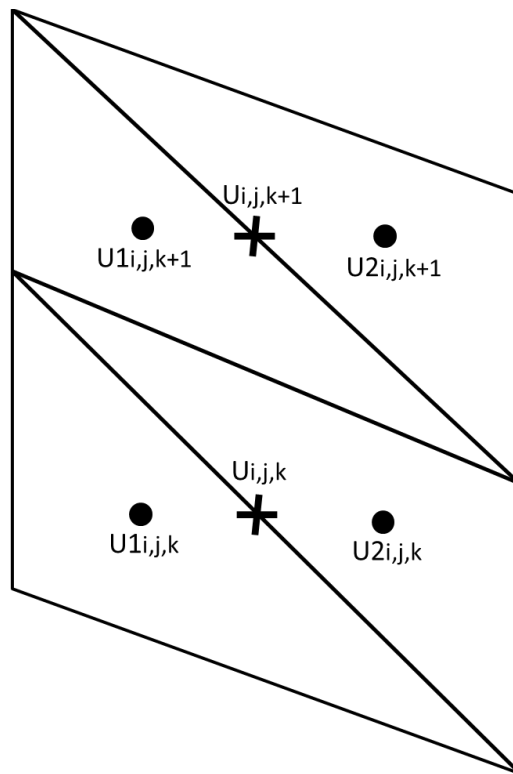
Detailed model description - Numerical schemes

$$\iint_O \bar{\rho} \mu_t \frac{\partial \bar{u}_1}{\partial x_3} dO = 0.5(\bar{\rho}_{ijk+1} + \bar{\rho}_{ijk}) 0.5(\mu_{tijk+1} + \mu_{tijk}) \frac{U_{ijk+1} - U_{ijk}}{Z_{ijk+1} - Z_{ijk}} A_{ZZ_{ijk+1}} -$$

$$0.5(\bar{\rho}_{ijk-1} + \bar{\rho}_{ijk}) 0.5(\mu_{tijk-1} + \mu_{tijk}) \frac{U_{ijk} - U_{ijk-1}}{Z_{ijk} - Z_{ijk-1}} A_{ZZ_{ijk}} \quad (E 82)$$

A_{ZZ} is the projection of AZ in z -direction (see Fig. 10), and Z is the height of the cell-centre of the primary grid. Expression (E 128) is used for both tetrahedrons of the secondary grid.

Fig. 8. Discretization points for the wind speed components (x - z plane) in the primary and secondary grid



All other spatial gradients of the diffusion terms are computed using the secondary grid. Below, the discretization equations are illustrated for the lower tetrahedron. While density is computed as the arithmetic mean between adjacent tetrahedrons, turbulent viscosity is computed as harmonic mean as suggested in Patankar (1980). According to the arguments brought forward by Patankar (1980) the arithmetic mean is appropriate as long as the spatial gradients are low, while the harmonic mean gives a better representation of fluxes in the other case. Density changes in the horizontal directions are relatively low, that's why the arithmetic mean is used here, while larger spatial gradients can occur for the turbulent viscosity.

$$\begin{aligned}
\iint_O \bar{\rho} \mu_t \frac{\partial \bar{u}_1}{\partial x_1} dO &= 2 \cdot \bar{\rho}_{ijk} \cdot \mu_{tijk} \frac{U2_{ijk} - U1_{ijk}}{0.5DX} (AX_{ijk} + AZX_{ijk}) - \\
2 \cdot 0.5(\bar{\rho}_{i-1jk} + \bar{\rho}_{ijk}) \cdot 2 \cdot \frac{\mu_{ti-1jk} \mu_{tijk}}{\mu_{ti-1jk} + \mu_{tijk}} \frac{U1_{ijk} - U2_{i-1jk}}{X_{ijk} - X_{i-1jk}} AX_{ijk} - \\
2 \cdot 0.5(\bar{\rho}_{ijk-1} + \bar{\rho}_{ijk}) \cdot 2 \cdot \frac{\mu_{tijk-1} \mu_{tijk}}{\mu_{tijk-1} + \mu_{tijk}} \frac{0.5[(U2_{ijk} - U1_{ijk}) + (U2_{ijk-1} - U1_{ijk-1})]}{0.5DX} AZX_{ijk}
\end{aligned} \tag{E 83}$$

In (E 83) DX is the horizontal grid spacing of the primary grid in x-direction. For the definition of AZX see Fig. 10. A similar expression can be derived for the turbulent fluxes in y-direction:

$$\begin{aligned}
\iint_O \bar{\rho} \mu_t \frac{\partial \bar{u}_1}{\partial x_2} dO &= \bar{\rho}_{ijk} \cdot \mu_{tijk} \frac{U2_{ijk} - U1_{ijk}}{0.5DY} (AY_{ijk} + AZY_{ijk}) - \\
0.5(\bar{\rho}_{ij-1k} + \bar{\rho}_{ijk}) \cdot 2 \cdot \frac{\mu_{tij-1k} \mu_{tijk}}{\mu_{tij-1k} + \mu_{tijk}} \frac{U1_{ijk} - U2_{ij-1k}}{Y_{ijk} - Y_{ij-1k}} AY_{ijk} - \\
0.5(\bar{\rho}_{ijk-1} + \bar{\rho}_{ijk}) \cdot 2 \cdot \frac{\mu_{tijk-1} \mu_{tijk}}{\mu_{tijk-1} + \mu_{tijk}} \frac{0.5[(U2_{ijk} - U1_{ijk}) + (U2_{ijk-1} - U1_{ijk-1})]}{0.5DY} AZY_{ijk}
\end{aligned} \tag{E 84}$$

$$\begin{aligned}
\iint_O \bar{\rho} \mu_t \frac{\partial \bar{u}_2}{\partial x_1} dO &= \\
\bar{\rho}_{ijk} \cdot \mu_{tijk} \frac{V2_{ijk} - V1_{ijk}}{0.5DX} (AY_{ijk} + AZY_{ijk}) - \\
0.5(\bar{\rho}_{ij-1k} + \bar{\rho}_{ijk}) \cdot 2 \cdot \frac{\mu_{tij-1k} \mu_{tijk}}{\mu_{tij-1k} + \mu_{tijk}} \cdot \frac{0.5[(V2_{ijk} - V1_{ijk}) + (V2_{ij-1k} - V1_{ij-1k})]}{0.5DX} AY_{ijk} - \\
0.5(\bar{\rho}_{ijk-1} + \bar{\rho}_{ijk}) \cdot 2 \cdot \frac{\mu_{tijk-1} \mu_{tijk}}{\mu_{tijk-1} + \mu_{tijk}} \frac{0.5[(V2_{ijk} - V1_{ijk}) + (V2_{ijk-1} - V1_{ijk-1})]}{0.5DX} AZY_{ijk}
\end{aligned} \tag{E 85}$$

And finally the turbulent fluxes in z-direction are:

$$\begin{aligned}
\iint_O \bar{\rho} \mu_t \frac{\partial \bar{u}_3}{\partial x_1} dO &= \\
\bar{\rho}_{ijk} \cdot \mu_{tijk} \frac{W2_{ijk} - W1_{ijk}}{0.5DX} AZZ_{ijk} - \\
0.5(\bar{\rho}_{ijk-1} + \bar{\rho}_{ijk}) \cdot 2 \cdot \frac{\mu_{tijk-1} \mu_{tijk}}{\mu_{tijk-1} + \mu_{tijk}} \frac{0.5[(W2_{ijk} - W1_{ijk}) + (W2_{ijk-1} - W1_{ijk-1})]}{0.5DX} AZZ_{ijk}
\end{aligned} \tag{E 86}$$

In a similar manner diffusive fluxes for the upper tetrahedron as well as for the other wind components can be derived.

Detailed model description - Numerical schemes

The advection terms in the u -momentum equation are equal to:

$$\iiint_V \frac{\partial(\overline{\rho u_1 u_j})}{\partial x_j} dV = \iint_O \overline{\rho u_1 u_j} dO \quad (\text{E } 87)$$

The advective mass fluxes $F_j = \iint_O \overline{\rho u_j} dO$ at the surfaces of the tetrahedron are obtained by:

$$\begin{aligned} F_b = & \left[0.5(\overline{\rho}_{ijk} W1_{ijk} + \overline{\rho}_{ijk} W2_{ijk}) AZZ_{ijk} \right] \\ & + \left[0.5(\overline{\rho}_{ijk} U1_{ijk} + \overline{\rho}_{ijk} U2_{ijk}) (AZX_{ijk} + AX_{ijk}) \right] \\ & + \left[0.5(\overline{\rho}_{ijk} V1_{ijk} + \overline{\rho}_{ijk} V2_{ijk}) (AZY_{ijk} + AY_{ijk}) \right] \end{aligned} \quad (\text{E } 88)$$

$$\begin{aligned} F_e = & \left[0.5(\overline{\rho}_{ijk} W1_{ijk} + \overline{\rho}_{ijk-1} W2_{ijk-1}) AZZ_{ijk} \right] \\ & + \left[0.5(\overline{\rho}_{ijk} U1_{ijk} + \overline{\rho}_{ijk-1} U2_{ijk-1}) AZX_{ijk} \right] \\ & + \left[0.5(\overline{\rho}_{ijk} V1_{ijk} + \overline{\rho}_{ijk-1} V2_{ijk-1}) AZY_{ijk} \right] \end{aligned} \quad (\text{E } 89)$$

$$F_s = \left[0.5(\overline{\rho}_{ijk} V1_{ijk} + \overline{\rho}_{ij-1k} V2_{ij-1k}) AY_{ijk} \right] \quad (\text{E } 90)$$

$$F_w = \left[0.5(\overline{\rho}_{ijk} U1_{ijk} + \overline{\rho}_{i-1jk} U2_{i-1jk}) AX_{ijk} \right] \quad (\text{E } 91)$$

The indices b (=bottom), e (=east), s (=south), and w (=west) denote the corresponding surfaces over which the mass fluxes occur. There exist numerous ways on how to define U at the surfaces. Below, a brief derivation of a discretization of the general form of the advection-diffusion equation in one dimension is outlined as proposed by Patankar (1980). It provides the basis to understanding of the underlying advection scheme used in the GRAMM model. The detailed derivation, though, is beyond the scope of this work. The reader is referred to chapter 5.2.7 of Patankar (1980).

Basically the total flux (advection and diffusion) between adjacent grid cells can be written in a general form:

$$J^* = B\phi_i - A\phi_{i+1}, \quad (\text{E } 92)$$

where A and B are arbitrary dimensionless coefficients that are functions of the Peclet number $P = \frac{\rho u dx}{\mu_t}$. It has to be mentioned that J^* is the total flux multiplied by the distance

between two grid points dx and divided by the turbulent viscosity $\frac{1}{D} = \frac{dx}{\mu_t}$. The relationship

$$B = A + P \quad (E 93)$$

results when $\phi_i = \phi_{i+1}$. Further symmetry arguments lead to:

$$A(P) = A(|P|) + \text{Max}(-P, 0) \quad (E 94)$$

$$B(P) = A(|P|) + \text{Max}(P, 0) \quad (E 95)$$

In the one-dimensional case neglecting source terms the momentum equation becomes:

$$J_e - J_w = 0 \quad (E 96)$$

J is the mass flux rate over a surface due to advection and diffusion. Multiplying (E 92) with

$D = \frac{\mu_t}{dx}$ and $F = \rho u$, and inserting in (E 96) gives:

$$[B\phi_P - A\phi_E]D_e - [B\phi_W - A\phi_P]D_w = 0 \quad (E 97)$$

Using (E 94) and (E 95) for A and B leads to:

$$\begin{aligned} &\phi_P(D_e A + \text{Max}(F_e, 0)) - \phi_E(D_e A + \text{Max}(-F_e, 0)) - \\ &\phi_W(D_w A + \text{Max}(F_w, 0)) + \phi_P(D_w A + \text{Max}(-F_w, 0)) = 0 \end{aligned} \quad (E 98)$$

Now we define:

$$a_E = D_e A(|P|) + \text{Max}(-F_e, 0) \quad (E 99)$$

$$a_W = D_w A(|P|) + \text{Max}(F_w, 0) \quad (E 100)$$

$$a_P = a_E + a_W + (F_e - F_w) \quad (E 101)$$

The expression for a_P follows from the fact that:

Detailed model description - Numerical schemes

$$\begin{aligned} \text{Max}(F_e, 0) &= \text{Max}(-F_e, 0) + F_e \\ \text{Max}(-F_w, 0) &= \text{Max}(F_w, 0) - F_w \end{aligned} \quad (\text{E } 102)$$

Thus, in one dimension the advection-diffusion equation becomes:

$$a_P \phi_P = a_E \phi_E + a_W \phi_W \quad (\text{E } 103)$$

Patankar (1980) formulates some basic rules that are necessary – for the way in which the momentum equations are discretized here – to ensure physical realism and overall balance:

- 1) Consistency of fluxes at faces common to two adjacent control volumes.
- 2) Positive coefficients: For pure convection-diffusion problems an increase at one point should lead to an increase in the value at the neighbouring grid point. Therefore the coefficients in (E 103) must all have the same sign (here positive).
- 3) Sum of the neighbour coefficients: In convection-diffusion equations only derivatives of a variable ϕ occur. That means that the functions ϕ and $\phi + c$ (where c is an arbitrary constant) both satisfy the differential equation. In order to maintain this property in the discretization equation, it is necessary that the coefficient a_P equals the sum of the neighbour coefficients as in (E 103).

The term $(F_e - F_w)$ in (E 101) is the mass divergence of a grid cell, which is zero as soon as the continuity equation is fulfilled. However, in the course of a simulation continuity is often not fulfilled perfectly. Therefore, the aforementioned rule 3) may not be satisfied. Multiplying the continuity equation (E 13) with $-\phi$ and adding it to the advection-diffusion equation diminishes the term $(F_e - F_w)$ so that rule 3) is satisfied in all circumstances.

In three-dimensions the momentum equation can be expressed for the lower tetrahedron for the quantity $U1$ as follows:

$$a_P U1_{ijk} = a_E U2_{ijk} + a_S U2_{ij-1k} + a_W U2_{i-1jk} + a_B U2_{ijk-1} + b, \quad (\text{E } 104)$$

$$a_P = a_E + a_S + a_W + a_B + a_P^0, \quad (\text{E } 105)$$

The term a_P^0 in (E 105) is given by:

$$a_P^0 = \frac{\bar{\rho}_{ijk} V_{ijk}}{\Delta t} \quad (\text{E } 106)$$

The upper index 0 stands for values defined at the beginning of a time step, while no index denotes values at the end of the time step. Thus, in GRAMM a full-implicit time integration scheme is used, which has the advantage that in general time steps can be larger than the limit imposed by the Courant-Friedrichs-Lewy condition on explicit methods. The disadvantage is the necessity to solve the equation iteratively to obtain the new velocities because of their dependencies on each other.

In GRAMM the so-called “power-law”-advection scheme is utilized (Patankar, 1980). That means function A in (E 94) is dependent on the Peclet number $\frac{F_w}{D_w}$ in the following way (here outlined for the west face of the lower tetrahedron):

$$A(|P|) = \text{Max} \left[0, 0.1 \frac{|F_w|^5}{D_w} \right] \quad (\text{E } 107)$$

$$a_w = D_w \text{Max} \left[0, 0.1 \frac{|F_w|^5}{D_w} \right] + \text{Max}[0, F_w] \quad (\text{E } 108)$$

$$F_w = \left[0.5 (\bar{\rho}_{ijk} U1_{ijk} + \bar{\rho}_{i-1jk} U2_{i-1jk}) A X_{ijk} \right] \quad (\text{E } 109)$$

$$D_w = 2 \cdot 0.5 (\bar{\rho}_{i-1jk} + \bar{\rho}_{ijk}) \cdot 2 \frac{\bar{\mu}_{t\ i-1jk} \bar{\mu}_{t\ ijk}}{\bar{\mu}_{t\ i-1jk} + \bar{\mu}_{t\ ijk}} \frac{1}{X_{ijk} - X_{i-1jk}} A X_{ijk} \quad (\text{E } 110)$$

The term b in (E 104) in general stands for all source terms in the momentum equations, such as the Coriolis force or the buoyancy force in the vertical. In addition, all terms not containing any of the neighbour velocities in x-direction as given in (E 104), such as e.g. the right-hand terms of (E 85) and (E 86), are summarized.

Numerical stability is enhanced by underrelaxation in the form:

$$U1 = U1^{n-1} + \alpha(U1 - U1^{n-1}) \quad (\text{E } 111)$$

For $\alpha = 1$ there is no underrelaxation, while for any value of α lower than 1 only a part of the new solution is taken into account. Typically a value between 0.1 – and 0.2 is used depending on the horizontal grid sizes. The upper index n denotes the value of the previous iteration. It should not be mixed up with the upper index 0 indicating values of the previous time step.

Equation (E 104) is solved iteratively using the so-called Thomas algorithm (or Tri-diagonal-matrix algorithm – TDMA). The TDMA provides correct solutions for every column, but as the solution of a column depends on its neighbours, the final solution can only be obtained by an iterative procedure. In the GRAMM model, whenever the TDMA is applied, vertical columns are utilized, because spatial gradients in most cases are more pronounced in the vertical rather than in the horizontal directions. In this way, numerical inaccuracies can be minimized. To do so, the momentum equation for the U wind component for the lower tetrahedron is arranged in the form

$$a_P U1_{ijk} = a_E U2_{ijk} + a_B U2_{ijk-1} + d \quad (\text{E } 112)$$

To enable the computation of U in a vertical column, hereafter including the upper tetrahedron in the derivation, it is necessary to have equations of the form such as

$$U1_{ijk} = P_k U2_{ijk} + Q_k \quad (\text{E } 113)$$

$$U2_{ijk} = P_k U1_{ijk+1} + Q_k \quad (\text{E } 114)$$

Inserting (E 114) in (E 112) gives:

$$a_P U1_{ijk} = a_B (P_{k-1} U1_{ijk} + Q_{k-1}) + a_E U2_{ijk} + d \quad (\text{E } 115)$$

$$U1_{ijk} (a_P - a_B P_{k-1}) = a_B Q_{k-1} + a_E U2_{ijk} + d \quad (\text{E } 116)$$

$$U1_{ijk} = \frac{a_E}{(a_P - a_B P_{k-1})} U2_{ijk} + \frac{d + a_B Q_{k-1}}{(a_P - a_B P_{k-1})} \quad (\text{E } 117)$$

Comparing (E 117) with (E 113) leads to

$$\begin{aligned}
P_k &= \frac{a_E}{(a_P - a_B P_{k-1})} \\
Q_k &= \frac{d + a_B Q_{k-1}}{(a_P - a_B P_{k-1})}
\end{aligned}
\tag{E 118}$$

First, P and Q are computed for vertical columns with increasing cell index k , where P_0 and Q_0 are zero (i.e. the ground is physically not linked with the air flow above). Second, the new velocities in the iteration step are obtained by using (E 113) and (E 114). Here the vertical cell index k is decreasing in the solution algorithm. The number of iterations in each time step is fixed to eight, and the propagation directions are changed in the same way as outlined in chapter 4.7.4 (see Fig. 11).

The Almbauer grid offers the advantage that all three velocity components are defined in the same grid, thus, the terms a_x are identical and only need to be computed once. Only the source term b needs to be calculated separately for each wind component.

In GRAMM the time step is not set constant for the entire simulation time, instead it is adjusted according to the rate of mass convergence. Mass balance is computed for each grid cell after each time step and summed up over all cells. After regular intervals (i.e. 10 time steps) the rate of increase or decrease in mass divergence is estimated by calculating the slope of a linear best fit using the least-square method. As soon as the total mass divergence is decreasing or if it remains constant, the time step is increased by 10 % until the upper threshold for the time step is reached, which is a user-defined value. Typically, such limits are of the order of 10 s and depend upon the horizontal grid size.

4.7.3 Discretization of the conservation equations for pot. Temperature, humidity, turbulent kinetic energy, and dissipation rate

As already stated the Almbauer grid consists of a primary grid (hexahedrons) and a secondary grid (tetrahedrons). Potential temperature, specific humidity, turbulent kinetic energy, and dissipation rate are located in the centre of the primary grid. Hereafter, these quantities are referred to as scalar ψ .

We start again with the general form of the advection-diffusion equation:

$$\frac{\partial(\bar{\rho}\phi)}{\partial t} + \frac{\partial(\bar{\rho}u_j\bar{\phi})}{\partial x_j} = \bar{\rho} \frac{\partial}{\partial x_j} \left(\mu_\phi \frac{\partial \bar{\phi}}{\partial x_j} \right) + S_\phi
\tag{E 119}$$

Integration over the volume and applying the divergence theorem for the diffusion terms in (E 119) gives:

$$\iiint_V \bar{\rho} \frac{\partial}{\partial x_j} \left(\mu_\phi \frac{\partial \bar{\phi}}{\partial x_j} \right) dV = \iint_O \bar{\rho} \mu_\phi \frac{\partial \bar{\phi}}{\partial x_j} dO \quad (\text{E } 120)$$

The diffusive flux rates based on the primary grid are therefore obtained by:

$$\begin{aligned} \iint_O \bar{\rho} \mu_\phi \frac{\partial \bar{\phi}}{\partial x_j} dO = & \\ & 0.5(\bar{\rho}_{i+1,jk} + \bar{\rho}_{ijk}) 0.5(\mu_{\phi_{i+1,jk}} + \mu_{\phi_{ijk}}) (\bar{\phi}_{i+1,jk} - \bar{\phi}_{ijk}) \left(\frac{AX_{i+1,jk}}{DX} \right) - \\ & 0.5(\bar{\rho}_{i-1,jk} + \bar{\rho}_{ijk}) 0.5(\mu_{\phi_{i-1,jk}} + \mu_{\phi_{ijk}}) (\bar{\phi}_{ijk} - \bar{\phi}_{i-1,jk}) \left(\frac{AX_{ijk}}{DX} \right) + \\ & 0.5(\bar{\rho}_{ij+1k} + \bar{\rho}_{ijk}) 0.5(\mu_{\phi_{ij+1k}} + \mu_{\phi_{ijk}}) (\bar{\phi}_{ij+1k} - \bar{\phi}_{ijk}) \left(\frac{AY_{ij+1k}}{DY} \right) - \\ & 0.5(\bar{\rho}_{ij-1k} + \bar{\rho}_{ijk}) 0.5(\mu_{\phi_{ij-1k}} + \mu_{\phi_{ijk}}) (\bar{\phi}_{ijk} - \bar{\phi}_{ij-1k}) \left(\frac{AY_{ijk}}{DY} \right) + \\ & 0.5(\bar{\rho}_{ijk+1} + \bar{\rho}_{ijk}) 0.5(\mu_{\phi_{ijk+1}} + \mu_{\phi_{ijk}}) (\bar{\phi}_{ijk+1} - \bar{\phi}_{ijk}) \left(\frac{AZZ_{ijk+1}}{Z_{ijk+1} - Z_{ijk}} + \frac{AZX_{ijk+1}}{DX} + \frac{AZY_{ijk+1}}{DY} \right) - \\ & 0.5(\bar{\rho}_{ijk-1} + \bar{\rho}_{ijk}) 0.5(\mu_{\phi_{ijk-1}} + \mu_{\phi_{ijk}}) (\bar{\phi}_{ijk} - \bar{\phi}_{ijk-1}) \left(\frac{AZZ_{ijk}}{Z_{ijk} - Z_{ijk-1}} + \frac{AZX_{ijk}}{DX} + \frac{AZY_{ijk}}{DY} \right) \end{aligned} \quad (\text{E } 121)$$

The advective mass fluxes $F_j = \iint_O \bar{\rho} \bar{u}_j dO$ at the surfaces of the tetrahedron are obtained by:

$$\begin{aligned}
F_t = & 0.5(\bar{\rho}_{ijk} + \bar{\rho}_{ijk+1}) \left[0.5(w_{ijk} + w_{ijk+1}) AZZ_{ijk+1} + \right. \\
& + 0.5(\bar{\rho}_{ijk} + \bar{\rho}_{ijk+1}) \left[0.5(u_{ijk} + u_{ijk+1}) AZX_{ijk+1} \right] \\
& + 0.5(\bar{\rho}_{ijk} + \bar{\rho}_{ijk+1}) \left[0.5(v_{ijk} + v_{ijk+1}) AZY_{ijk+1} \right]
\end{aligned} \tag{E 122}$$

$$\begin{aligned}
F_b = & 0.5(\bar{\rho}_{ijk} + \bar{\rho}_{ijk-1}) \left[0.5(w_{ijk} + w_{ijk-1}) AZZ_{ijk} + \right. \\
& + 0.5(\bar{\rho}_{ijk} + \bar{\rho}_{ijk-1}) \left[0.5(u_{ijk} + u_{ijk-1}) AZX_{ijk} \right] \\
& + 0.5(\bar{\rho}_{ijk} + \bar{\rho}_{ijk-1}) \left[0.5(v_{ijk} + v_{ijk-1}) AZY_{ijk} \right]
\end{aligned} \tag{E 123}$$

$$F_e = 0.5(\bar{\rho}_{ijk} + \bar{\rho}_{i+1,jk}) \left[0.5(u_{ijk} + u_{i+1,jk}) AX_{i+1,jk} \right] \tag{E 124}$$

$$F_w = 0.5(\bar{\rho}_{ijk} + \bar{\rho}_{i-1,jk}) \left[0.5(u_{ijk} + u_{i-1,jk}) AX_{ijk} \right] \tag{E 125}$$

$$F_s = 0.5(\bar{\rho}_{ijk} + \bar{\rho}_{ij-1k}) \left[0.5(v_{ijk} + v_{ij-1k}) AY_{ijk} \right] \tag{E 126}$$

$$F_n = 0.5(\bar{\rho}_{ijk} + \bar{\rho}_{ij+1k}) \left[0.5(v_{ijk} + v_{ij+1k}) AY_{ij+1k} \right] \tag{E 127}$$

The advection scheme, time integration method, and iterative solution of the discretized advection-diffusion equation for scalars are the same as for the momentum equations.

4.7.4 Discretization of the non-hydrostatic pressure equation

After each time step mass conservation is fulfilled by correcting the velocity field by the non-hydrostatic pressure field using:

$$\frac{\partial^2 p^*}{\partial x_i^2} = \frac{\partial}{\partial t} \left(\frac{\partial \rho_0 \bar{u}_i}{\partial x_i} \right). \tag{E 128}$$

Integrating (E 128) over an arbitrary control volume and applying the divergence theorem gives:

$$\iiint_V \left[\frac{\partial^2 p^*}{\partial x_i^2} - \frac{\partial}{\partial t} \left(\frac{\partial \rho_0 \bar{u}_i}{\partial x_i} \right) \right] dV = 0 = \iint_O \left[\frac{\partial p^*}{\partial x_i} - \frac{\partial}{\partial t} (\rho_0 \bar{u}_i) \right] dO \tag{E 129}$$

Detailed model description - Numerical schemes

The right-hand side of (E 129) can be interpreted in a way that the non-hydrostatic pressure gradients at the surfaces of the control volume generate a mass-flow through a surface such that mass-balance is finally achieved. In the GRAMM model mass balance is controlled at every interface of the tetrahedrons, while e.g. when using the Arakawa-C grid, mass balance is controlled over every grid cell. In this case, non-hydrostatic pressure is defined in the centre of every grid cell, thus, the spatial pressure gradients are equal on both sides of a surface. In the Almbauer grid, non-hydrostatic pressure is defined at the surfaces, which leads to the circumstance that the pressure gradients are different on both sides and have to be computed separately. Therefore, the term $\frac{\partial p^*}{\partial x_i}$ is computed as a volume-average for each tetrahedron on both sides of the surface of interest:

$$\frac{1}{V} \iiint_V \frac{\partial p^*}{\partial x_i} dV = \frac{1}{V} \iint_O p^* dO, \quad (\text{E } 130)$$

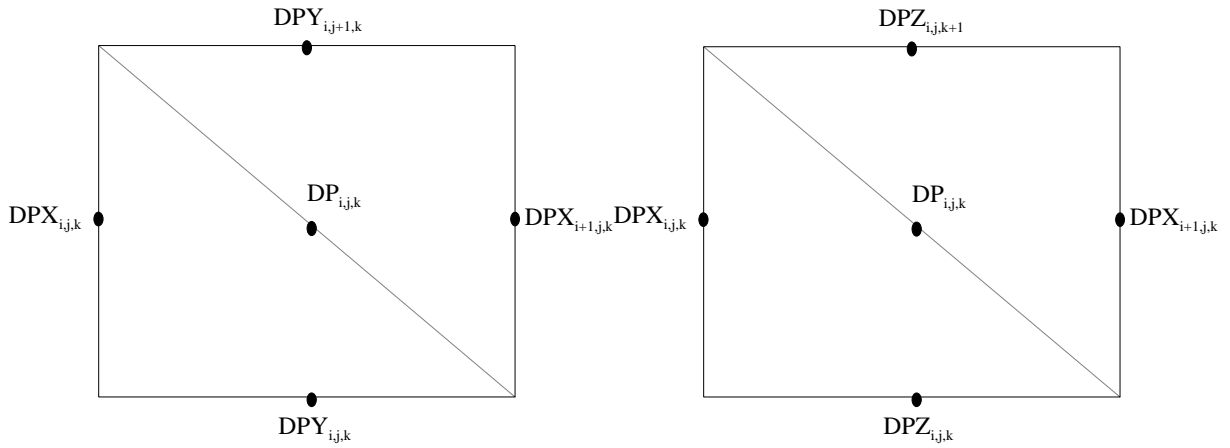
Note that V in (E 130) is the volume of a tetrahedron. Once again the divergence theorem is used to obtain the right-hand side. Applying the right-hand side of (E 129), which was derived for a control volume, in a similar way for an arbitrary surface separating two tetrahedrons and using (E 130) gives

$$A_i \cdot \left(\bar{u}_i^* \rho_0 + \left[\frac{1}{V} \iint_O p^* dO \right] \Delta t \right)_a - A_i \cdot \left(\bar{u}_i^* \rho_0 + \left[\frac{1}{V} \iint_O p^* dO \right] \Delta t \right)_b \stackrel{!}{=} 0. \quad (\text{E } 131)$$

u_i^* are the velocities obtained after solving the conservation equations for momentum, and the subscripts a and b indicate two neighbouring tetrahedrons. A_i are the projections of the surface of interest in x -, y -, and z -directions. (E 131) describes the case in which the total mass-flux over an entire arbitrary surface is equal on both sides.

The equation is solved iteratively using the TDMA. As already mentioned, non-hydrostatic pressure is constant at the interfaces of the tetrahedrons and is located at the centre of each surface as depicted in Fig. 9.

Fig. 9. Discrete locations for the non-hydrostatic pressure (left: x-y plane; right: x-z plane)



As already mentioned, the TDMA provides correct solutions for every column, but as the solution of a column depends on its neighbours, the final solution can only be obtained by an iterative procedure. To do so, the Poisson equation for the non-hydrostatic pressure is arranged in the form

$$a_k DPZ_{i,j,k} = b_k DP_{i,j,k} + c_k DP_{i,j,k-1} + d_k. \quad (\text{E } 132)$$

To enable the computation of the pressures in a vertical column, it is necessary to have a form like

$$DPZ_{i,j,k} = P_k DP_{i,j,k-1} + Q_k \quad (\text{E } 133)$$

$$DP_{i,j,k} = P_{k+1} DPZ_{i,j,k} + Q_{k+1}. \quad (\text{E } 134)$$

Inserting (E 134) in (E 132) gives

$$a_k DPZ_{i,j,k} = b_k (P_{k+1} DPZ_{i,j,k} + Q_{k+1}) + c_k DP_{i,j,k-1} + d_k. \quad (\text{E } 135)$$

$$DPZ_{i,j,k} (a_k - b_k P_{k+1}) = b_k Q_{k+1} + c_k DP_{i,j,k-1} + d_k. \quad (\text{E } 136)$$

$$DPZ_{i,j,k} = \frac{c_k}{(a_k - b_k P_{k+1})} DP_{i,j,k-1} + \frac{d_k + b_k Q_{k+1}}{(a_k - b_k P_{k+1})}. \quad (\text{E } 137)$$

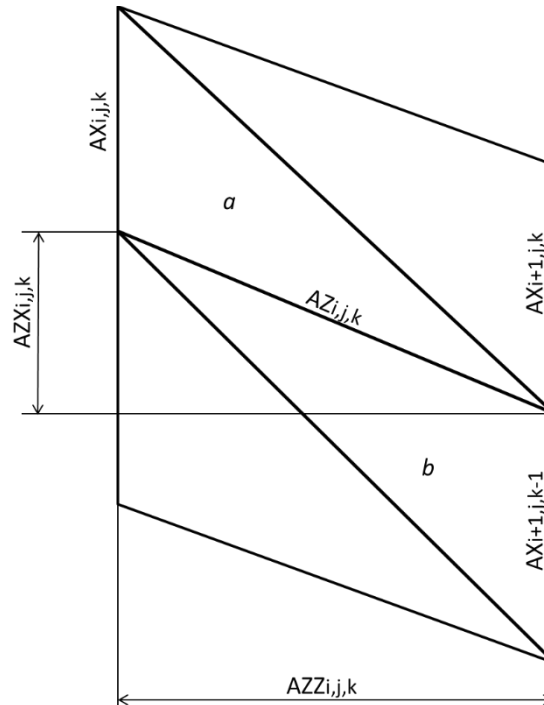
Comparing (E 137) with (E 133) leads to

$$\begin{aligned} P_k &= \frac{c_k}{(a_k - b_k P_{k+1})} \\ Q_k &= \frac{d_k + b_k Q_{k+1}}{(a_k - b_k P_{k+1})} \end{aligned} \quad (E 138)$$

The influence of the neighbour pressures DPX and DPY appear in the term d_k . It can be seen that after solving this set of equations, solutions for DP and DPZ for one column are obtained, while DPX and DPY still have to be computed based on the new values for DP and DPZ .

In a vertical column there are two different kinds of surfaces separating the tetrahedrons: one, which divides the primary grid into two tetrahedrons, and one which forms the ground surface of the primary grid. In the following the non-hydrostatic pressure equation is derived for the latter. In a similar way, the equation for the other surface can be deduced. Fig. 10 shows the definitions of surfaces and their projections in the x - z plane. AZX is the projection of the surface AZ in x -direction. Not shown is the surface AZY , which is the projection of the surface AZ in y -direction. AZZ is the projection of AZ in z -direction.

Fig. 10. Area definitions used in the derivation of the non-hydrostatic pressure equation for the bottom surface AZ of the primary grid



Applying (E 131) for the surface AZ gives the following expression for the tetrahedron indexed b in Fig. 10:

$$\begin{aligned}
& AZZ_{i,j,k} \frac{\Delta t}{V_{i,j,k-1}} (AZZ_{i,j,k} DPZ_{i,j,k} - AZZ_{i,j,k-1} DP_{i,j,k-1}) + \\
& AZX_{i,j,k} \frac{\Delta t}{V_{i,j,k-1}} (AX_{i+1,j,k-1} DPX_{i+1,j,k-1} + AZX_{i,j,k} DPZ_{i,j,k} - [AZX_{i,j,k-1} + AX_{i,j,k-1}] DP_{i,j,k-1}) + \\
& AZY_{i,j,k} \frac{\Delta t}{V_{i,j,k-1}} (AY_{i,j+1,k-1} DPY_{i,j+1,k-1} + AZY_{i,j,k} DPZ_{i,j,k} - [AZY_{i,j,k-1} + AY_{i,j,k-1}] DP_{i,j,k-1})
\end{aligned} \tag{E 139}$$

The first line is the mass-flux via the surface $AZ_{i,j,k}$ induced by the pressure-gradient in z-direction, the second and third lines are the same but in x-, and y-directions. Note, that $AZX_{i,j,k-1} + AX_{i,j,k-1}$ is the projection of the surface, which divides the primary grid into the secondary grid (diagonal), in x-direction.

For the tetrahedron b we get in a similar manner:

$$\begin{aligned}
& AZZ_{i,j,k} \frac{\Delta t}{V_{i,j,k}} (AZZ_{i,j,k} DPZ_{i,j,k} - AZZ_{i,j,k-1} DP_{i,j,k}) + \\
& AZX_{i,j,k} \frac{\Delta t}{V_{i,j,k}} (AX_{i,j,k} DPX_{i,j,k} + AZX_{i,j,k} DPZ_{i,j,k} - [AZX_{i,j,k} + AX_{i,j,k}] DP_{i,j,k}) + \\
& AZY_{i,j,k} \frac{\Delta t}{V_{i,j,k}} (AY_{i,j,k} DPY_{i,j,k} + AZY_{i,j,k} DPZ_{i,j,k} - [AZY_{i,j,k} + AY_{i,j,k}] DP_{i,j,k})
\end{aligned} \tag{E 140}$$

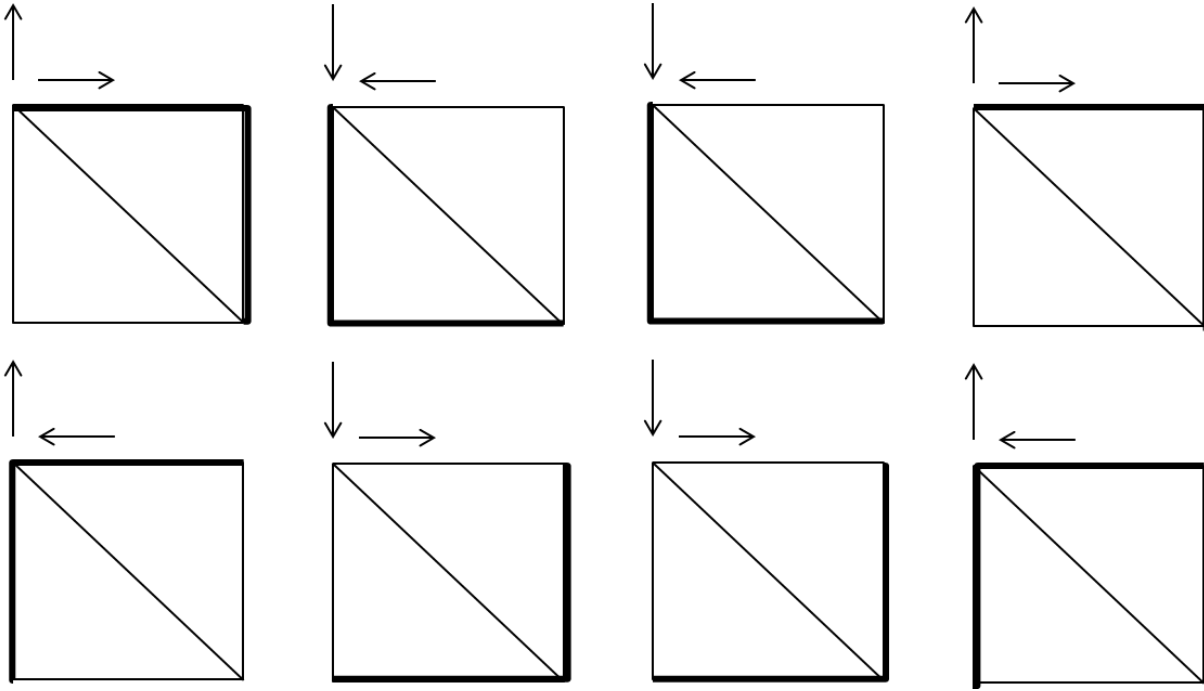
The net mass-fluxes over the surface $AZ_{i,j,k}$ caused by the uncorrected velocities \bar{u}_i^* are computed as:

$$\begin{aligned}
& AZZ_{i,j,k} \left(\rho_{0i,j,k-1} \bar{u}_{3i,j,k-1}^* - \rho_{0i,j,k} \bar{u}_{3i,j,k}^* \right) + \\
& AZX_{i,j,k} \left(\rho_{0i,j,k-1} \bar{u}_{1i,j,k-1}^* - \rho_{0i,j,k} \bar{u}_{1i,j,k}^* \right) + \\
& AZY_{i,j,k} \left(\rho_{0i,j,k-1} \bar{u}_{2i,j,k-1}^* - \rho_{0i,j,k} \bar{u}_{2i,j,k}^* \right)
\end{aligned} \tag{E 141}$$

Next, all terms in (E 139), (E 140), and (E 141) are re-arranged such that all terms not containing DP or DPZ are summarized in the term d_k of (E 135). In an equal manner the terms a_k , b_k , and c_k are obtained. In this way, by using the TDMA, new values for DP and DPZ are computed in every vertical column. After a column has been solved the adjacent values for DPX and DPY are calculated. In order to improve convergence of the iterative procedure, DPX and DPY are updated corresponding to the propagation directions of the loops in x- and y-directions as illustrated in Fig. 11. The number of iterations is fixed to eight, such that each pattern depicted in Fig. 11 is applied once.

Detailed model description - Numerical schemes

Fig. 11. Cell-faces (thick lines) at which the values for DPX and DPY are updated after the calculation of DP and DPZ in a vertical column. The arrows indicate the propagation directions in x- and y-directions of the iterative procedure.



In the following, the equations to obtain DPX are derived. An equation for DPY can be obtained similarly. We start again with (E 131) and deduce a formulation for the surface $AX_{i,j,k}$. For the left-hand tetrahedron at the surface $AX_{i,j,k}$ we get for the mass-flux caused by the pressure-gradient:

$$AX_{i,j,k} \frac{\Delta t}{V_{i-1,j,k}} \left(AX_{i,j,k} DPX_{i,j,k} + AZX_{i-1,j,k+1} DPZ_{i-1,j,k+1} - [AZX_{i-1,j,k} + AX_{i-1,j,k}] DP_{i-1,j,k} \right) \quad (E 142)$$

For the tetrahedron on the right-hand side:

$$AX_{i,j,k} \frac{\Delta t}{V_{i,j,k}} \left(AX_{i,j,k} DPX_{i,j,k} + AZX_{i,j,k} DPZ_{i,j,k} - [AZX_{i,j,k} + AX_{i,j,k}] DP_{i,j,k} \right) \quad (E 143)$$

The net mass-flux over the surface $AX_{i,j,k}$ caused by the uncorrected velocity \bar{u}_1^* is computed as:

$$AX_{i,j,k} \left(\rho_{0i-1,j,k} \bar{u}_{1i-1,j,k}^* - \rho_{0i,j,k} \bar{u}_{1i,j,k}^* \right) \quad (E 144)$$

The terms (E 142), (E 143) and (E 144) are used to get an expression for DPX .

4.7.5 Algorithm to solve the set of conservation equations

The solution algorithm follows closely the so-called SIMPLE (Semi-implicit method for pressure-linked equations) algorithm developed by Patankar (1980). After the model has been initialized with first-guess fields (see chapter 4.8), the following procedure is repeated for each time step:

First, the advection terms for the momentum equations and the advection-diffusion equation for scalars are calculated [see (E 104) and (E 132)]. Second, the geometrical terms needed for the non-hydrostatic pressure equation are computed.

Third, surface layer (Prandtl-layer) quantities such as friction velocity, Obukhov length, and convective velocity are calculated. At this stage the soil temperature, sensible and latent heat fluxes are determined iteratively.

Fourth, the momentum equations are iteratively solved using the TDMA algorithm. Subsequently, lateral boundary values are determined.

Afterwards the mass divergence for each cell of the tetrahedrons is calculated followed by iteratively solving the Poisson equation for the non-hydrostatic pressure field, which is used to determine pressure gradients needed to correct the velocities for fulfilling the continuity equation. It should be noted that these pressure gradients are not used in the momentum equations.

In a next step, the advection-diffusion equations for potential temperature, specific humidity, and optionally those for turbulent kinetic energy and dissipation rate are solved iteratively.

Hereafter, the radiation model is invoked to obtain new values for the incoming and outgoing short- and longwave radiation, taking into account shadowing effects of orography.

Finally, all calculated quantities of the current time-step are stored as “old” values and the procedure is repeated.

4.8 Model initialization

The GRAMM model offers two ways for initialization: (i) a simple point observation of wind speed and –direction near the surface in combination with a stability class (see the accompanying GRAL recommendation guide on how to determine stability classes for the GRAMM-GRAL modelling system; Oettl, 2014a), and (ii) by interpolation of multiple point and profile observations of wind speed, -direction and temperature.

4.8.1 Single point observation of wind speed and –direction plus stability class

In this case, a hypothetical temperature profile is assumed based on the wind speed and stability class. It should be noted that this method is less suited to investigate flow fields for a specific date and time, but is designed for statistical approaches where weather conditions are categorized by wind speed, -direction, and stability classes. In the majority of cases GRAMM is run using such input data.

During daytime conditions the problem of setting the sun's angle above the horizon arises, as no time information is available for the model anymore when using categorized meteorological input data. In the GRAMM model the sun's angle depends on wind speed, stability class, and latitude in order to convey the classification scheme recommended for GRAL. When evoking the "dynamic sunrise scheme", global radiation is computed in a transient way. For convective stability classes a fixed period of 6 hours is computed, whereby the initial time at the beginning of the simulation depends on the stability class in the following way: 6 o'clock for SC 1, 12 o'clock for SC 2, and 9 o'clock for SC 3. The date is computed as mentioned before in dependence on latitude, wind speed, and stability class.

In order to calculate profiles for wind speed, the Obukhov length is required, which is computed in the same manner as for the GRAL model (see Oettl, 2014b). For convenience, the method, which follows closely the German standard boundary-layer model (VDI 3783-8), is presented here:

$$L = \text{Min} \left(\frac{1}{(a \cdot [z_0 \cdot 100]^b)}, -4 \right) \text{ for stability classes A-C,} \quad (\text{E } 145)$$

$$L = \text{Max} \left(\frac{1}{(a \cdot [z_0 \cdot 100]^b)}, 4 \right) \text{ for stability classes E-G} \quad (\text{E } 146)$$

$L = 1000$ for stability class D

a and b are constants, which depend on the stability class, and z_0 is the roughness length.

Table 2. Empirical constants a , and b for the determination of the Obukhov length

	a	b
A	-0.37	-0.55
B	-0.12	-0.50
C	-0.067	-0.56
E	0.01	-0.50

F	0.05	-0.50
G	0.20	-0.55

Standard wind profiles in GRAMM/GRAL are kept quite simple and follow closely those proposed by the US-EPA (2000):

$$u(z) = u(z_a) \cdot \left(\frac{z}{z_a} \right)^{ex} \quad (\text{E 147})$$

$$ex = \text{Max}(0.35 - 0.4 \cdot |L|^{-0.15}, 0.05) \text{ for } L < 0, \text{ and} \quad (\text{E 148})$$

$$ex = 0.56 \cdot L^{-0.15} \text{ for } L \geq 0$$

Thus, standard wind profiles change continuously with stratification. In neutral conditions and/or for high roughness lengths (urban conditions), the wind profile exponent is close to 0.20, while for strongly convective conditions it decreases to 0.05, for strongly stable conditions it increases to about 0.40.

The temperature profile is defined by a constant temperature decrease with increasing height with an adiabatic lapse rate of -0.0065 K m^{-1} representing a humid atmosphere.

The only exception is stability class 7 (most stable atmosphere) at wind speeds lower 0.35 m/s where it is assumed that a temperature inversion up to a height of $1/3$ the difference between the maximum and minimum elevation within the model domain has already been developed. The minimum height of this inversion is set to 400 m . Strong ground inversions are typically encountered between midnight and sunrise. Note that such inversions suppress cold air drainage flows but may lead to distinctive flow effects such as separating flows around hills generating recirculation flows (e.g. Oettl, 2000) or internal gravity waves.

The inversion height is set between 80 and 400 m at stability class 7 and wind speeds above 0.35 m/s depending on the wind speed.

Further, in case of stability class 7 it is assumed that the temperature difference between the lowest layer of the atmosphere and the soil temperature in 1 m depth is $+2 \text{ K}$.

For stability class 6 (moderately stable) this temperature difference is set to $+5 \text{ K}$, which quickly leads to the generation of cold-air drainage flows in the simulation as the corresponding initial vertical temperature profile is neutral causing no damping of vertical movements at the beginning of the simulation.

For stability classes 6 and 7, the initial soil temperature in valleys and bassins (see chapter 9.2) is reduced by further 2° .

Detailed model description - Model initialization

The soil temperature is raised by 1° inside continuous urban fabric areas (corine class 111), by 0.5° inside discontinuous urban fabric areas (corine class 112) and 0.2° in industrial areas (corine class 121) for stability classes > 2.

For all other cases the temperature difference between air and soil is set to zero.

For water bodies the initial temperature difference to the surrounding soil- and surface is set in dependence on the stability class as listed in the following table:

Table 3. Initial temperature difference between water bodies and other surface types in dependence on the stability class

SC	1	2	3	4	5	6	7
ΔT	-10 K	-5 K	-2 K	0	+2 K	+5 K	+10 K

As could be demonstrated by Oettl (2015) these assumptions lead to reasonable wind fields from a statistical point of view (e.g. when examining observed and modeled wind roses, or frequency bins of wind speeds).

GRAMM requires the potential temperature, which is obtained by establishing a pressure field by using

$$p = p_{top} \cdot \exp \frac{g \cdot \left(\frac{1}{T} \right) \cdot \Delta x_3}{R}, \quad (E 149)$$

where the mean (spatial) harmonic temperature is given by

$$\left(\frac{1}{T} \right) = \frac{1}{\Delta x_3} \cdot \int_{z_{top}}^{x_3} \frac{dx_3'}{T(x_3')}, \quad (E 150)$$

As a first guess the pressure at the top of the model domain is calculated by

$$p_{top} = p_0 \cdot \exp \frac{z_{top}}{8000}, \quad (E 151)$$

where the default value for p_0 is 101300 Pa. The hydrostatic pressure at the top of the model domain p_{top} is iteratively adjusted until the pressure at the lowest level p_0 obtained by (E 149) equals the default value.

$$\theta = T \cdot \left(\frac{p_0}{p} \right)^{\frac{R}{c_p}} \quad (\text{E } 152)$$

4.8.2 Using observed profiles of wind and temperature

GRAMM supports the usage of multiple observations either at single stations or at vertical profiles (Sodar, RASS, tethered balloon). Please visit section 9.4 on how to invoke this opportunity. In this case, GRAMM enables the user to define observed hydrostatic pressures at sea level and at the lowest elevation within the model domain. Again, the hydrostatic pressure at the top of the model domain p_0 is iteratively adjusted in order to fit the user defined pressure at the lowest level of the model domain.

In the following the interpolation procedures of observed temperature and wind fields are highlighted.

4.8.2.1 Interpolation between observed temperature profiles

Fig. 12 illustrates schematically the interpolation method for profiles. Interpolated temperatures are assigned to the corresponding grid-cell centre. The distance between the grid-cell in consideration and the location of the observed temperature profile is taken into account by weighting factors. Only those observations are used, which are closest to the height of the grid-cell centre. In case that the grid-cell is higher than the highest observation, both the highest and second highest observations are utilized to compute a vertical temperature gradient for the upward extrapolation. In case that the grid-cell is lower than the lowest observation, the two lowest observations of a profile are utilized to get a temperature gradient for the downward extrapolation. (E 153) is used for the interpolation of temperature profiles, where interpolated values are marked with the subscript *int*.

$$T_{\text{int}}(z_{\text{int}}) = \frac{\sum_i \left(\frac{T_i - T_{i+1}}{z_i - z_{i+1}} \cdot (z_{\text{int}} - z_i) \right) \cdot gew_i}{\sum_i gew_i} \quad (\text{E } 153)$$

$$gew_i = \begin{cases} \frac{1}{r_i^2 + r_{i+1}^2} \Leftrightarrow r_i \neq r_{i+1} \\ \frac{1}{\sqrt{r_i^2 + r_{i+1}^2}} \Leftrightarrow r_i = r_{i+1} \end{cases} \quad (\text{E } 154)$$

r_i Distance between the grid-cell centre and observations

i Total number of observations used for the interpolation

Detailed model description - Model initialization

As indicated in Fig. 12, gradients are computed between every pair of observation used in the interpolation procedure. Crosses depict the location of observations and encircled crosses highlight those observations, which are finally used to get the temperature $T_?$ for a specific grid-cell. In this way surface observations can be included in the interpolation scheme. Although the horizontal temperature gradients between vertical profiles have less influence than vertical ones according to (E 154), their inclusion in the calculations has a smoothing effect on the resulting temperature field.

Fig. 12. Scheme for interpolating temperature profiles and surface stations

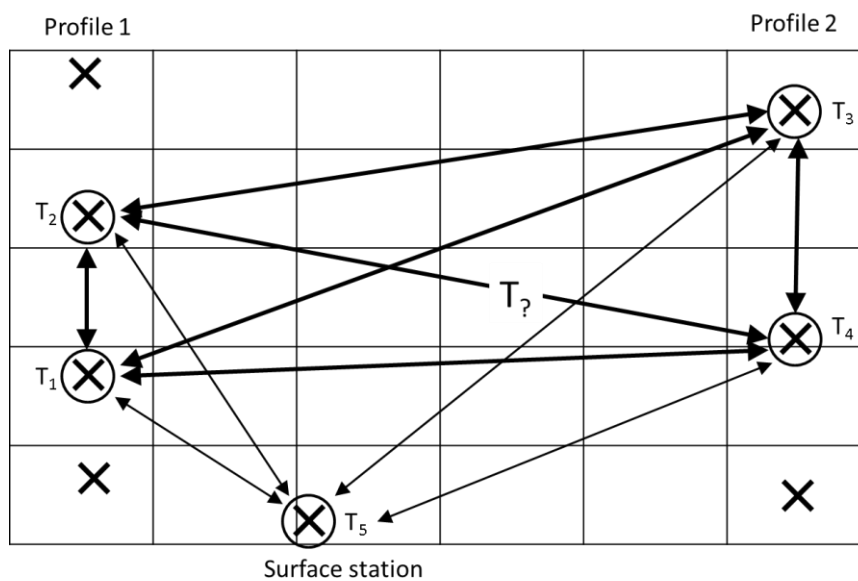
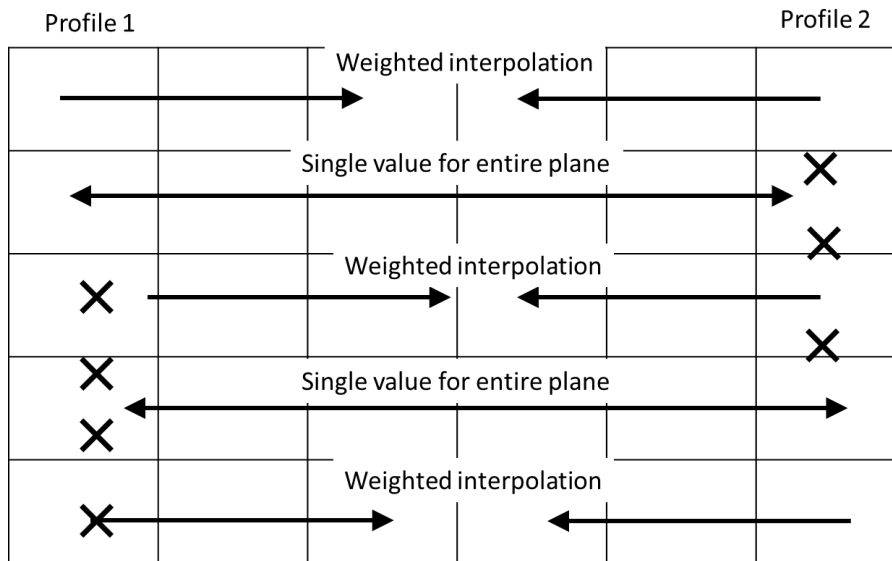


Fig. 13 illustrates the interpolation scheme applied when two observed temperature profiles cover different height ranges. Above the highest observation and below the lowest observation extrapolation is carried out using both profiles, while temperatures at grid levels covered by just one profile are computed on the basis of this observation only.

Fig. 13. Interpolation scheme for observed temperature profiles covering different height ranges



4.8.2.2 Interpolation between observed surface temperatures

Temperatures observed at surface stations can optionally be treated as being part of a vertical profile (see Fig. 12). This is the recommended method in case that the surface observation is representative for large areas (undisturbed atmosphere). Whenever a surface station is included in the interpolation procedure as surface station (note that a different file format is used for surface stations), the observation is treated as local effect and the observed temperature is successively levelled towards the observations at vertical profiles, which are interpreted by the algorithm as being more representative for larger areas (e.g. the atmosphere not influenced by local effects).

Note that wind observations can either be interpolated like temperature observations, i.e. if just a single vertical profile is available, identical values will be found at the same heights above sea level. Optionally, wind observations can be interpolated such that identical values are found at the same heights above orography (terrain-following interpolation). The latter is the standard scheme.

4.8.2.3 Interpolation between observed surface winds

At the first model level, a simple power-law profile for wind speed is applied to extrapolate observed wind speeds close to the Earth's surface:

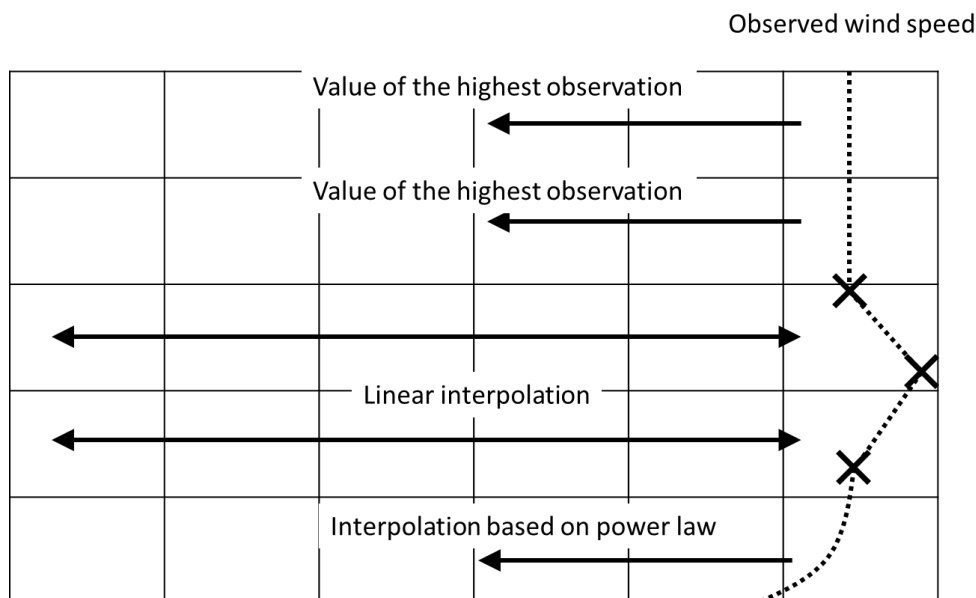
$$u_i(x_3) = u_a \cdot \left(\frac{x_3}{a} \right)^p \quad (\text{E } 155)$$

u_a is the observed wind speed at height a , and p is set equal to 0.25 (ON M9440). Note that above the first model level, surface stations are excluded from the interpolation procedure.

4.8.2.4 Interpolation between observed wind profiles

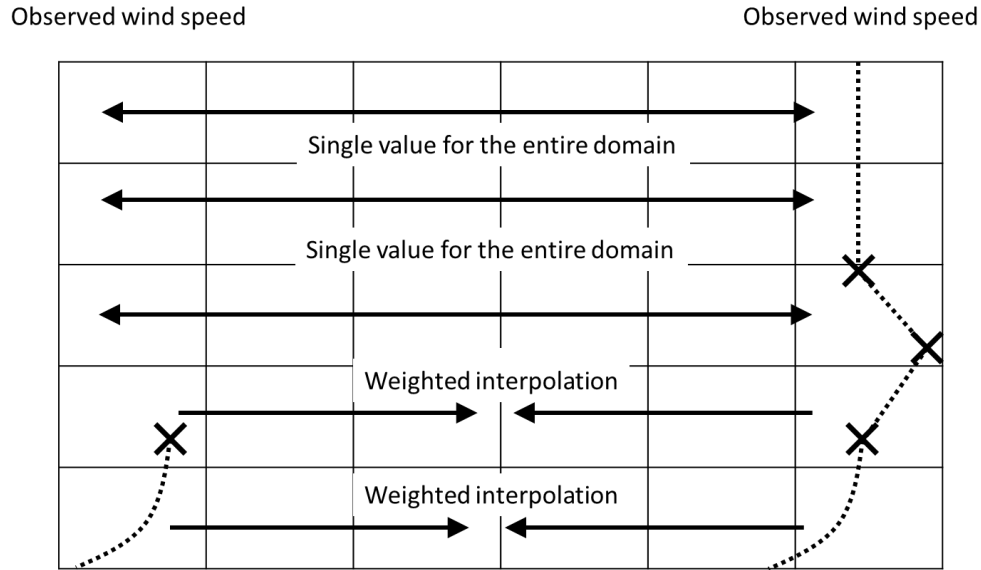
Linear interpolation is applied between pairs of observations. Below the lowest observation the power law as given by (E 155) is utilized, while above the highest observation the wind speed is kept constant and equal to this measurement (Fig. 14).

Fig. 14. Interpolation scheme of an observed wind profile



Whenever multiple observations are available at similar heights, weighted interpolation is carried out, while at heights, where just single observations are at hand, these are used for the entire domain at each grid level (Fig. 15).

Fig. 15. Interpolation scheme for multiple observations (here surface station plus vertical profile)



The interpolation between multiple observations at a certain grid level is quite similar to the one for temperature:

$$u_{\text{int}} = \frac{\sum_i u_{mi} \cdot \frac{1}{r_i}}{\sum_i \frac{1}{r_i}} \quad (\text{E } 156)$$

$$u_{mi} = \begin{cases} u_b \cdot \left(\frac{z_r}{z_b} \right)^p & \forall z_r \leq 70m \\ u_u & \forall z_r \geq z_e \\ u_1 \cdot \left(\frac{z_r}{z_1} \right)^p & \forall z_r < z_1 \\ \left(\frac{u_{n+1} - u_n}{z_{n+1} - z_n} \right) \cdot (z_r - z_n) + u_n & \forall z_n \leq z_r \leq z_{n+1} \end{cases} \quad (\text{E } 157)$$

r_i Distances between observations and grid-cell centres

i Number of observations used for the interpolation

u_{mi} Computed wind speeds according to (E 155)

z_r Height of grid-cell centres above the surface

u_b Observed wind speeds at surface stations

z_b Height of the surface station above ground

u_e Uppermost observed wind speed along a vertical profile

Detailed model description - Model initialization

u_1	Observed wind speeds at the lowest height of a vertical profile
z_1	Height of the lowest point of a profile
u_n, u_{n+1}	Neighbouring wind speeds along an observational profile
z_n, z_{n+1}	Heights above ground level of neighbouring points
z_e	Height above ground of the highest observation at a vertical profile

4.8.2.5 Smoothing of an interpolated wind field

Finally, the interpolated wind field is spatially filtered by using a 5-point filter (Goodin et al., 1980). In this way mass divergence is reduced in a first step.

$$u_{i,j,k} = 0.2 \cdot (u_{i,j,k} + u_{i-1,j,k} + u_{i+1,j,k} + u_{i,j-1,k} + u_{i,j+1,k}) \quad (\text{E } 158)$$

4.8.3 Initialization of the soil temperatures and sensible heat fluxes

Soil temperatures are initialized in dependence on the height above sea level and latitude as outlined in chapter 4.6.3. Optionally, a user may specify that sensible and latent heat fluxes are set to constant values, instead of having them computed by the model itself. In some cases this can make sense, e.g. in the absence of land-use data or if it is not possible to include all of the topography influencing the development of e.g. drainage flows. Currently, the values for latent heat fluxes are set to zero, if this option is used, while those for sensible heat flux depend on the stability class:

Slightly stable (class 5):	15 W/m ²
Stable (class 6):	25 W/m ²
Strongly stable (class 7):	30 W/m ²

5 Compliance with the German Guideline VDI 3783 - 7

The Association of German Engineers (VDI) issued a draft evaluation guideline for prognostic mesoscale wind-field models in 2015 (VDI, 2015), which provides theoretical and field-data for testing model performance. The evaluation procedure outlined in the guideline is based on three major steps: (1) The 'general evaluation' step, which is about traceability and proper documentation of a model. (2) A 'scientific evaluation' step dealing with obligations regarding publications in peer-reviewed scientific journals. (3) The 'validation' section, where a model is checked based on a number of test cases. These can be classified into two sub-groups: (a) Test cases addressing general model properties, such as the dependency of modelled flows on grid resolution or tests to ensure flow convergence.

(b) Test cases using reference data from field observations.

In the following, all of the requirements of that guideline are listed in detail and compliance or non-compliance with the current GRAMM version is discussed.

5.1 General model evaluation

The following documents about the model and the programme (source code) are required:

Brief description: Most of the information required by the guideline for the brief description can be found at the GRAL website (<http://lampx.tugraz.at/~gral/>). Further information is given at the beginning of this document.

Detailed description of the model: It shall comprise the basic equations, approximations, parameterizations, and boundary conditions employed. Furthermore, the evaluation of the model according to the VDI guideline 3783-7 shall be explained in detail. It is this report that aims at providing all of this information in combination with the report about recommendations when using GRAL/GRAMM (<http://lampx.tugraz.at/~gral/>).

Manual: The manual is about the installation, user interface, and general operation of the model. All these subject matters are included in this document as well as in the User Guide for the Graphical User Interface (GUI). Applications are described in the model validation section of this report.

Technical reference (optional): Shall consist of the programming conventions, the programming language, and a list of variables. As GRAMM is written in C# language, programming conventions are very strict. It is felt that an extra documentation about the general programming conventions of C# is not necessary. All variables used in the GRAMM model are listed in this report (see sect. 9.3).

Furthermore, it is required that a third party is allowed to inspect the source code of the programme. The source code of GRAMM can be requested by anyone interested in it.

Finally, there have to be two certified publications on model physics and model results in at least two different professional journals. This criterion is already fulfilled as can be seen from the publication list in this report (see references in sect. 7). A complete list of peer-reviewed publications in international journals can be found on the website <http://lampx.tugraz.at/~gral/>.

5.2 Scientific model evaluation

This section deals with the basic equations and parameterizations used in the model. The guideline requires the following methodological approaches:

Table 4: Scientific evaluation according to VDI 3783-7

	YES	NO
Equations are Reynolds averaged	X	
All three wind components prognostic	X	
Potential temperature prognostic	X	
Coriolis force included	X	
Continuity equation complete or inelastic approximation	X	
Bouyancy included	X	
Turbulence parameterization in dependence on stability	X	
Continuous fluxes as a function of space	X	
Continuous fluxes as a function of stability	X	
Application of Monin-Obukhov similarity theory in the lowest grid layer	X	
Convective case: parameterizing subscale boundary-layer convection	X	
Land-use properties	X	
Radiation fluxes at the surface	X	
Inclination of orography, shadowing effects	X	
Specific humidity prognostic	X	
Humidity budget at the surface	X	

5.3 Model validation

In the following, model results for eight different test cases are outlined. In each chapter results for the test cases are presented in detail. The guideline defines so-called hit rates q in the following way:

$$q = \frac{N}{n} = \frac{1}{n} \sum_{i=1}^n N_i \quad (\text{E } 159)$$

$$N_i = \begin{cases} 1 & \text{if } \left| \frac{P_i - O_i}{O_i} \right| \leq D \quad \text{or} \quad |P_i - O_i| \leq W \\ 0 & \text{else} \end{cases} \quad (\text{E } 160)$$

where N is the number of data points counted as hit, n is the total number of data points, and O_i and P_i are reference and modelled quantities at location i , respectively. Hit rates are required to be equal or higher than 95 %. The required maximum relative difference D and the maximum absolute difference W are defined as listed in Table 5.

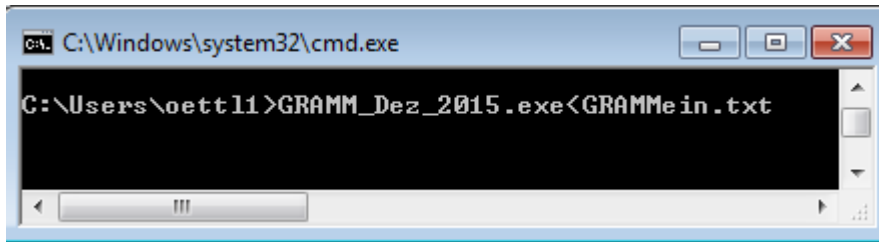
Table 5: Eligible absolute deviation W and relative deviation D for computing hit rates.

<i>Parameter</i>	<i>W</i>	<i>D</i>
u, v, w	0.35 m s ⁻¹	10 %
Wind speed	0.5 m s ⁻¹	10 %
Temperature	0.5 K	0.2 %
Specific humidity	0.0002 kg kg ⁻¹	2 %

The guideline differentiates between the so-called prognostic domain, which defines the area where model results are compared with reference values, and the model domain. The latter needs to be at least three times the size of the prognostic domain in the horizontal directions.

Note that all test cases require very specific wind and temperature profiles for initialization. These have to be provided by text files that can be generated as outlined in sect. 9.4.1.7. In a first step the model grid can be established using the GUI. Afterwards the users has to change the first letter in the file GRAMMin.dat from “y” to “n”, which means that GRAMM is not driven by meteorological data stored in the standard file “meteopgt.all” but by user defined profiles. Using this way of initialization causes the model to prompt for some more information (e.g. the pressures at sea level and at the the lowest model elevation). It is recommended to save all this required information in a text file (here the file name “GRAMMein.txt” was used). For convenience, a screen shot of this file is provided for each test case. Once all necessary information has been stored in a file, GRAMM is best launched in a command window using the command as illustrated in the following figure:

Fig. 16. Launching GRAMM in a command window utilizing a text file containing all necessary information required by GRAMM when it is not driven by the file "meteopgt.all"



5.3.1 Test case E1 (gravity waves)

In this test case the ability of a model to simulate gravity waves induced by terrain is tested. Model results are compared with an analytical solution for stable stratified flows over a single Gaussian shaped hill.

5.3.1.1 Model domain and grid sizes

Size of prognostic domain: 8 x 0.4 x 5 km

Size of model domain: 24 x 8 x 11.7 km

Horizontal grid size: 100 m

Vertical grid size, stretching factor, number of vertical grid cells: 20 m, 1.11, 40

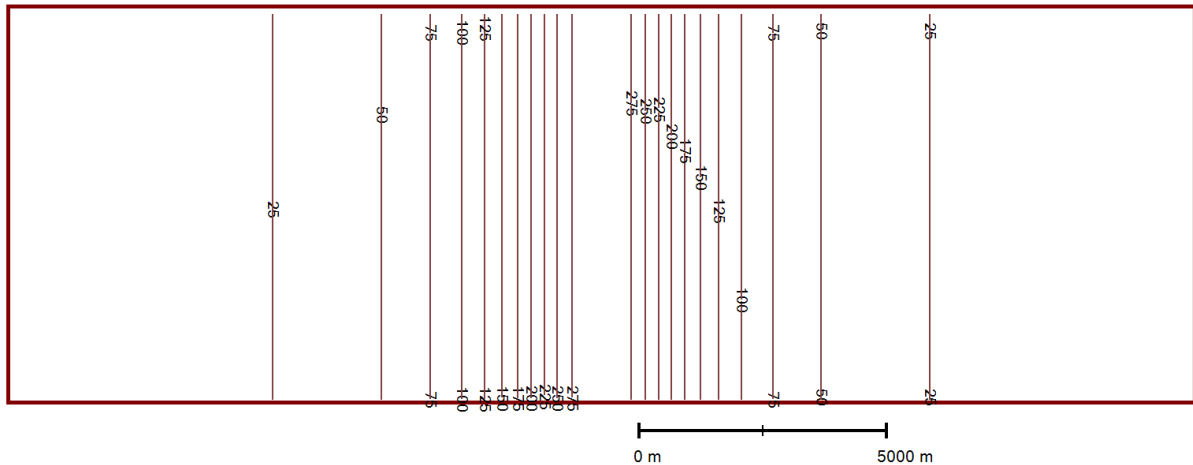
5.3.1.2 Orography and land-use

Homogenous land-use properties are assumed. The roughness length is set to 0.0003 m.

Orography is defined by the following equation:

$$z(x) = 300 \frac{2000^2}{2000^2 + x^2} \quad (\text{E 161})$$

Fig. 17. Orography for test case E1



5.3.1.3 Initialization

The potential temperature gradient is set to 0.0035 K m^{-1} , and at the surface the potential temperature is 290 K. Specific humidity is set to 0.5 %. The model has been initialized with the following wind profile.

Table 6. Wind profile use for initializing GRAMM for test case E1

<i>Height [m]</i>	<i>Wind speed [m s^{-1}]</i>	<i>Wind direction [deg]</i>
10	3.6	244
30	4.2	245
50	4.9	248
70	5.4	251
90	5.8	254
120	6.3	259
150	6.5	265
190	6.4	269
220	6.2	271
300	6.0	270

By default the flags in the GRAMM input file IIN.dat in line 18 to compute surface temperatures (TB) and radiation (STR) are equal to one. In this specific test case it is required to keep the surface temperature constant, thus, both flags need to be set to zero as can be seen from Fig. 18.

Simulation time is 5 hours (=18,000 s; see line 4 in IIN.dat). The date and time of the simulation start has no influence on the results as the radiation model is not invoked.

Fig. 18. Screen shot of the GRAMM input file IIN.dat

```

COMPUTATION DATE          (YYMMDD)          : 080210
BEGINNING OF COMPUTATION  (hhmm)            : 0000
MAXIMUM ALLOWED TIME STEP DT [ s ]          : 5
MODELLING TIME (FOR VALUES >1(s) AND <1[%]) : 18000
BUFFERING AFTER TIMESTEPS : 3600
MAX. PERMISSIBLE W-DEVIATION ABOVE < 1 [mm/s] : 0.01
RELATIVE INITIAL HUMIDITY [ % ] GT.0        : 0.5
HEIGHT OF THE LOWEST COMPUTATION HEIGHT [ m ] : 330.
AIR TEMPERATURE AT GROUND [ K ]             : 283.0
GRADIENT OF THE POTENTIAL TEMPERATURE [K/100m]: -0.01
NEUTRAL LAYERING UP TO THE HEIGHT ABOVE GROUND: 10000
SURFACE TEMPERATURE [ K ]                   : 283.0
TEMPERATURE OF THE SOIL IN 1 M DEPTH [ K ]   : 283.0
LATITUDE : 47
UPDATE OF RADIATION (TIMESTEPS) : 10000
DEBUG LEVEL 0 none, 3 highest : 0
COMPUTE U V W PN T GW FO YES=1 : 1111100
COMPUTE BR PR QU PSI TE TB STR YES=1 : 1100000
DIAGNOSTIC INITIAL CONDITIONS YES=1 : 1
EXPLICIT/IMPLICIT TIME INTEGRATION I=1/E=0 : 1
RELAXATION VELOCITY : 0.15
RELAXATION SCALARS : 0.15
Force catab flows -40/-25/-15 W/m² AKLA 7/6/5 : 0
NESTING (=1) OR LARGE SCALE FORCING (=0) : 0
BOUNDARY CONDITION (1,2,3,4,5,6) : 6
NON-STEADYSTATE/FORCING W. 'METEO.ALL' YES=1 : 0
NON-STEADYSTATE/FORCING W. IFU-MM5 DATA YES=1 : 0
GRAMM IN GRAMM NESTING YES=1 : 0
Flowfield Output Format : 0

```

Fig. 19. Screen shot of the GRAMM input file GRAMMein.txt

```

1 n
2 y
3 input_e2.txt
4 n
5 n
6 100000
7 100000
8 .1
9 .0003
10 .1
11 .000001
12 1
13 1

```

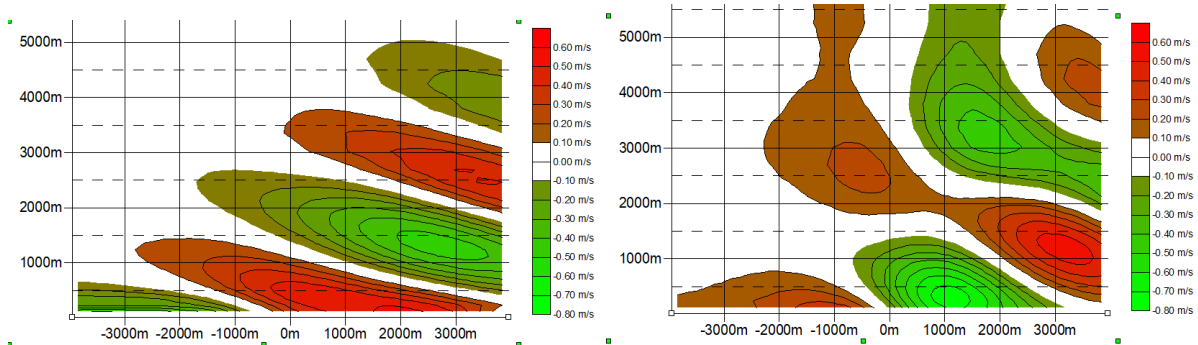
5.3.1.4 Results

The guideline requires that the wave length of the gravity waves on the lee-ward side of the mountain is in the range 3150 – 3800 m. As can be seen from Fig. 20, the wave length computed by GRAMM is about 3200 m.

Further, it is required that computed wind- and temperature fields in the prognostic domain are horizontally homogenous. Compared are the computed values at the most southern with

those of the most northern vertical slice. Hit rates for the GRAMM model are 100 % for all parameters.

Fig. 20. Gravity waves computed with GRAMM for test case E1; u -component (left); w -component (right)



5.3.2 Test case E2 (gravity waves, influence of wind speed)

Test case E2 is essentially the same as E1 and differs solely in the initial wind profile:

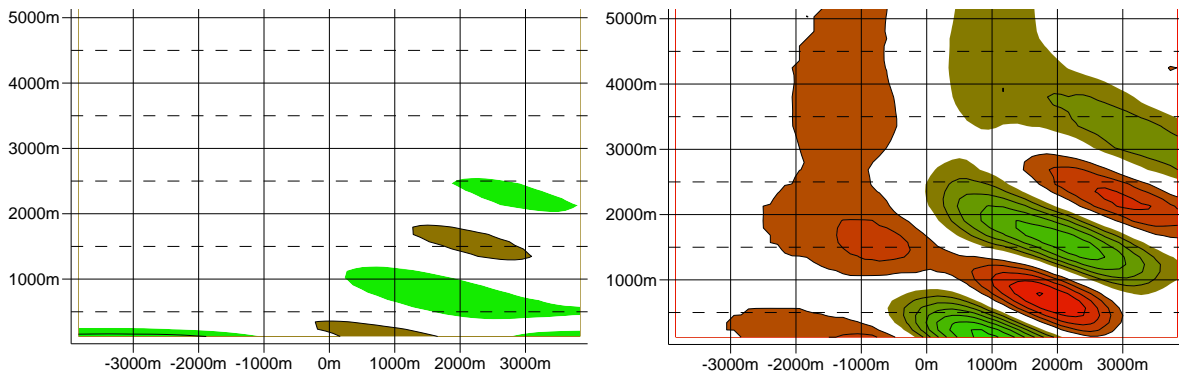
Table 7. Wind profile use for initializing GRAMM for test case E2

Height [m]	Wind speed [$m s^{-1}$]	Wind direction [deg]
10	1.9	242
30	2.4	245
50	2.9	254
70	3.2	261
90	3.3	266
120	3.1	271
150	3.0	270
190	3.0	270

5.3.2.1 Results

The guideline requires that the wave length of the gravity waves on the lee-ward side of the mountain is in the range 1550 – 1900 m. As can be seen from Fig. 21, the wave length computed with GRAMM is about 1700 m for the u -component and about 1800 m for the w -component.

Fig. 21. Gravity waves as computed with GRAMM for test case E2; u -component (left); w -component (right)



5.3.3 Test case E3 (influence of grid size)

Here, the dependency of model results on the horizontal grid size as well as flow convergence is tested. Model results are compared for two different horizontal grid sizes and at two subsequent instants in time for a stable stratified flow over a single three-dimensional Gaussian shaped hill.

5.3.3.1 Model domain and grid sizes

Size of prognostic domain: 5 x 5 x 3 km

Size of model domain: 30 x 30 x 10.6 km

Horizontal grid sizes: 75 m and 100 m

Vertical grid size, stretching factor, number of vertical grid cells: 20 m, 1.16, 30

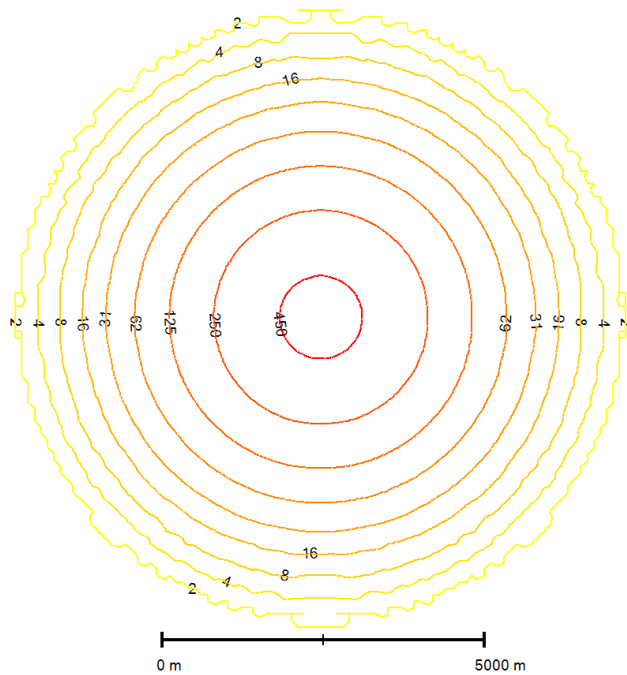
5.3.3.2 Orography and land-use

Homogenous land-use properties are assumed. The roughness length is set to 0.01 m.

Orography is defined by the following equation:

$$z(x) = 500 \exp\left(-\frac{x^2 + y^2}{2000^2}\right) \quad (\text{E } 162)$$

Fig. 22. Orography for test case E3



5.3.3.3 Initialization

The potential temperature gradient is set to 0.0035 K m^{-1} , and at the surface the potential temperature is 290 K. Specific humidity is set to 0.5 %. The model has been initialized with the following wind profile.

Table 8. Wind profile use for initializing GRAMM for test case E3

<i>Height [m]</i>	<i>Wind speed [m s^{-1}]</i>	<i>Wind direction [deg]</i>
10	2.2	239
30	2.8	242
50	3.5	247
70	3.9	252
90	4.3	259
120	4.4	267
150	4.1	271
190	4.0	270

By default the flags in the GRAMM input file IIN.dat in line 18 to compute surface temperatures (TB) and radiation (STR) are equal to one. In this specific test case it is required to keep the surface temperature constant, thus, both flags need to be set to zero as can be seen from Fig. 23.

Simulation time is 5 hours (=18,000 s; see line 4 in IIN.dat). The date and time of the simulation start has no influence on the results as the radiation model is not invoked.

Fig. 23. Screen shot of the GRAMM input file IIN.dat

```

1  COMPUTATION DATE      (YYMMDD)      : 080210
2  BEGINNING OF COMPUTATION (hhmm)      : 0000
3  MAXIMUM ALLOWED TIME STEP DT [ s ]    : 10
4  MODELLING TIME (FOR VALUES >1(s) AND <1[%]) : 18000
5  BUFFERING AFTER TIMESTEPS              : 3600
6  MAX. PERMISSIBLE W-DEVIATION ABOVE < 1 [mm/s] : 0.01
7  RELATIVE INITIAL HUMIDITY [ % ] GT.0  : 0.5
8  HEIGHT OF THE LOWEST COMPUTATION HEIGHT [ m ] : 330.
9  AIR TEMPERATURE AT GROUND [ K ]       : 283.0
10 GRADIENT OF THE POTENTIAL TEMPERATURE [K/100m]: 0.01
11 NEUTRAL LAYERING UP TO THE HEIGHT ABOVE GROUND: 10000
12 SURFACE TEMPERATURE [ K ]             : 283.0
13 TEMPERATURE OF THE SOIL IN 1 M DEPTH [ K ] : 283.0
14 LATITUDE                          : 50
15 UPDATE OF RADIATION (TIMESTEPS)       : 10000
16 DEBUG LEVEL 0 none, 3 highest         : 0
17 COMPUTE U V W PN T GW FO              YES=1 : 1111100
18 COMPUTE BR PR QU PSI TE TB STR        YES=1 : 1110000
19 DIAGNOSTIC INITIAL CONDITIONS          YES=1 : 1
20 EXPLICIT/IMPLICIT TIME INTEGRATION I=1/E=0 : 1
21 RELAXATION VELOCITY                   : 0.05
22 RELAXATION SCALARS                    : 0.05
23 Force catab flows -40/-25/-15 W/m2 AKLA 7/6/5 : 0
24 NESTING (=1) OR LARGE SCALE FORCING (=0) : 0
25 BOUNDARY CONDITION (1,2,3,4,5,6)      : 6
26 NON-STEADYSTATE/FORCING W. 'METEO.ALL' YES=1 : 0
27 NON-STEADYSTATE/FORCING W. IFU-MMS DATA YES=1 : 0
28 GRAMM IN GRAMM NESTING YES=1          : 0
29 Flowfield Output Format                 : 0

```

Fig. 24. Screen shot of the GRAMM input file GRAMMein.txt

```

1  n
2  y
3  input_e3a.txt
4  n
5  n
6  100000
7  100000
8  .1
9  .01
10 .1
11 .000001
12 1
13 1

```

5.3.3.4 Results

The guideline requires that computed wind fields in the prognostic domain are steady-state at the lowest grid level. For the evaluation the computed wind fields (based on the horizontal grid size of 75 m) four and five hours after the model start are compared. Hit rates for the GRAMM model are 100 % for all wind components.

In a next step, computed wind fields at the lowest level five hours after the simulation start for the two different grid sizes of 75 m and 100 m are compared. Also in this case hit rates are 100 % for all wind components.

Fig. 25. Flow fields as computed with GRAMM for test case E3; four hours simulation time (left); five hours simulation time (right); horizontal grid size is 100 m

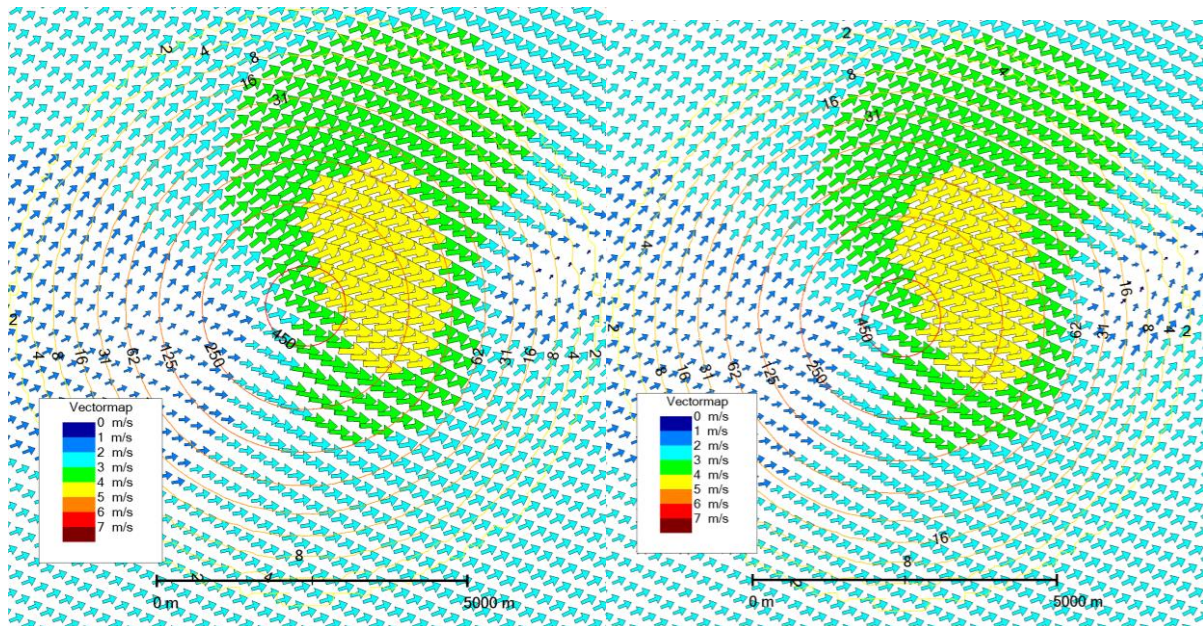
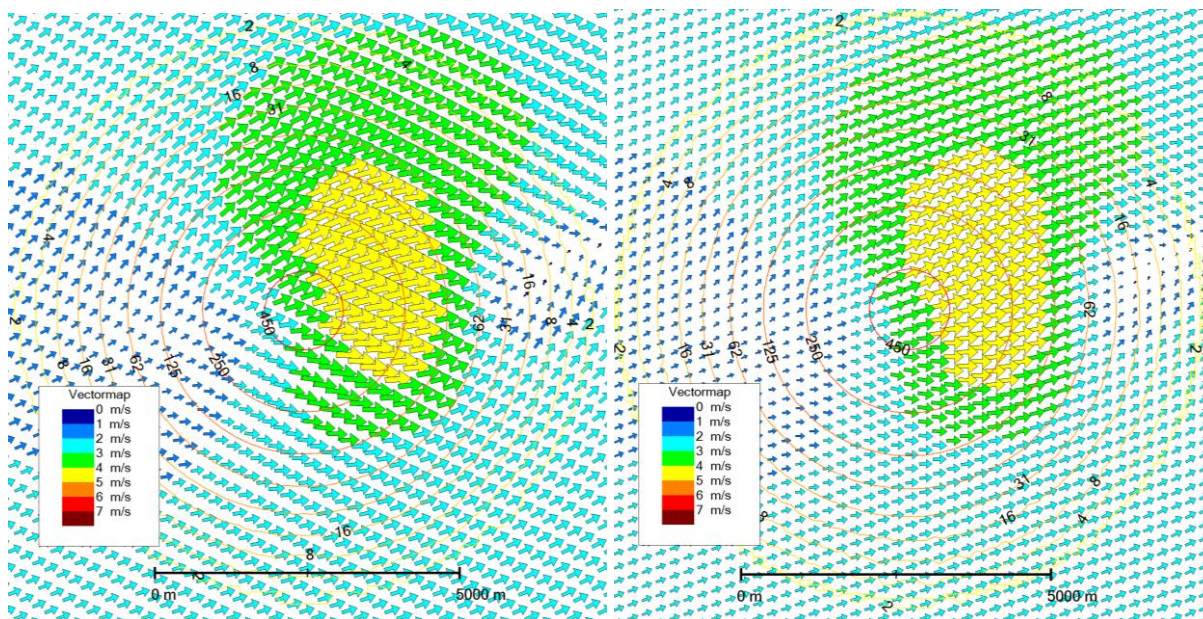


Fig. 26. Flow fields as computed with GRAMM for test case E3; 100 m resolution (left); 75 m resolution (right)



5.3.4 Test case E4 (shadowing effects and drainage flows)

In this test case diurnal variations of flow- and temperature fields influenced by a simple orography are evaluated.

5.3.4.1 Model domain and grid sizes

Same as E3 with a horizontal grid size of 100 m.

5.3.4.2 Orography and land-use

Homogenous land-use properties are assumed. The roughness length is set to 0.01 m. Orography is the same as in case E3.

5.3.4.3 Initialization

The potential temperature gradient is set to 0.0035 K m^{-1} , and at the surface the potential temperature is 290 K. Specific humidity is set to 40 % near the surface and about 5 % at an elevation of 10 km. The model has been initialized with the following wind profile.

Table 9. Wind profile use for initializing GRAMM for test case E4

<i>Height [m]</i>	<i>Wind speed [m s^{-1}]</i>	<i>Wind direction [deg]</i>
10	0.76	244
30	1.0	251
50	1.0	269
70	1.0	270
90	1.0	270
120	1.0	270

Simulation time is 20 hours ($=72,000 \text{ s}$; see line 4 in IIN.dat). The date is set to 21 June and the start-time is set such that sunset in the model takes place about 1.5 hours after the simulation start. The radiation and surface model is invoked. Deep soil temperature is set to 286 K, the height dependency of the deep soil temperature is fixed in GRAMM and is set to 0.005 K m^{-1} . Surface temperature is the same as the temperature of the atmosphere in the lowest grid level. This is achieved in GRAMM by setting equal values for “Air temperature at ground” and “Surface temperature” in the file IIN.dat (for more information visit chap. 9.4.1.8).

Fig. 27. Screen shot of the GRAMM input file IIN.dat

```

COMPUTATION DATE      (YYMMDD)      : 100621
BEGINNING OF COMPUTATION (hhmm)      : 1830
MAXIMUM ALLOWED TIME STEP DT [ s ]    : 5
MODELLING TIME (FOR VALUES >1(s) AND <1[%]) : 57600
BUFFERING AFTER TIMESTEPS              : 3600
MAX. PERMISSIBLE W-DEVIATION ABOVE < 1 [mm/s] : 0.01
RELATIVE INITIAL HUMIDITY [ % ] GT.0   : 40
HEIGHT OF THE LOWEST COMPUTATION HEIGHT [ m ] : 0.
AIR TEMPERATURE AT GROUND [ K ]        : 290.0
GRADIENT OF THE POTENTIAL TEMPERATURE [K/100m]: -0.01
NEUTRAL LAYERING UP TO THE HEIGHT ABOVE GROUND: 10000
SURFACE TEMPERATURE [ K ]              : 290.0
TEMPERATURE OF THE SOIL IN 1 M DEPTH [ K ] : 286.0
LATITUDE                  : 50
UPDATE OF RADIATION (TIMESTEPS)        : 50
DEBUG LEVEL 0 none, 3 highest          : 0
COMPUTE U V W PN T GW FO YES=1        : 1111100
COMPUTE BR PR QU PSI TE TB STR YES=1  : 1110011
DIAGNOSTIC INITIAL CONDITIONS YES=1    : 1
EXPLICIT/IMPLICIT TIME INTEGRATION I=1/E=0 : 1
RELAXATION VELOCITY          : 0.15
RELAXATION SCALARS           : 0.15
Force catab flows -40/-25/-15 W/m2 AKLA 7/6/5 : 0
NESTING (=1) OR LARGE SCALE FORCING (=0) : 0
BOUNDARY CONDITION (1,2,3,4,5,6)       : 6
NON-STEADYSTATE/FORCING W. 'METEO.ALL' YES=1 : 0
NON-STEADYSTATE/FORCING W. IFU-MM5 DATA YES=1 : 0
GRAMM IN GRAMM NESTING YES=1          : 0
Flowfield Output Format                : 0

```

Fig. 28. Screen shot of the GRAMM input file GRAMMein.txt

```

1 n
2 y
3 input_e4.txt
4 n
5 n
6 100000
7 100000
8 .1
9 .01
10 .1
11 .000001
12 1
13 1

```

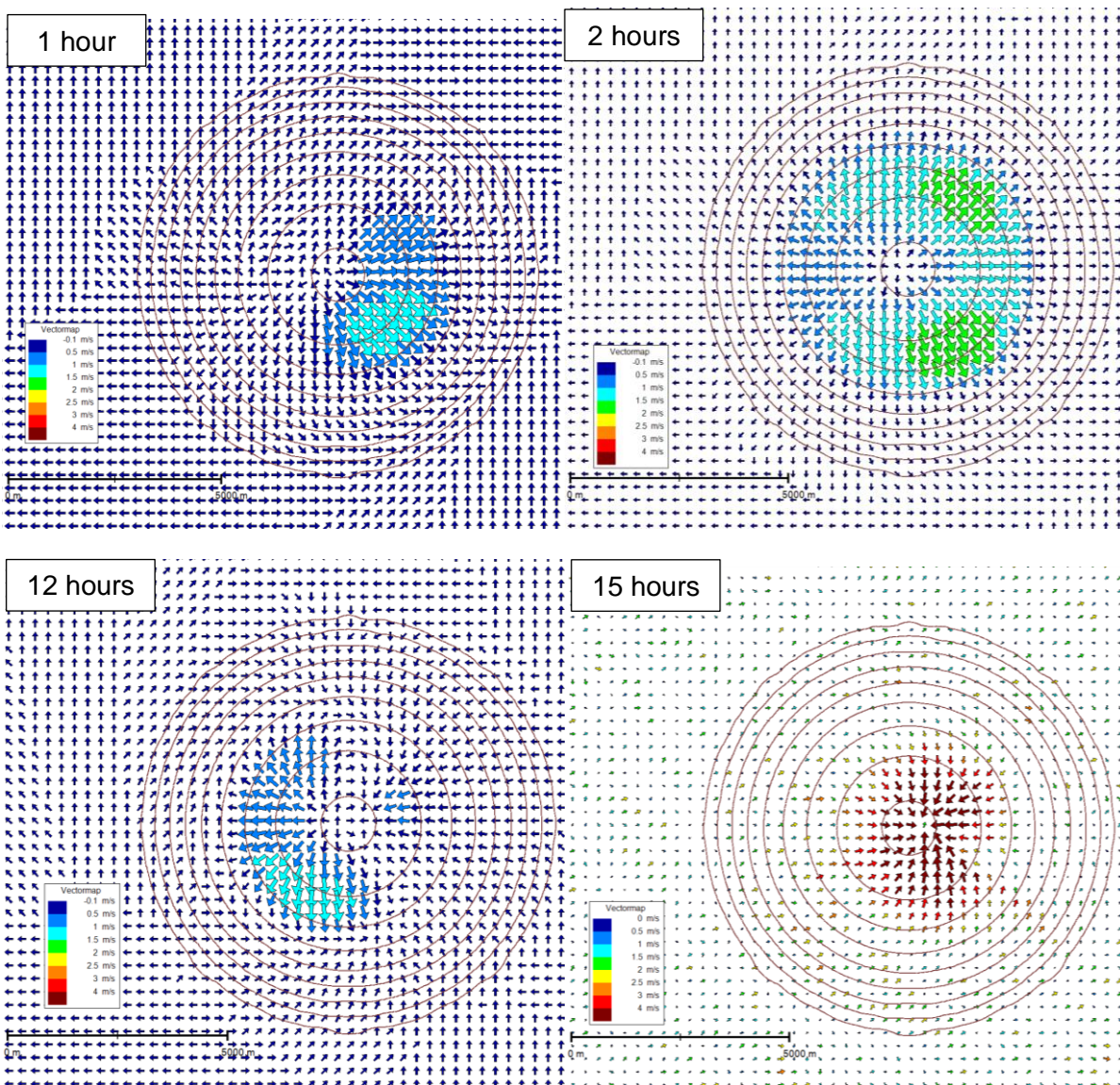
5.3.4.4 Results

The guideline requires that in the course of the night cold air drainage flows develop that superimpose the westerly large-scale flow. It is expected that these drainage flows develop first at the hill slopes orientated eastwards. After sunrise drainage flows are expected to stop primarily at these hill slopes as these are heated up by the sun at first. Some selected GRAMM flow fields are depicted in Fig. 29 to illustrate this behaviour. After one hour weak drainage flows have developed at the slopes orientated westwards, while on the opposite slopes drainage flows superimposing the westerly large-scale flow are much stronger. Already two hours after starting the simulation drainage flows develop in every orientation.

Compliance with the German Guideline VDI 3783 - 7 - Model validation

The development of upslope winds can be seen after 12 modelling time. Upslope winds at the slopes orientated eastwards develop first compared to the other slopes, where still downslope winds prevail, while 15 hours after the simulation start, upslope winds dominate all slopes and became stronger.

Fig. 29. Flow fields as computed with GRAMM for test case E4 several hours after the simulation start



5.3.5 Test case E5 (homogeneity)

In this test case invariance of modelled flow fields with regard to wind direction is tested for the symmetric hill already used in test cases E3 and E4.

5.3.5.1 Model domain and grid sizes

Same as for E3; horizontal grid size is 100 m.

5.3.5.2 Orography and land-use

Homogenous land-use properties are assumed. The roughness length is set to 0.01 m. Orography is the same as in case E3.

5.3.5.3 Initialization

The potential temperature gradient is set to 0.0035 K m^{-1} , and at the surface the potential temperature is 290 K. Specific humidity is set to 0.5 %. The model has been initialized with the following wind profile.

Table 10. Wind profile use for initializing GRAMM for test case E5

<i>Height [m]</i>	<i>Wind speed [m s^{-1}]</i>	<i>Wind direction [deg]</i>
10	2.2	14
30	2.8	17
50	3.5	22
70	3.9	27
90	4.3	34
120	4.4	42
150	4.1	46
190	4.0	45

By default the flags in the GRAMM input file IIN.dat in line 18 to compute surface temperatures (TB) and radiation (STR) are equal to one. In this specific test case it is required to keep the surface temperature constant, thus, both flags need to be set to zero as can be seen from Fig. 30.

Simulation time is 5 hours (=18,000 s; see line 4 in IIN.dat). The date and time of the simulation start has no influence on the results as the radiation model is not invoked.

Fig. 30. Screen shot of the GRAMM input file IIN.dat

```

1 COMPUTATION DATE      (YYMMDD)      : 080210
2 BEGINNING OF COMPUTATION (hhmm)      : 0000
3 MAXIMUM ALLOWED TIME STEP DT [ s ]    : 10
4 MODELLING TIME (FOR VALUES >1(s) AND <1[%]) : 18000
5 BUFFERING AFTER TIMESTEPS              : 3600
6 MAX. PERMISSIBLE W-DEVIATION ABOVE < 1 [mm/s] : 0.01
7 RELATIVE INITIAL HUMIDITY [ % ] GT.0   : 0.5
8 HEIGHT OF THE LOWEST COMPUTATION HEIGHT [ m ] : 330.
9 AIR TEMPERATURE AT GROUND [ K ]        : 283.0
10 GRADIENT OF THE POTENTIAL TEMPERATURE [K/100m]: 0.01
11 NEUTRAL LAYERING UP TO THE HEIGHT ABOVE GROUND: 10000
12 SURFACE TEMPERATURE [ K ]             : 283.0
13 TEMPERATURE OF THE SOIL IN 1 M DEPTH [ K ] : 283.0
14 LATITUDE                          : 50
15 UPDATE OF RADIATION (TIMESTEPS)       : 10000
16 DEBUG LEVEL 0 none, 3 highest         : 0
17 COMPUTE U V W PN T GW FO              YES=1 : 1111100
18 COMPUTE BR PR QU PSI TE TB STR        YES=1 : 1110000
19 DIAGNOSTIC INITIAL CONDITIONS          YES=1 : 1
20 EXPLICIT/IMPLICIT TIME INTEGRATION I=1/E=0 : 1
21 RELAXATION VELOCITY                   : 0.05
22 RELAXATION SCALARS                    : 0.05
23 Force catab flows -40/-25/-15 W/m² AKLA 7/6/5 : 0
24 NESTING (=1) OR LARGE SCALE FORCING (=0) : 0
25 BOUNDARY CONDITION (1,2,3,4,5,6)       : 6
26 NON-STEADYSTATE/FORCING W. 'METEO.ALL' YES=1 : 0
27 NON-STEADYSTATE/FORCING W. IFU-MMS DATA YES=1 : 0
28 GRAMM IN GRAMM NESTING YES=1          : 0
29 Flowfield Output Format                 : 0

```

Fig. 31. Screen shot of the GRAMM input file GRAMMein.txt

```

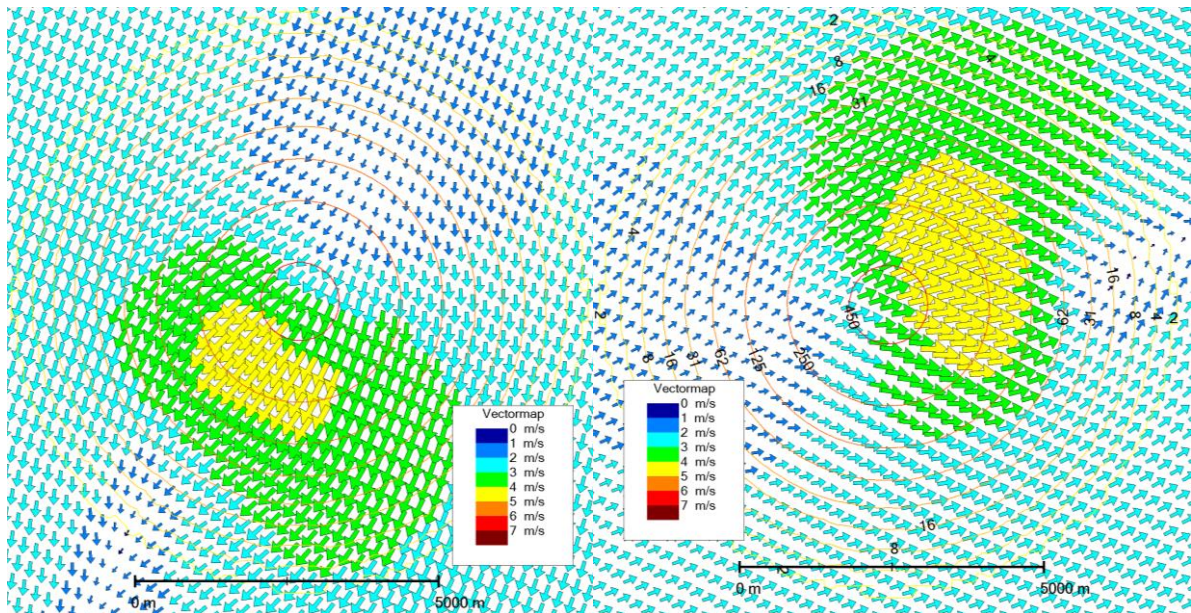
1 n
2 y
3 input_e5.txt
4 n
5 n
6 100000
7 100000
8 .1
9 .01
10 .1
11 .000001
12 1
13 1

```

5.3.5.4 Results

The guideline requires that computed wind fields in the lowest grid level of the prognostic domain are identical with those from test case E3 (results from E5 are rotated by 135 deg. at first). For wind speed the maximum deviations are 0.5 m s^{-1} or 10 %, while for wind direction the maximum deviation is ± 10 deg. For wind speeds below 1 m s^{-1} wind direction is not considered in the calculation of the hit rates. Hit rates achieved with GRAMM are 98.2 % for wind speed and 100 % for wind direction.

Fig. 32. Flow fields as computed with GRAMM for test case E5; 45 deg. large-scale wind direction (left); 270 deg. large-scale wind direction (right)



5.3.6 Test case E6 (field data Sophienhoehe)

Here, model results are compared with field data observed at an isolated peak (Sophienhoehe) in Germany.

5.3.6.1 Model domain and grid sizes

Size of prognostic domain: 5.5 x 5.8 km (no specific limit in the vertical direction)

Size of model domain: 19.6 x 19.4 x 2.1 km

Horizontal grid size: 100 m

Vertical grid size, stretching factor, number of vertical grid cells: 10 m, 1.21, 20

5.3.6.2 Orography and land-use

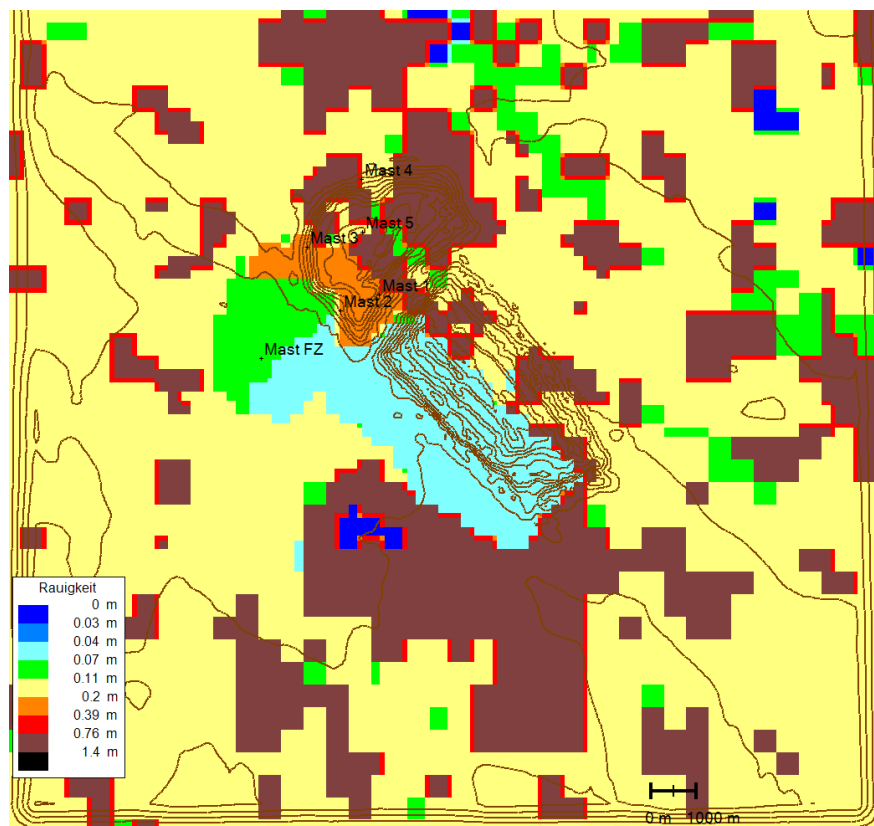
Orographical and land-use data is provided by the VDI guideline. The lowest elevation is given by -136 m, the original orography has been elevated by 136 m to avoid negative values.

The following land-use categories are provided: water (0), sand (2), mixed land-use (3), pasture (4), low forest (5), shrubs (6), mixed forest (7), coniferous forest (8), and built-up area (9). The numbers in brackets denote the corresponding land-use code used in Table 11. The values listed in Table 11 have been defined based on the default GRAMM values for CORINE land-use categories (see Table 1).

Table 11. Land-use categories for test case E6 and corresponding roughness lengths

Code	Roughness length [m]
0	0.0001
2	0.05
3	0.2
4	0.1
5	0.25
6	0.10
7	1.00
8	1.00
9	1.00

Fig. 33. Orography, roughness length, and location of wind observations for test case E6



5.3.6.3 Initialization

Profiles for wind and temperature are provided by the guideline (see Table 12). Note that all data has been shifted vertically by 136 m. At the lowest elevation it was assumed that wind speed is zero, while the temperature gradient of the lowest observations has been extrapolated to obtain the temperature at the lowest model elevation.

Table 12. Initialization profile used in GRAMM for test case E6

<i>Height [m]</i>	<i>Wind speed [ms^{-1}]</i>	<i>Wind direction</i>	<i>Temperature [K]</i>
-136	0	0	278.3
6	1.0	205	287.1
14	1.7	216	287.6
22	2.3	227	287.8
32	3.1	223	288.1
39	3.3	219	288.1
47	3.6	218	288.3
55	4.1	223	288.4
63	4.4	226	288.5
70	4.5	226	288.7
79	5.1	230	288.8
87	5.4	234	289.1
104	5.4	236	289.5
120	5.5	242	289.8
135	5.5	244	289.6
152	5.3	248	289.7
170	4.7	250	289.6
194	4.5	257	289.5
220	3.9	264	289.3
249	3.7	271	289.1
282	4.0	267	288.9
328	4.1	274	288.5
429	4.7	279	287.7
531	5.2	278	286.9
676	7.5	283	285.8
821	5.2	285	285.4
1016	5.5	272	283.5

By default the flags in the GRAMM input file IIN.dat in line 18 to compute surface temperatures (TB) and radiation (STR) are equal to one. Though, in this specific test case it would have been required to keep the surface temperature constant, the radiation and surface temperature were computed in order to keep the stable stratification of the initial temperature profile as good as possible. Without surface energy balance, the temperature profile starts quickly to erode due to vertical diffusion.

Simulation time is 1 hour (=3600 s; see line 4 in IIN.dat). The date and time of the simulation start has no influence on the results as the radiation model is not invoked.

Fig. 34. Screen shot of the GRAMM input file IIN.dat

```

COMPUTATION DATE          (YYMMDD)           : 080210
BEGINNING OF COMPUTATION  (hhmm)             : 0000
MAXIMUM ALLOWED TIME STEP DT [ s ]           : 10
MODELLING TIME (FOR VALUES >1(s) AND <1[%]) : 3600
BUFFERING AFTER TIMESTEPS : 3600
MAX. PERMISSIBLE W-DEVIATION ABOVE < 1 [mm/s] : 0.01
RELATIVE INITIAL HUMIDITY [ % ] GT.0         : 50
HEIGHT OF THE LOWEST COMPUTATION HEIGHT [ m ] : 330.
AIR TEMPERATURE AT GROUND [ K ]               : 290.0
GRADIENT OF THE POTENTIAL TEMPERATURE [K/100m]: -0.008
NEUTRAL LAYERING UP TO THE HEIGHT ABOVE GROUND: 10000
SURFACE TEMPERATURE [ K ]                    : 280.0
TEMPERATURE OF THE SOIL IN 1 M DEPTH [ K ]    : 280.0
LATITUDE : 55
UPDATE OF RADIATION (TIMESTEPS) : 10
DEBUG LEVEL 0 none, 3 highest : 0
COMPUTE U V W PN T GW FO YES=1 : 1111100
COMPUTE BR PR QU PSI TE TB STR YES=1 : 1110011
DIAGNOSTIC INITIAL CONDITIONS YES=1 : 1
EXPLICIT/IMPLICIT TIME INTEGRATION I=1/E=0 : 1
RELAXATION VELOCITY : 0.15
RELAXATION SCALARS : 0.15
Force catab flows -40/-25/-15 W/m2 AKLA 7/6/5 : 0
NESTING (=1) OR LARGE SCALE FORCING (=0) : 0
BOUNDARY CONDITION (1,2,3,4,5,6) : 6
NON-STEADYSTATE/FORCING W. 'METEO.ALL' YES=1 : 0
NON-STEADYSTATE/FORCING W. IFU-MM5 DATA YES=1 : 0
GRAMM IN GRAMM NESTING YES=1 : 0
Flowfield Output Format : 0

```

Fig. 35. Screen shot of the GRAMM input file GRAMMein.txt

```

1 n
2 y
3 Sophie_Gelaendefolgend.txt
4 n
5 n
6 99000
7 100000
8 .1
9 .01
10 .1
11 .000001
12 1
13 1

```

5.3.6.4 Results

The guideline requires that the mean absolute error (*MAE*) is lower than 0.6 m s^{-1} and 13° , and that the root mean square error (*RMSE*) is lower than 0.9 m s^{-1} and 19° . Computed wind speeds and –directions with GRAMM are listed in Table 13. The maximum *MAE* are 0.14 m s^{-1} and 5.5° , and the maximum *RMSE* are 0.24 m s^{-1} and 9.2° , respectively.

In addition, the VDI guideline requires that the difference in wind direction between the stations E6-1 and E6-3 is larger than 60° . GRAMM suggests 78° .

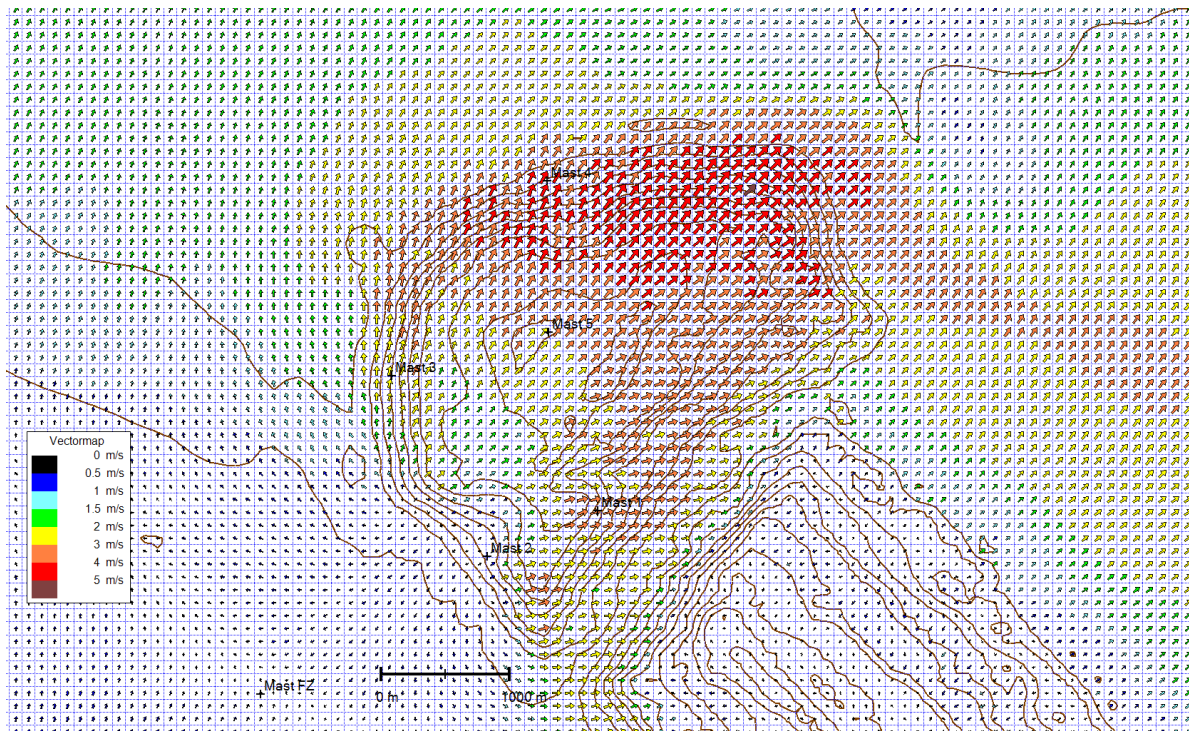
$$MAE = \frac{1}{N_R} \cdot \begin{cases} \sum_{i=1}^{N_R} |P_i - O_{\min}| & \text{if } P_i < O_{\min} \\ \sum_{i=1}^{N_R} |P_i - O_{\max}| & \text{if } P_i > O_{\max} \\ 0 & \text{if } O_{\min} \leq P_i \leq O_{\max} \end{cases} \quad (\text{E } 163)$$

$$RMSE = \begin{cases} \sqrt{\frac{1}{N_R} \sum_{i=1}^{N_R} (P_i - O_{\min})^2} & \text{if } P_i < O_{\min} \\ \sqrt{\frac{1}{N_R} \sum_{i=1}^{N_R} (P_i - O_{\max})^2} & \text{if } P_i > O_{\max} \\ 0 & \text{if } O_{\min} \leq P_i \leq O_{\max} \end{cases} \quad (\text{E } 164)$$

Table 13. Observed and modelled wind speeds and –directions, and deviations from the allowed range of observations (last two columns)

Station	<i>z(m)</i>	<i>OBS WS</i>		<i>OBS WD</i>		<i>GRAMM WS</i>	<i>GRAMM WD</i>	ΔWS	ΔWD	
N-E6-FZ1	50m	2.5	3.1	218	225	2.9	224	0.00	0	
N-E6-FZ2	80m	3.6	4.4	-	-	4.0		0.00		
N-E6-FZ2	100m	4.2	4.9	-	-	4.5		0.00		
N-E6-FZ3	120m	4.4	5.3	235	244	4.9	240	0.00	0	
N-E6-1	10m	1.7	2.2	280	282	2.8	260	0.60	20	
N-E6-2	7,5m	0.1	0.5	-	-	0.3		0.00		
N-E6-2	15m	0.2	0.6	283	322	0.3		0.00		
N-E6-3	10m	1.9	2.2	158	172	2.3	182	0.10	10	
N-E6-4	7,5m	3.3	3.4	-	-	3.9		0.50		
N-E6-4	15m	4.4	4.7	206	216	4.5	211	0.00	0	
N-E6-5	15m	3.5	4.1	226	234	3.2	223	0.30	3	
								MDE	0.14	5.5
								RMSE	0.25	9.2

Fig. 36. Flow field computed with GRAMM for test case E6 (lowest level)



5.3.7 Test case E7 (field data Grazer basin)

Here, model results are compared with field data observed during nighttime in the Graz basin in Austria.

5.3.7.1 Model domain and grid sizes

Size of prognostic domain: 14 x 20 km (no specific limit in the vertical direction)

Size of model domain: 50 x 61 x 10.6 km

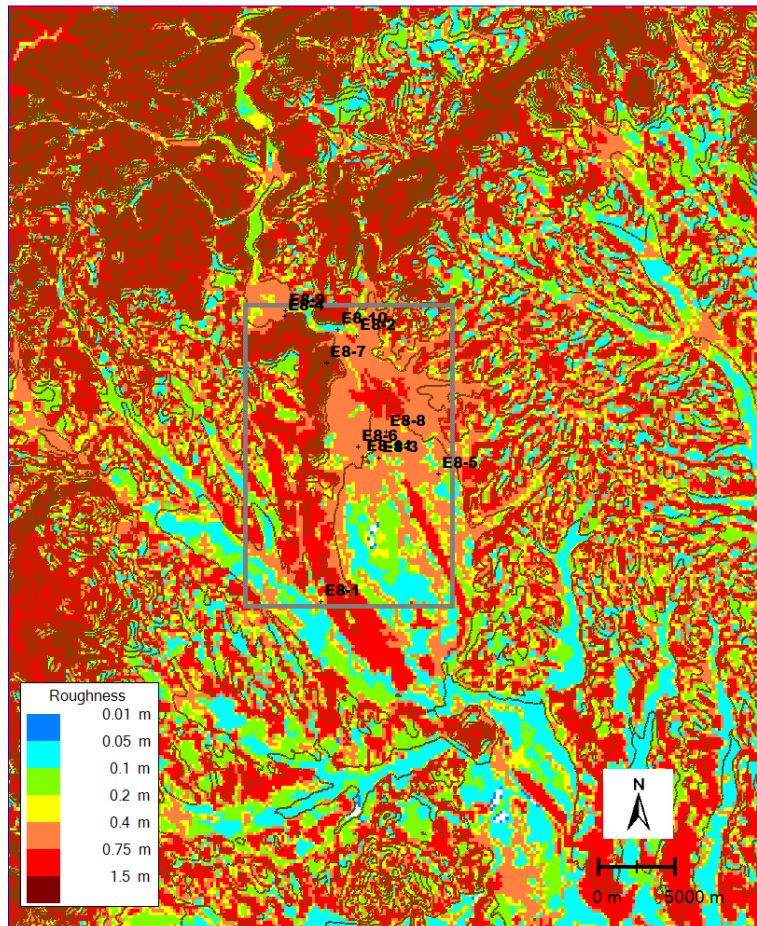
Horizontal grid size: 200 m

Vertical grid size, stretching factor, number of vertical grid cells: 10 m, 1.25, 25

5.3.7.2 Orography and land-use

Orographical and land-use data is provided by the VDI guideline.

Fig. 37. Orography, roughness length, and location of wind observations for test case E7



5.3.7.3 Initialization

Profiles for wind and temperature are provided by the guideline (see Table 14).

Table 14. Initialization profile used in GRAMM for test case E7

<i>Height [m]</i>	<i>Wind speed [ms^{-1}]</i>	<i>Wind direction</i>	<i>Temperature [K]</i>
0	0	304	274.7
10	1.5	304	275.1
20	2	304	275.5
40	3.2	304	275.8
60	3.2	304	276.0
80	3.3	305	276.1
100	3.5	306	276.2
120	3.7	306	276.4
140	3.9	309	276.4
160	4	311	276.5
180	4	313	277.3
200	3.9	316	277.5
220	3.7	319	278.0
240	3.4	322	278.5

260	3.2	325	279.1
280	2.9	327	279.4
300	2.6	330	279.5
320	2.5	331	279.9
340	2.3	331	280.9
360	2.2	328	280.6
380	2	325	280.7
400	1.9	321	280.6
420	1.7	315	280.4
440	1.6	310	280.3
460	1.5	305	280.2
480	1.5	302	280.1
500	1.4	299	279.8
520	1.4	298	279.7
540	1.5	297	279.5
560	1.5	297	279.4
580	1.5	296	279.2
600	1.5	296	279.1
620	1.5	294	279.0
928	3.6	215	278.7
2500	3.1	85	272.2
5090	5.7	10	252.5
6700	6.2	345	240.5
8660	5.1	360	224.7
9830	7.2	335	215.7

By default the flags in the GRAMM input file IIN.dat in line 18 to compute surface temperatures (TB) and radiation (STR) are equal to one. In this specific test case it is required to keep the surface temperature constant, thus, both flags need to be set to zero as can be seen from Fig. 38.

Simulation time is 1 hour (=3600 s; see line 4 in IIN.dat). The date and time of the simulation start has no influence on the results as the radiation model is not invoked.

Fig. 38. Screen shot of the GRAMM input file IIN.dat

```

COMPUTATION DATE      (YYMMDD)      : 080210
BEGINNING OF COMPUTATION (hhmm)      : 0000
MAXIMUM ALLOWED TIME STEP DT [ s ]    : 10
MODELLING TIME (FOR VALUES >1(s) AND <1[%]) : 3600
BUFFERING AFTER TIMESTEPS              : 3600
MAX. PERMISSIBLE W-DEVIATION ABOVE < 1 [mm/s] : 0.01
RELATIVE INITIAL HUMIDITY [ % ] GT.0   : 20
HEIGHT OF THE LOWEST COMPUTATION HEIGHT [ m ] : 330.
AIR TEMPERATURE AT GROUND [ K ]        : 280.0
GRADIENT OF THE POTENTIAL TEMPERATURE [K/100m]: -0.01
NEUTRAL LAYERING UP TO THE HEIGHT ABOVE GROUND: 4000
SURFACE TEMPERATURE [ K ]              : 270.0
TEMPERATURE OF THE SOIL IN 1 M DEPTH [ K ] : 280.0
LATITUDE                      : 47
UPDATE OF RADIATION (TIMESTEPS)        : 300
DEBUG LEVEL 0 none, 3 highest          : 0
COMPUTE U V W PN T GW FO              YES=1 : 1111100
COMPUTE BR PR QU PSI TE TB STR        YES=1 : 1110000
DIAGNOSTIC INITIAL CONDITIONS          YES=1 : 1
EXPLICIT/IMPLICIT TIME INTEGRATION I=1/E=0 : 1
RELAXATION VELOCITY                   : 0.15
RELAXATION SCALARS                    : 0.15
Force cat flows -40/-25/-15 W/m2 AKLA 7/6/5 : 0
NESTING (=1) OR LARGE SCALE FORCING (=0) : 0
BOUNDARY CONDITION (1,2,3,4,5,6)       : 6
NON-STEADYSTATE/FORCING W. 'METEO.ALL' YES=1 : 0
NON-STEADYSTATE/FORCING W. IFU-MM5 DATA YES=1 : 0
GRAMM IN GRAMM NESTING YES=1           : 0
Flowfield Output Format                   : 0

```

Fig. 39. Screen shot of the GRAMM input file GRAMMein.txt

```

1 n
2 y
3 dategraz
4 n
5 n
6 99000
7 101300
8 .1
9 .01
10 .1
11 .000001
12 1
13 1

```

5.3.7.4 Results

The guideline requires that the mean absolute error (*MAE*) is lower than 0.9 m s^{-1} and 18° , and that the root mean square error (*RMSE*) is lower than 1.3 m s^{-1} and 25° . Computed wind speeds and –directions with GRAMM are listed in Table 15. The maximum *MAE* are 0.62 m s^{-1} and 11.3° , and the maximum *RMSE* are 1.1 m s^{-1} and 21.7° , respectively.

In addition the VDI guideline requires that the wind direction at site E7-3 between 10 m and 60 m falls within $110 - 250^\circ$, which is the case in the GRAMM simulations. In addition, the wind speed needs to be lower than 1.5 m s^{-1} , which is also fulfilled. Furthermore, near the surface in the vicinity of station E7-3 southerly and at an elevation of 150 m northwesterly flows shall be modelled, which is also the case (see Fig. 40 and Fig. 41).

Compliance with the German Guideline VDI 3783 - 7 - Model validation

Table 15. Observed and modelled wind speeds and –directions, and deviations from the allowed range of observations (last two columns). Some stations are differently numbered in the VDI guideline. The VDI numbers are indicated in brackets.

<i>Station</i>	<i>Height [m]</i>	<i>Obs</i>	<i>Obs</i>	<i>Obs</i>	<i>Obs</i>	<i>Weight</i>	<i>GRAMM</i>	<i>GRAMM</i>	ΔWG	ΔWR
N-E7-8(9)	60	0	0.5			1	0.24	225	0.00	
N-E7-6(7)	72	0.2	1.5	169	199	1	0.79	170	0.00	0
N-E7-7(8)	35	2.1	3.3	293	316	1	1.48	266	0.62	27
N-E7-5	15	0	0.5			1	0.89	166	0.39	
N-E7-9(10)	10	0.4	1	199	309	1	0.44	328	0.00	19
N-E7-11(12)	30	1.2	1.4	116	142	1	1.57	138	0.17	0
N-E7-10(11)	12	4.7	5.7	258	270	1	2.58	267	2.12	0
N-E7-1	60	3.6	3.8	296	319	0.3	2.47	300	1.13	0
N-E7-1	90	4.4	4.5	251	256	0.3	2.82	306	1.58	50
N-E7-1	140	4.2	4.6	295	306	0.3	2.84	309	1.36	3
N-E7-4	20	0.5	1.0	241	298	0.14	1.08	359	0.10	61
N-E7-4	40	1.0	1.3	272	326	0.14	0.76	340	0.20	14
N-E7-4	60	1.1	1.5	289	335	0.14	0.53	310	0.61	0
N-E7-4	80	0.8	1.4	301	354	0.14	0.54	283	0.27	19
N-E7-4	100	0.9	1.6	255	343	0.14	0.63	268	0.25	0
N-E7-4	145	1.3	1.8	271	319	0.14	0.85	279	0.48	0
N-E7-4	295	3.3	3.6	296	330	0.14	1.44	290	1.91	6
N-E7-3	20	0.3	0.7	120	208	0.17	1.47	143	0.82	0
N-E7-3	40	0.7	1.3	104	194	0.17	1.15	149	0.00	0
N-E7-3	65	0.9	1.3	47	161	0.17	0.49	186	0.38	25
N-E7-3	145	1.9	3.3	305	320	0.17	1.71	279	0.20	26
N-E7-3	215	5.6	6.7	315	325	0.17	2.23	289	3.40	26
N-E7-3	285	3.6	4.9	301	321	0.17	2.35	292	1.29	9
MDE									0.62	11.3
RMSE									1.13	21.7

Fig. 40. Surface flow field computed with GRAMM for test case E7

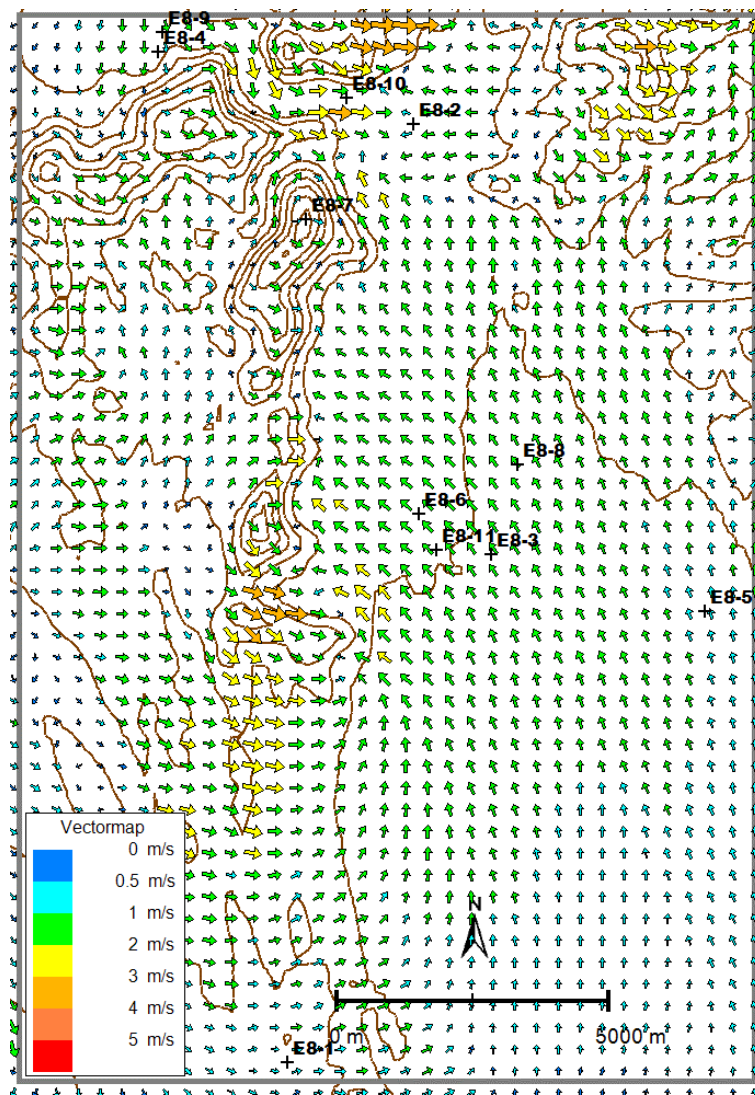
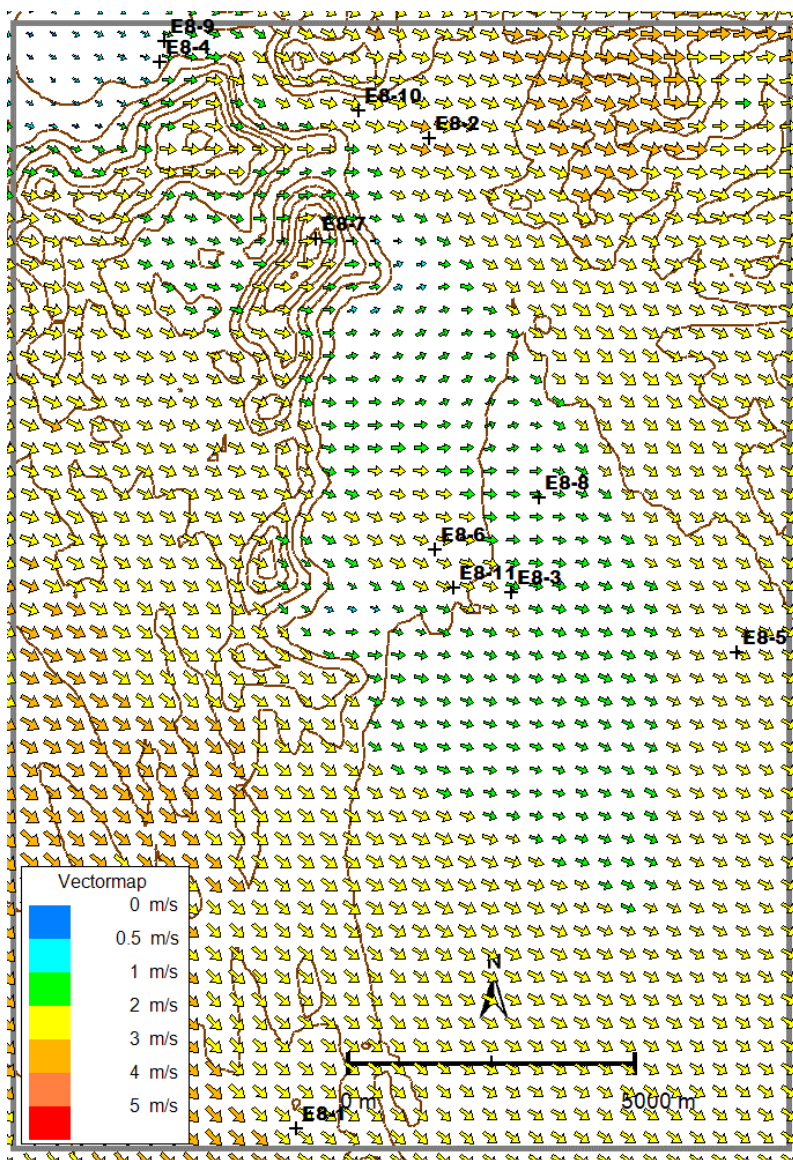


Fig. 41. Flow field 150 m above ground level computed with GRAMM for test case E7



5.3.8 Test case E8 (field data Stuttgart basin)

Here, model results are compared with tethered balloon soundings observed during nighttime in the Stuttgart basin in Germany.

5.3.8.1 Model domain and grid sizes

Size of prognostic domain: 10 x 10 km (no specific limit in the vertical direction)

Size of model domain: 30 x 30 x 5.5 km

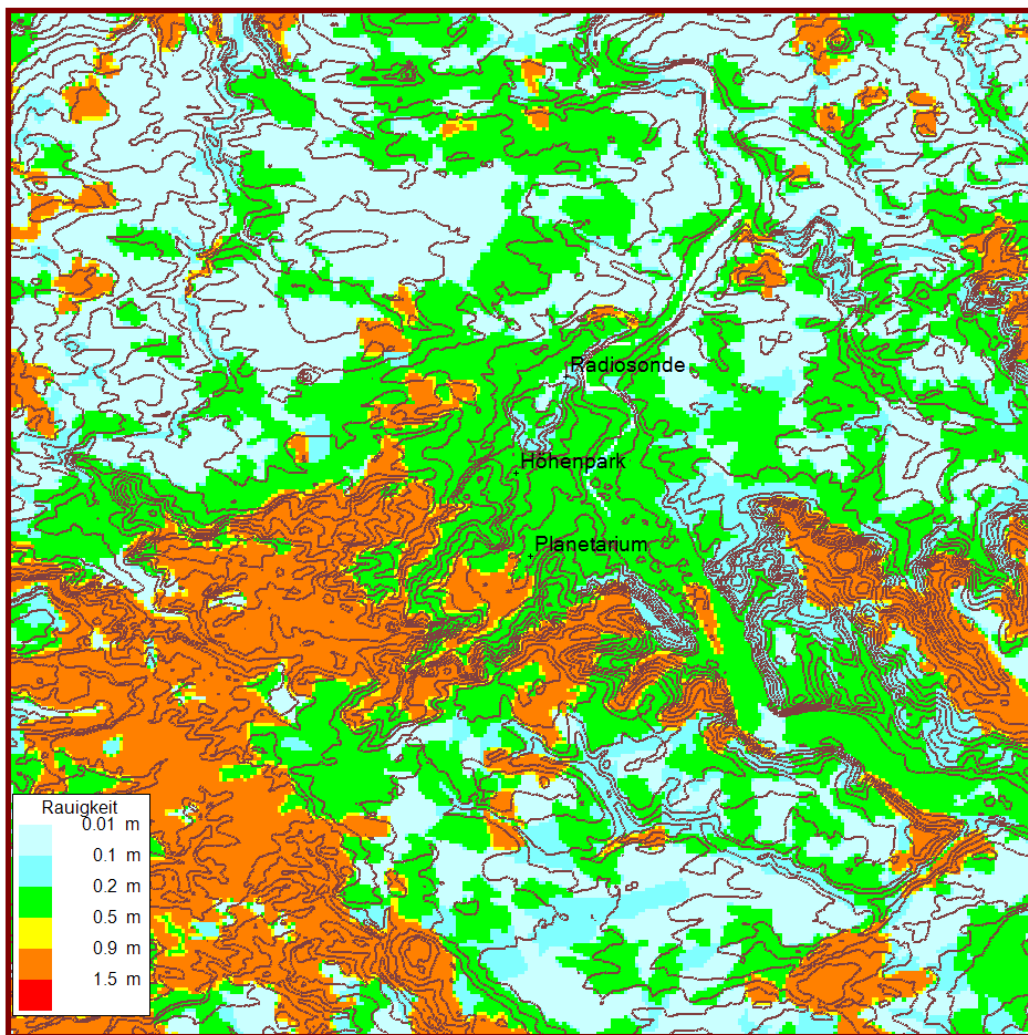
Horizontal grid size: 100 m

Vertical grid size, stretching factor, number of vertical grid cells: 7 m, 1.18, 30

5.3.8.2 Orography and land-use

Orographical and land-use data is provided by the VDI guideline.

Fig. 42. Orography, roughness length, and location of wind observations for test case E8



5.3.8.3 Initialization

Profiles for wind and temperature are provided by the guideline (see Table 16).

Table 16. Initialization profile used in GRAMM for test case E8

Height [m]	Wind speed [m s^{-1}]	Wind direction	Temperature [K]
10	2.2	47	289.6
30	2.4	48	289.4
50	2.5	49	289.2
70	2.6	49	289.0
90	2.6	49	288.8
110	2.6	50	288.6
130	2.7	50	288.4
170	2.7	50	288.0
230	2.7	51	287.5
320	2.8	51	286.7
420	2.8	52	285.7
570	2.8	52	284.2

Compliance with the German Guideline VDI 3783 - 7 - Model validation

640	2.8	53	283.5
790	2.8	53	282.1
1340	2.8	54	278.5
1455	2.8	54	277.8
1620	2.8	54	276.7
2220	2.8	54	272.8
2530	2.8	54	270.8
2890	2.8	54	268.5
3980	2.8	54	261.2
4320	2.8	54	259.1
5470	2.8	54	251.2
5990	2.8	54	247.7
6855	2.8	54	241.6
7850	2.8	54	234.5
9040	2.8	54	225.7
9960	2.8	54	219.0
10440	2.8	54	216.9

Simulation time is 5,5 hours (=19,800 s; see line 4 in IIN.dat). The date is set to 1 April and the time is set that sunset takes place about 2 1/4 hours after the simulation begin. The radiation and surface model is invoked. Deep soil temperature is set to 278 K (valid for the lowest elevation within the model domain which is 200 m), the height gradient is fixed in GRAMM and set to 0.005 K m^{-1} . Surface temperature is set 1 K lower than the temperature of the atmosphere in the lowest grid level. This is achieved in GRAMM by setting the value for "Air temperature at ground" 1 K above those for "Surface temperature" in the file IIN.dat (for information visit chap. 9.4.1.8).

Fig. 43. Screen shot of the GRAMM input file IIN.dat

```

COMPUTATION DATE      (YYMMDD)      : 970401
BEGINNING OF COMPUTATION (hhmm)      : 1600
MAXIMUM ALLOWED TIME STEP DT [ s ]    : 10
MODELLING TIME (FOR VALUES >1(s) AND <1[%]) : 19800
BUFFERING AFTER TIMESTEPS              : 3600
MAX. PERMISSIBLE W-DEVIATION ABOVE < 1 [mm/s] : 0.01
RELATIVE INITIAL HUMIDITY [ % ] GT.0   : 47
HEIGHT OF THE LOWEST COMPUTATION HEIGHT [ m ] : 180.
AIR TEMPERATURE AT GROUND [ K ]        : 282.0
GRADIENT OF THE POTENTIAL TEMPERATURE [K/100m]: -0.008
NEUTRAL LAYERING UP TO THE HEIGHT ABOVE GROUND: 10000
SURFACE TEMPERATURE [ K ]              : 281.0
TEMPERATURE OF THE SOIL IN 1 M DEPTH [ K ] : 280.0
LATITUDE                  : 49
UPDATE OF RADIATION (TIMESTEPS)        : 50
DEBUG LEVEL 0 none, 3 highest           : 0
COMPUTE U V W PN T GW FO               YES=1 : 1111100
COMPUTE BR PR QU PSI TE TB STR         YES=1 : 1110011
DIAGNOSTIC INITIAL CONDITIONS           YES=1 : 1
EXPLICIT/IMPLICIT TIME INTEGRATION I=1/E=0 : 1
RELAXATION VELOCITY                    : 0.15
RELAXATION SCALARS                     : 0.15
Force calab flows -40/-25/-15 W/m2 AKLA 7/6/5 : 0
NESTING (=1) OR LARGE SCALE FORCING (=0) : 0
BOUNDARY CONDITION (1,2,3,4,5,6)        : 6
NON-STEADYSTATE/FORCING W. 'METEO.ALL' YES=1 : 0
NON-STEADYSTATE/FORCING W. IFU-MM5 DATA YES=1 : 0
GRAMM IN GRAMM NESTING YES=1           : 0
Flowfield Output Format                  : 0

```

Fig. 44. Screen shot of the GRAMM input file GRAMMein.txt

```

1 n
2 y
3 Stuttgart.txt
4 n
5 n
6 99600
7 100000

```

5.3.8.4 Results

The guideline requires that the integrated volume flux at site Planetarium up to the height of the wind direction shift from southwest to northeast falls in the range $90 - 195 \text{ m}^{-3} \text{ m}^{-1} \text{ s}^{-1}$. The average volume flux should be evaluated after 4.5 hours and 5.5 hours modelling time. GRAMM suggests a volume flux of $143 \text{ m}^{-3} \text{ m}^{-1} \text{ s}^{-1}$. Furthermore, the height of the southwesterly flow needs to be in the range $85 - 160 \text{ m}$. The height modelled in GRAMM is 95 m . Between 25 m and 65 m the wind direction must be within 174° and 221° . GRAMM computed $193 - 204^\circ$.

Fig. 45. Observed and modelled wind speed and –direction

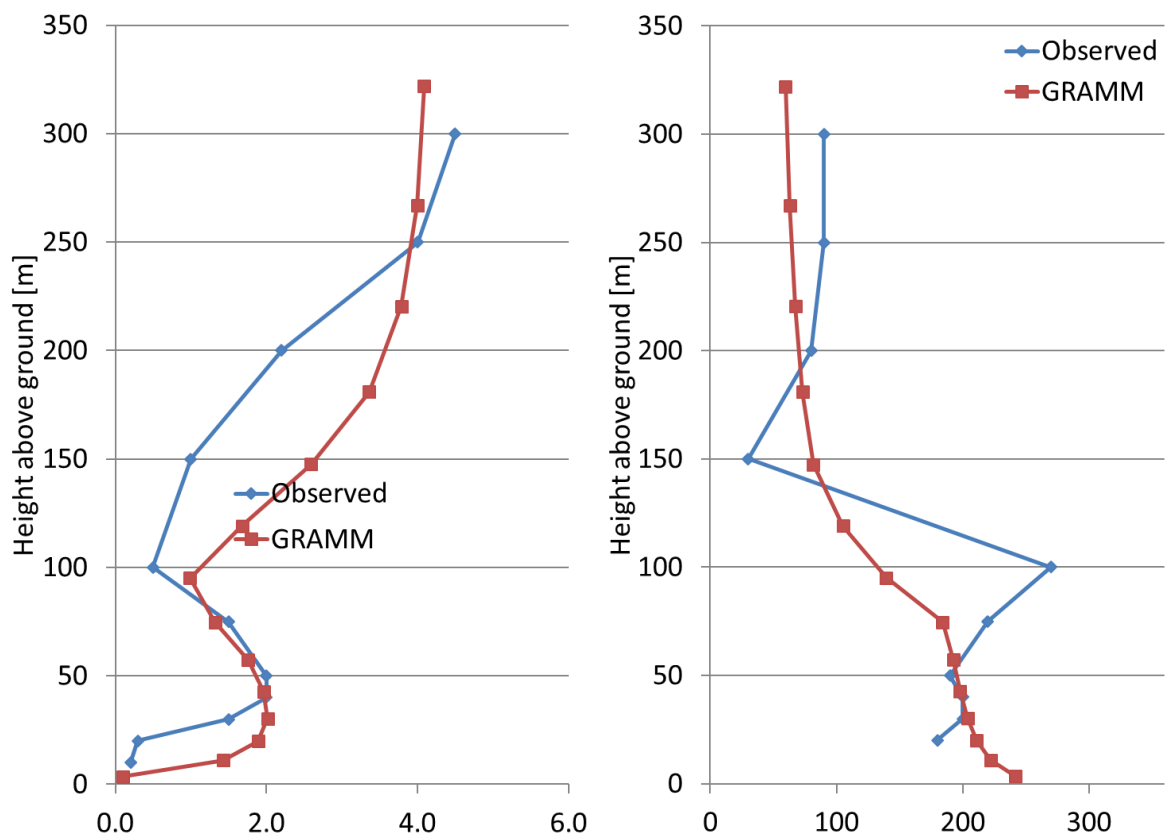
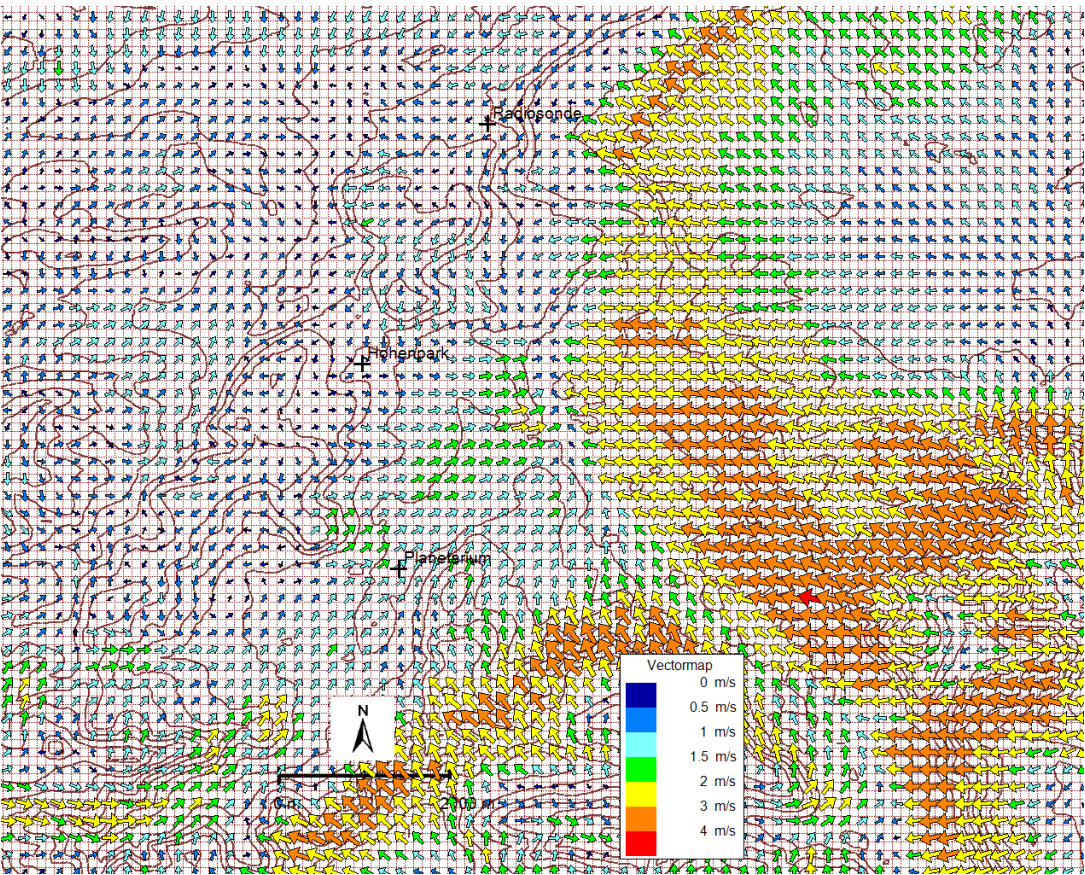


Fig. 46. Surface flow field as computed with GRAMM for test case E8



6 Additional validation cases

6.1 Test case boundary layer

Here, the ability of the model to simulate a boundary layer in one dimension is evaluated. Furthermore, it is tested if model results are invariant with regard to wind direction.

6.1.1.1 Model domain and grid sizes

Number of grid points: 3 x 3 x 30

Horizontal grid size: 100 m

Vertical grid size, stretching factor: 20 m, 1.16

6.1.1.2 Orography and land-use

Flat terrain is assumed. The roughness length varies for the different test cases.

6.1.1.3 Initialization

The potential temperature gradient varies for the different test cases as well as the latitude. The radiation and soil models are not invoked in this test case, therefore, the corresponding flags in the file IIN.dat need to be set to zero as can be seen from Fig. 47.

Simulation time is 5 hours (=18,000 s; see line 4 in IIN.dat). The date and time of the simulation start has no influence on the results as the radiation model is not invoked.

It should be stressed that the development of a boundary-layer in the GRAMM model, when initialized with a vertical constant large-scale wind speed, is only possible when homogeneous Neumann boundary-conditions are set (i.e. boundary condition 1 in line 25 in the file IIN.dat). Otherwise the boundary values are forced towards the large-scale values preventing a complete boundary-layer development in this specific case, where only one vertical column is used to solve the conservation equations.

Additional validation cases - Test case boundary layer

Fig. 47. Screen shot of the GRAMM input file IIN.dat

```

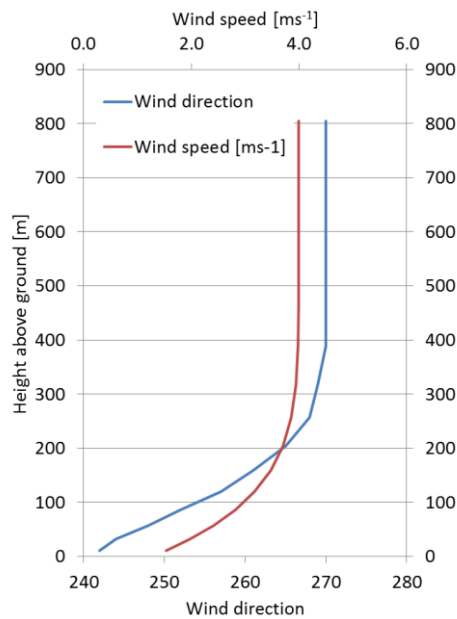
1 COMPUTATION DATE (YYMMDD) : 080210
2 BEGINNING OF COMPUTATION (hhmm) : 0000
3 MAXIMUM ALLOWED TIME STEP DT [ s ] : 10
4 MODELLING TIME (FOR VALUES >1(s) AND <1[%]) : 18000
5 BUFFERING AFTER TIMESTEPS : 72000
6 MAX. PERMISSIBLE W-DEVIATION ABOVE < 1 [mm/s] : 0.01
7 RELATIVE INITIAL HUMIDITY [ % ] GT.0 : 0.5
8 HEIGHT OF THE LOWEST COMPUTATION HEIGHT [ m ] : 330.
9 AIR TEMPERATURE AT GROUND [ K ] : 283.0
10 GRADIENT OF THE POTENTIAL TEMPERATURE [K/100m]: 0.01
11 NEUTRAL LAYERING UP TO THE HEIGHT ABOVE GROUND: 10000
12 SURFACE TEMPERATURE [ K ] : 283.0
13 TEMPERATURE OF THE SOIL IN 1 M DEPTH [ K ] : 283.0
14 LATITUDE : 50
15 UPDATE OF RADIATION (TIMESTEPS) : 10000
16 DEBUG LEVEL 0 none, 3 highest : 0
17 COMPUTE U V W PN T GW FO YES=1 : 1111000
18 COMPUTE BR PR QU PSI TE TB STR YES=1 : 1100000
19 DIAGNOSTIC INITIAL CONDITIONS YES=1 : 1
20 EXPLICIT/IMPLICIT TIME INTEGRATION I=1/E=0 : 1
21 RELAXATION VELOCITY : 0.9
22 RELAXATION SCALARS : 0.9
23 Force catag flows -40/-25/-15 W/m2 AKLA 7/6/5 : 0
24 NESTING (=1) OR LARGE SCALE FORCING (=0) : 0
25 BOUNDARY CONDITION (1,2,3,4,5,6) : 1
26 NON-STEADYSTATE/FORCING W. 'METEO.ALL' YES=1 : 0
27 NON-STEADYSTATE/FORCING W. IFU-MM5 DATA YES=1 : 0
28 GRAMM IN GRAMM NESTING YES=1 : 0
29 Flowfield Output Format : 0

```

6.1.1.4 Results

In a first step, the model has been initialized with a vertical constant large-scale wind speed of 4 m s^{-1} and a wind direction of 270° . Latitude was set to 50 deg. N, and the roughness length is 0.1 m. Fig. 48 depicts the simulated boundary layer with GRAMM. As expected, the wind speed decreases near the surface due to the surface friction. Due to the Coriolis force a wind-direction shift (i.e. Ekman spiral) of 28 deg. is modelled. The shift is towards the left-hand side in flow direction, which is to be expected on the northern hemisphere. A similar simulation for 50 deg. S gives a wind-direction shift of 29 deg. in the opposite direction indicating a correct implementation of the Coriolis force.

Fig. 48. Simulated wind speed and wind direction with the GRAMM model



In a second step, the calculated friction velocity with GRAMM is compared with the theoretical value obtained with the equation for friction velocity for neutral conditions:

$$u_* = \kappa v_1 \cdot \left[\ln \left(\frac{z}{z_0} \right) \right]^{-1} \quad (\text{E } 165)$$

The simulated wind speed of GRAMM in the lowest grid level v_1 is 1.53 m s^{-1} , which gives a friction velocity of 0.116 m s^{-1} for the utilized roughness length of 0.1 m . The simulated friction velocity of GRAMM is slightly higher and reads 0.120 m s^{-1} . A second simulation utilizing a roughness length of 0.5 m results in a theoretical value of 0.129 m s^{-1} and a modelled value of 0.133 m s^{-1} .

In a third step, it is tested whether the model simulates identical boundary layers regardless the large-scale wind direction. The wind speed and wind-direction shift at the lowest GRAMM grid layer are compared. Results are listed in Table 17. As can be seen the maximum difference in wind speed is 0.027 m s^{-1} and in wind direction 1.8 deg pointing towards slight numerical inaccuracies in the model.

Table 17. Simulated wind direction shifts and wind speeds in the lowest GRAMM grid level for different large-scale wind directions for a latitude of 50 deg. north

Large-scale wind direction	West	South	North	East
----------------------------	------	-------	-------	------

Additional validation cases - Test case boundary layer

Wind direction shift	30.1	31.8	31.7	28.1
Wind speed 10 m a.g.l.	1.621	1.642	1.647	1.528

7 References

- Almbauer, R.A., 1995: A new finite volume discretisation for solving the Navier-Stokes-equations. *Numerical Methods in Laminar and Turbulent Flow*, 9, 286 – 295
- Apsley, D.D., 2016: Turbulence modelling. Lecture notes University of Manchester, <http://personalpages.manchester.ac.uk/staff/david.d.apsley/lectures/comphydr/turbmodel.pdf>
- Arakawa, A., and V.R. Lamb, 1977: *Methods of Computational Physics*. New York : Academic Press, 17, pp 173-265
- Blackadar, A.K., J. A. Dutton, H. A. Panofsky, and A. Chaplin, 1969: Investigations of the Turbulent Wind Field below 150m Altitude at the Eastern Test Range. NASA Contractor Report, NASA CR-1410
- Boussinesq, J., 1897: *Theorie de l'ecoulement Turbillonnant et Tumultueux des Liquides Dans les lits Rectilignes a Grande Section*, Gauthier-Villars
- Burns, S.P., T.W. Horst, P.D. Blanken, and R.K. Monson, 2012: Using sonic anemometer temperature to measure sensible heat flux in strong winds. *Atmos. Meas. Tech. Discuss.*, 5, 447-469
- Businger, J.A., J.C. Wyngaard, Y. Izumi, and E.F. Bradley, 1971: Flux-profile relationships in the atmospheric surface layer. *J. Atmos. Sci.*, 28, 181-189
- Dutton, J. A., and G. H. Fichtel, 1969: Approximate equations of motion for gases and liquids. *J. Atmos. Sci.*, 26, 241 – 254
- Flassak, Th., 1990: Ein nicht-hydrostatisches mesoskaliges Modell zur Beschreibung der Dynamik der planetaren Grenzschicht. *VDI-Fortschrittsberichte Nr. 74, Reihe 15*, 203 pp
- Gauss, C.F., 1813: *Theoria attractionis corporum sphaeroidicorum ellipticorum homogeneorum method nova tractata*. *Commentationes societatis regiae scientiarum Gottingensis recentiores*, 2, 355-378
- Goodin, W.R., J. M. Gregory, and J. H. Seinfeld, 1980: An Objective Analysis Technique for Constructing Three-Dimensional Urban-Scale Wind Fields. *J. Appl. Met.*, 19, 98 – 108
- Greuell, W., Knap. W.H., and Smeets, P.C. (1997): Elevational changes in meteorological variables along a mid-latitdude glacier during summer. *Journal of Geophysical Research* 102 (D22), 25941-54
- Hurley, P.J., 2005: The Air Pollution Model (TAPM) Version 3. Part1: Technical Description.

References - Test case boundary layer

- CSIRO Atmospheric Research Technical Paper 71, Australia, 57pp, ISBN 0643068910
- Ineichen, P., 2008: Conversion function between the Linke turbidity and the atmospheric water vapour and aerosol content. *Solar Energy*, 82, 1095-1097
- Launiainen, S., J. Rinne, J. Pumpanen, L. Kulmala, P. Kolari, P. Keronen, E. Siivola, T. Pohja, P. Hari, and T. Vesala, 2005: Eddy covariance measurements of CO₂ and sensible and latent heat fluxes during a full year in a boreal pine forest trunk-space. *Boreal Environment Research*, 10, 569-588
- Lindroth, A., M. Mölder, and F. Lagergren, 2010: Heat storage in forest biomass improves energy balance closure. *Biogeosciences*, 7, 301-313
- Moderow, U., 2010: Energy balance of forests with special consideration of advection. Ph.D. Technical Univ. Dresden, Germany, pp 53
- ON M9440, 1996: Dispersion of pollutants in the atmosphere – Calculation of ambient air concentrations and determination of stack heights. Austrian Standards (ON), Vienna, 29 pp
- Oettl, D., R.A. Almbauer, P.J. Sturm, M. Piringer, and K. Baumann, 2000: Analysing the nocturnal wind field in the city of Graz, *Atmos. Environ.*, 35, pp 379-387
- Oettl, D., 2014a: Recommendations when using the GRAL/GRAMM modelling system. Internal report delivered with the GRAMM/GRAL modelling system via <http://lampx.tugraz.at/~gral/>, pp 21
- Oettl, D., 2014b: Documentation of the Lagrangian Particle Model GRAL (Graz Lagrangian Model) Vs. 15.6, Government of Styria, Department of Air Quality Control, Rep.Nr. LU-08-14, pp 139
- Oettl, D., 2015: A multiscale modelling methodology applicable for regulatory purposes taking into account effects of complex terrain and buildings on the pollutant dispersion: a case study for an inner Alpine basin. *Environmental Science and Pollution Research*, 22 (22), pp 17860-17875
- Patankar, S.V., 1980: Numerical heat transfer and fluid flow. Hemisphere publishing corporation, Washington.
- Pielke, R.A., 1984: Mesoscale Meteorological Modeling. Academic Press, London.
- Somieski, F., 1988: Mesoscale Model Parameterizations for Radiation and Turbulent Fluxes at the Lower Boundary. DLR Oberpfaffenhofen, Inst. für Nachrichtentechnik und Erkundung, Institut für Physik der Atmosphäre, Oberpfaffenhofen, D-8031 Weißling/Obb., pp 39

US-EPA, 2000: Meteorological Monitoring Guidance for Regulatory Modeling Applications. EPA-454/R-99-005. Office of Air and Radiation. Office of Air Quality Planning and Standards. Research Triangle Park, NC 27711, pp 171

VDI 3783-8, 2012: Environmental meteorology – Turbulence parameters for dispersion models supported by measurement data. Kommission Reinhaltung der Luft im VDI und DIN, Postfach 101139, 40002 Düsseldorf.

VDI 3783-7, 2015: Environmental meteorology – Prognostic mesoscale wind field models. Evaluation for dynamically and thermally induced flow fields. Kommission Reinhaltung der Luft im VDI und DIN, Postfach 101139, 40002 Düsseldorf.

8 Acknowledgements

English corrections for part of the document by Dagny Ullmann are greatly acknowledged.

9 Appendix A

9.1 Startup parameters

The following parameters can optional be used when starting GRAMM from the console or terminal.

GRAMM console arguments:

```
'Working Directory' 'First Situation' 'Final Situation' 'Max. Time Step'  
'RelaxV' 'RelaxT'
```

or

```
'First Situation' 'Final Situation' 'Max. Time Step' 'RelaxV' 'RelaxT'
```

9.2 Topographic position and landform analysis

The algorithm is based on the work of Andrew D. Weiss (Ecoregional Data Management Team, The Nature Conservancy, Northwest Division, Seattle WA 98103) who originally developed the TPI methodology and concepts.

The landform analysis algorithm uses the slope and compares the elevation of each cell in the digital elevation map with the mean elevation of its adjacent cells and an annulus neighbourhood to categorize the landform.

The radius of the adjacent cells is set at 1.5 times the grid size. The inner radius of the annulus is 800 m by default, the outer radius 4000 m. Depending on the grid size and the topography of your study area, other values might be better to detect valleys and bassins. You can override the default values by creating a file “TPI_Settings.txt” in the computation folder (see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**).

The mean values of the original TPI grids should be ‘reasonably’ close to zero. The algorithm breaks, if t the mean values are too high (> 10 for the adjacent cells or > 20 for the annulus cells). In this case, a warning message is written to the console or terminal window.

The TPI values are normalized and classified according to their mean value and standard deviation to get the following landform classes:

STDI value	Description
1	V-shape river valleys
2	Lateral midslope incised drainages
3	Upland incised drainages
4	U-shape valleys
5	Broad Flat Areas

6	Broad open slopes
7	Flat ridge tops
8	Local ridge or hilltops within broad valleys
9	Lateral midslope drainage divides
10	Mountain tops

The algorithm writes two output files, called “TPI_SlopeMin.txt” and “TPI_STDI.txt” as ESRI ASCII files (see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** and **Fehler! Verweisquelle konnte nicht gefunden werden..**).

9.3 List of variables in the GRAMM code

In the following table the variables used in the GRAMM code are listed. Besides the name of the variables, the type and dimension(s) are also provided. One bracket indicates a one-dimensional array, two- and three brackets two- and three-dimensional arrays, respectively. According to the specified types of variables they have the following accuracy:

Table 18. Variable types as defined in C#

Type	Represents	Range	Default Value
bool	Boolean value	True or False	False
double	64-bit double-precision floating point type	(+/-)5.0 x 10 ⁻³²⁴ to (+/-)1.7 x 10 ³⁰⁸	0.0D
int32 or int	32-bit signed integer type	-2,147,483,648 to 2,147,483,647	0
Int 16	16-bit signed integer type	-32,768 to +32,767	0
String	Character value		

Table 19. List of variables used in the GRAMM code

Variable	Description
string decsep	global decimal separator of the system
Int32 NX	number of cells in x-direction
Int32 NY	number of cells in y-direction
Int32 NZ	number of cells in z-direction
Int32 NX1	number of cells in x-direction +1
Int32 NY1	number of cells in y-direction +1
Int32 NZ1	number of cells in z-direction +1
Int32 NX2	number of cells in x-direction +2

Appendix A - List of variables in the GRAMM code

Int32 NY2	number of cells in y-direction +2
Int32 NZ2	number of cells in z-direction +2
Int32 NZB	NUMBER OF VERTICAL GRID CELLS OF THE SOIL MODEL
Int32 NPROFMAX	NUMBER OF VERTICAL GRID CELLS OF THE RADIATION MODEL
Int16 IMETSTR	Flag needed that geometry data for the radiation model is read just once
Int16 IHOURO	Flag needed for the intermediate output of GRAMM flowfields
double[][] AH	Height of the surface
double[][][] VOL	Volume of grid cells
double[][][] AREAX	Area of the grid cell in x-direction
double[][][] AREAY	Area of the grid cell in y-direction
double[][][] AREAZ	Bottom area of the grid cell
double[][][] AREA	Projection of the ground area of the grid cell in z-direction
double[][][] AREAXYZ	Area between the two halves of the grid cell
double[][][] AREAZX	Projection of the ground area of the grid cell in x-direction
double[][][] AREAZY	Projection of the ground area of the grid cell in y-direction
double[][][] AHE	Heights of the corner points of each grid cell
double[][][] ZSP	Height of the centre point of each grid cell
double[] DDX	Horizontal grid size in x-direction
double[] DDY	Horizontal grid size in y-direction
double[] ZAX	Distance between neighbouring grid cells in x-direction
double[] ZAY	Distance between neighbouring grid cells in y-direction
double[] X	Distance of grid cell centre from eastern domain border
double[] Y	Distance of grid cell centre from southern domain border
double[] Z	Temporary array for generating the terrain-following grid
double[][] LAND	CORINE land-use categories
double[] PBZZ	Base state for pressure used in the radiation model
double[] RHOBZZ	Base state for density used in the radiation model
double[][][] TABS	Absolute temperature in K
double[][] RSOLG	Solar ingoing radiation
double[][][] DT_SOL	Solar heating rate
double[][] RL	Longwave outgoing radiation
double[][][] DT_TERR	Terrestrial heating rate
double[][] TG	Absolute soil temperature at the second level beneath the surface
double[][] EPSG	Surface emissivity
int[][] KST	Defines the cell in vertical direction from which onward the radiation model is computed
double[] ZZ	Average value of two neighbouring cell heights
double[][] ALBEDO	Surface albedo
double[] ZPROF	Cell heights for the radiation model
double[] TPROF	Vertical temperature profile in the radiation model
double[] PPROF	Vertical pressure profile in the radiation model
double[] VNORM	standard optical length
double[] QVAP	water content in the atmosphere used in radiation model
double[] QCLD	water content in clouds used in radiation model
double[] QRAIN	water content in rain used in radiation model
double[] QICE	water content in ice used in radiation model
double[] QSNOW	water content in snow used in radiation model

Appendix A - List of variables in the GRAMM code

double[][][] U	Average wind speed in x-direction over the two half-cells at time t
double[][][] V	Average wind speed in y-direction over the two half-cells at time t
double[][][] W	Average wind speed in z-direction over the two half-cells at time t
double[][][] RHO	Density of the air
double[][][] UN	Average wind speed in x-direction over the two half-cells at time t+1
double[][][] VN	Average wind speed in y-direction over the two half-cells at time t+1
double[][][] WN	Average wind speed in z-direction over the two half-cells at time t+1
double[][][] U1	Average wind speed in x-direction in the lower half-cells at time t
double[][][] V1	Average wind speed in y-direction in the lower half-cells at time t
double[][][] W1	Average wind speed in z-direction in the lower half-cells at time t
double[][][] U2	Average wind speed in x-direction in the upper half-cells at time t
double[][][] V2	Average wind speed in y-direction in the upper half-cells at time t
double[][][] W2	Average wind speed in z-direction in the upper half-cells at time t
double[][][] U1N	Average wind speed in x-direction in the lower half-cells at time t+1
double[][][] V1N	Average wind speed in y-direction in the lower half-cells at time t+1
double[][][] W1N	Average wind speed in z-direction in the lower half-cells at time t+1
double[][][] U2N	Average wind speed in x-direction in the upper half-cells at time t+1
double[][][] V2N	Average wind speed in y-direction in the upper half-cells at time t+1
double[][][] W2N	Average wind speed in z-direction in the upper half-cells at time t+1
double[][][] PN	Non-hydrostatic pressure in the cell-centre
double[][][] DP	Non-hydrostatic pressure in the cell-centre
double[][][] DPX	Non-hydrostatic pressure in x-direction
double[][][] DPY	Non-hydrostatic pressure in y-direction
double[][][] DPZ	Non-hydrostatic pressure in z-direction
double[][][] DDP1DX	Pressure gradient used for the velocity correction in x-direction
double[][][] DDP1DY	Pressure gradient used for the velocity correction in y-direction
double[][][] DDP1DZ	Pressure gradient used for the velocity correction in z-direction
double[][][] DDP2DX	Pressure gradient used for the velocity correction in x-direction
double[][][] DDP2DY	Pressure gradient used for the velocity correction in y-direction
double[][][] DDP2DZ	Pressure gradient used for the velocity correction in z-direction
double[][][] SUX	Mass divergence in x-direction
double[][][] SUY	Mass divergence in y-direction
double[][][] SUZ	Mass divergence in z-direction
double[][][] SUXYZ	Mass divergence in between the two half-cells
double[][][] U1NRHO	Mass-flux in x-direction
double[][][] U2NRHO	Mass-flux in y-direction
double[][][] V1NRHO	Mass-flux in z-direction
double[][][] V2NRHO	Mass-flux in x-direction
double[][][] W1NRHO	Mass-flux in y-direction
double[][][] W2NRHO	Mass-flux in z-direction
double[][][] PBZ	Hydrostatic pressure
double[][][] RHOBZ	Air density at the simulation start
double[][][] PNBZKP	Temporary field in the calculation of the radiation fluxes
double[][][] NBZKP	Temporary field in the calculation of the radiation fluxes
double[][][] UG1	Geostrophic wind in x-direction at time t-1
double[][][] VG1	Geostrophic wind in y-direction at time t-1

Appendix A - List of variables in the GRAMM code

double[][] UG2	Geostrophic wind in x-direction at time t+1
double[][] VG2	Geostrophic wind in y-direction at time t+1
double[][] UG	Geostrophic wind in x-direction at time t
double[][] VG	Geostrophic wind in y-direction at time t
double[][] VISV	Turbulent viscosity in z-direction
double[][] VISH	Turbulent viscosity in horizontal-directions
double[][] QU	Specific humidity at time t
double[][] QUN	Specific humidity at time t+1
double[][] QBZ	Basic state of specific humidity at the simulation start
double[] QUG	Soil moisture content
double[] FW	Specific soil moisture parameter (e.g. water = 1)
double[][] T	Potential temperature of air at time t
double[][] TN	Potential temperature of air at time t+1
double[][] FACTOR	Conversion factor between potential and absolute temperature of air
double[][] DISS	Turbulent dissipation rate at time t
double[][] DISSN	Turbulent dissipation rate at time t+1
double[][] TE	Turbulent kinetic energy at time t
double[][] TEN	Turbulent kinetic energy at time t+1
double[] ZI	Mixing height
double[] UST	Friction velocity
double[] USTV	Friction velocity normalized by the wind speed
double[] TST	Potential temperature scale
double[] OL	Obukhov-length
double[] XWQ	Temporary field used for calculating the sensible and latent heat fluxes
double[] ZO	Aerodynamic surface roughness
double[][] FAC	Conversion factor between potential and absolute temperature of air
double[][] RITSCH	Gradient Richardson number
double[] WQU	Sensible heat flux
double[] AWQ	Anthropogenic heat flux
double[][] TB	Absolute soil temperature at time t
double[][] TBN	Absolute soil temperature at time t+1
double[][] TBA	Absolute soil temperature at time t-1
double[][] TBZ	Basic state of potential temperature at the simulation start
double[] DWB	Vertical cell heights of the soil model
double[] DZB	Vertical distance between neighbouring cells of the soil model
double[] RHOB	Soil density
double[] ALAMBDA	Surface albedo
double[] ALPHA	Damping parameter used at the model boundaries
double[][] A_PS	Temporary field used in the TDMA solver for passive scalars
double[][] B_PS	Temporary field used in the TDMA solver for passive scalars
double[][] C_PS	Temporary field used in the TDMA solver for passive scalars
double[][] AWEST_PS	Temporary field used in the TDMA solver for passive scalars
double[][] ASOUTH_PS	Temporary field used in the TDMA solver for passive scalars
double[][] AEAST_PS	Temporary field used in the TDMA solver for passive scalars
double[][] AN-ORTH_PS	Temporary field used in the TDMA solver for passive scalars

Appendix A - List of variables in the GRAMM code

double[][][] AP0_PS	Temporary field used in the TDMA solver for passive scalars
double[][][] AIM	Temporary field used in the TDMA solver for velocities
double[][][] BIM	Temporary field used in the TDMA solver for velocities
double[][][] CIM	Temporary field used in the TDMA solver for velocities
double[][][] AW1	Temporary field used in the TDMA solver for velocities
double[][][] AS1	Temporary field used in the TDMA solver for velocities
double[][][] AE2	Temporary field used in the TDMA solver for velocities
double[][][] AN2	Temporary field used in the TDMA solver for velocities
double[][][] AP0	Temporary field used in the TDMA solver for velocities
double[][][] DIMU	Temporary field used in the TDMA solver for velocities
double[][][] DIMV	Temporary field used in the TDMA solver for velocities
double[][][] DIMW	Temporary field used in the TDMA solver for velocities
double[][][] AP	Temporary field used in the TDMA solver for non-hydrostatic pressure
double[][][] AB	Temporary field used in the TDMA solver for non-hydrostatic pressure
double[][][] AT	Temporary field used in the TDMA solver for non-hydrostatic pressure
double[][][] AW	Temporary field used in the TDMA solver for non-hydrostatic pressure
double[][][] AE	Temporary field used in the TDMA solver for non-hydrostatic pressure
double[][][] AS	Temporary field used in the TDMA solver for non-hydrostatic pressure
double[][][] AN	Temporary field used in the TDMA solver for non-hydrostatic pressure
double[][] VDATA1	Temporary field used for online output of prognostic variables
double[][] VDATA2	Temporary field used for online output of prognostic variables
double[][] VDATA3	Temporary field used for online output of prognostic variables
double[][] VDATA4	Temporary field used for online output of prognostic variables
double[][] VDATA8	Temporary field used for online output of prognostic variables
double[] VDATA5	Temporary field used for online output of prognostic variables
double[] VDATA6	Temporary field used for online output of prognostic variables
double[] VDATA7	Temporary field used for online output of prognostic variables
double[] VDATA9	Temporary field used for online output of prognostic variables
double[] MASSOURCE	Control variable for the temporal development of the entire mass divergence
double[][] U_TEMP	Control variable for checking flow convergence in the lowest layer
double[][] V_TEMP	Control variable for checking flow convergence in the lowest layer
double[][] W_TEMP	Control variable for checking flow convergence in the lowest layer
double[][] USE1	Large scale value of u-wind at eastern boundary at time t
double[][] VSE	Large scale value of v-wind at eastern boundary at time t
double[][] WSE	Large scale value of w-wind at eastern boundary at time t
double[][] TSE	Large scale value of pot. temp. at eastern boundary at time t
double[][] QUSE	Large scale value of spec. humidity at eastern boundary at time t
double[][] USEN	Large scale value of u-wind at eastern boundary at time t+1
double[][] VSEN	Large scale value of v-wind at eastern boundary at time t+1
double[][] WSEN	Large scale value of w-wind at eastern boundary at time t+1
double[][] TSEN	Large scale value of pot. temp. at eastern boundary at time t+1
double[][] QUSEN	Large scale value of spec. humidity at eastern boundary at time t+1
double[][] USW	Large scale value of u-wind at western boundary at time t
double[][] VSW	Large scale value of v-wind at western boundary at time t
double[][] WSW	Large scale value of w-wind at western boundary at time t
double[][] TSW	Large scale value of pot. temp. at western boundary at time t

Appendix A - List of variables in the GRAMM code

double[][] QUSW	Large scale value of spec. humidity at western boundary at time t
double[][] USWN	Large scale value of u-wind at western boundary at time t+1
double[][] VSWN	Large scale value of v-wind at western boundary at time t+1
double[][] WSWN	Large scale value of w-wind at western boundary at time t+1
double[][] TSWN	Large scale value of pot. temp. at western boundary at time t+1
double[][] QUSWN	Large scale value of spec. humidity at western boundary at time t+1
double[][] USN	Large scale value of u-wind at northern boundary at time t
double[][] VSN	Large scale value of v-wind at northern boundary at time t
double[][] WSN	Large scale value of w-wind at northern boundary at time t
double[][] TSN	Large scale value of pot. temp. at northern boundary at time t
double[][] QUSN	Large scale value of spec. humidity at northern boundary at time t
double[][] USNN	Large scale value of u-wind at northern boundary at time t+1
double[][] VSNN	Large scale value of v-wind at northern boundary at time t+1
double[][] WSNN	Large scale value of w-wind at northern boundary at time t+1
double[][] TSNN	Large scale value of pot. temp. at northern boundary at time t+1
double[][] QUSNN	Large scale value of spec. humidity at northern boundary at time t+1
double[][] USS	Large scale value of u-wind at southern boundary at time t
double[][] VSS	Large scale value of v-wind at southern boundary at time t
double[][] WSS	Large scale value of w-wind at southern boundary at time t
double[][] TSS	Large scale value of pot. temp. at southern boundary at time t
double[][] QUSS	Large scale value of spec. humidity at southern boundary at time t
double[][] USSN	Large scale value of u-wind at southern boundary at time t+1
double[][] VSSN	Large scale value of v-wind at southern boundary at time t+1
double[][] WSSN	Large scale value of w-wind at southern boundary at time t+1
double[][] TSSN	Large scale value of pot. temp. at southern boundary at time t+1
double[][] QUSSN	Large scale value of spec. humidity at southern boundary at time t+1
double[][] Alfa	Inclination angle of the surface fall-line (=line of greates slope)
double[][] Beta	Azimuth angle of fall-line (=line of greates slope)
double[][] Q	Projection on the surface in the radiation model
double[][] YG	Integrated visibility in the radiation model
double[][] R	Integrated water vapour in the radiation model
double[][] W1rad	? something in the radiation model
double[][] W2rad	? something in the radiation model
double[][] Wrad	? something in the radiation model
double[] Dz1	Lower cell height of the radiation model
double[] Dz2	Upper cell height of the radiation model
double[][] Ag	Surface albedo in the radiation model
double[] Tstern	Linke turbidity coefficient (~1 - 8)
double[] eS	Cloud parameter
double[] eSeS	Cloud parameter
double[] IG	Shortwave solar radiation
double[][][] EG	Total solar radiation density
double[][][] SG	Direct solar radiation density
double[][][] DG	Diffusive solar radiation density
double[] nS	Transmission function
double[] nD	Transmission function

Appendix A - List of variables in the GRAMM code

double[] Tau	Temporary field for cloud
double[][][] Arad	Temporary field in the radiation model
double[] Myx	Temporary field in the radiation model
double[][] Rp	Integrated water vapour
double[][] CGp	Integrated CO2
double[] L_Strich	Downward terrestrial radiation
double[][] RTerrG	Total incoming terrestrial radiation
double[] EpsAp	Temporary field in the radiation model
double[] EpsAm	Temporary field in the radiation model
double[] Tab	Temperature of the black cloud (at top of the atmosphere)
double[] Eab	Emissivity? of the black cloud (at top of the atmosphere)
double ATTENU	= 2; Constant used for nudging model values towards large-scale values at the lateral boundaries
double B0	= 38.00699; Constant used for the computation of the saturation temperature
double B1	= 6,546.307; Constant used for the computation of the saturation temperature
double B2	= -3.257572; Constant used for the computation of the saturation temperature
double B3	= -0.003163528; Constant used for the computation of the saturation temperature
Int32 NDATUM	Starting date of computation
Int32 ISTUNDE	Starting hour of computation
double DT	Maximum allowed time step
double TLIMIT2	Total modelling time
Int32 IOUTPUT	Write intermediate output files after IOUTPUT seconds
double DELW	Historic value not used currently
double QUINIT	Initial relative humidity
double ZSEEH	Height of the lowest computation height - not used anymore
double TINIT	Temperature at 2m height
double TGRAD	Gradient of potential temperature - not used anymore
double ZNEUT	Height of the neutral boundary layer - not used anymore
double TBINIT	Surface temperature
double TBINIT1	Soil temperature in 1m depth -> Note that TBINIT1, TBINIT and TINIT are not used by their absolute values but by their differences to define the initial state of the atmospheric and soil temperatures
double BGRAD	Latitude
Int32 IRAD	Number of timesteps after which the radiation is updated
Int32 IDEBUG	Debug level 0 none, 3 highest
Int32 IFLAGS1	Compute u v w p n t g w f o
Int32 IFLAGS2	Compute b r p r q u p s i t e t b s t r
double DIA	Flag to determine whether initial state of the model is computed using the diagnostic model
double DIAI	Not used anymore - former flag to switch between explicit and implicit solution algorithm
double RELAXV	Underrelaxation factor for momentum equations
double RELAXT	Underrelaxation factor for passive scalar equations
Int32 ICATAFORC	Forcing of catabatic flows with -40
Int32 INEST	NESTING (=1) OR LARGE SCALE FORCING (=0) - not yet implemented in this version
Int32 IBOUND	BOUNDARY CONDITION (1,6,10,11,12)
Int32 ISTAT	NON-STEADYSTATE
Int32 MM5IFU	NON-STEADYSTATE
Int32 NESTGRAMM	GRAMM IN GRAMM NESTING YES=1 -> not yet implemented in this version
Int32 IOUT	flow out as binary-stream, little_endian, with header 0, or 3 without header
double DTI	Total simulation time

Appendix A - List of variables in the GRAMM code

double DTMAX	Maximum allowed time step
Int32 IPROC	Maximum number of processors
Int32 IJAHR	Year of the simulation (no effect)
Int32 IMON	Month of the simulation start
Int32 ITAG	Day of the simulation start
Int32 ISTU	Hour of the simulation start
Int32 IMIN	Minute of the simulation start
Int32 IKOOA	Western border of model domain
Int32 JKOOA	Southern border of model domain
double AHMIN	Minimum surface elevation
double AHMAX	Maximum surface elevation
Int32 AHMINI	Cell indices of minimum surface elevation
Int32 AHMINJ	Cell indices of minimum surface elevation
Int32 irec	Total number of receptor points
Int32[] inrec	Index in x-direction of receptor points
Int32[] jnrec	Index in y-direction of receptor points
Int32[] knrec	Index in z-direction of receptor points
double[] Xrec	x-coordinate of receptor points
double[] Yrec	y-coordinate of receptor points
double[] Zrec	z-coordinate of receptor points
double[] Urec	u-component at receptor points
double[] Vrec	v-component at receptor points
double[] Trec	temperature at receptor points
double STRETCH	Stretching-factor in the vertical
double FN	Coriolis frequency normal to the earths surface
double FH	Coriolis frequency parallel to the earths surface
double[] PSAT	saturation pressure
double GERD	= 9.81; Gravitational acceleration
double GASCON	= 287; General gas constant
double CPLUFT	= 1,000; Heat capacity of air for constant pressure
double ALW	= 2,500,000; Evaporation heat of water
Int16 ISOL	Switch for radiation model: 1=thin clouds 2=thick clouds
Int16 IWETTER	Number of the actual calculated flow field situation
string METEO	Character determining whether GRAMM is driven by the file meteopgt.all or not
double ANEMO	Anemometer heat when using meteopgt.all as input file
Int16 AKLA	Stability class when using meteopgt.all as input file
double TLIMIT	Modified total simulation time in dependence on the stability class
double Obini	Initial Obukhov length
double Rauigkeit	Initial surface-roughness length
double CK	= 0.35; van Karman constant
double VISEL	= 0.5; minimum turbulent viscosity
double CPBOD	= 900; heat capacity of soil
double SIGMA	= 5.6697E-8; Stefan Boltzmann constant
double TBZ1	temperature value used to improve numerical accuracy of the solution algorithm for temperature
Int16 IPGT	Flag used to check whether the file meteopgt.all is used as input file
double TJETZT	specific modelling time format used for the radiation model

Appendix A - List of variables in the GRAMM code

double REALTIME	integrated modelling time
Int16 ITIME	total time steps
Int16 INUMS	number of pressure-iterations
double DIVSUM, SUMG	total mass divergence
Int16 IDIV	counter for computing the trend of the mass-divergence used to adjust the time-step
double STEIGUNG	temporal trend of the total mass-divergence used to compute the actual time-step
double MAS- SOURCE_FIRST	initial total mass-divergence at the beginning of each computation
double PRTE	= 0.9; turbulent Prandtl-number
Boolean ICU	Flag switching the computation of the u-component on
Boolean ICV	Flag switching the computation of the v-component on
Boolean ICW	Flag switching the computation of the w-component on
Boolean ICPN	Flag switching the computation of the non-hydrostatic pressure on
Boolean ICT	Flag switching the computation of the pot. temperature on
Boolean ICPH	Flag switching the computation of the hydrostatic pressure on
Boolean IFOU	Flag switching the computation of the Fourier-Transformed upper boundary conditions on
Boolean ICQU	Flag switching the computation of the spec. humidity on
Boolean ICPSI	Flag switching the computation of a passive scalar on
Boolean ICTE	Flag switching the computation of the turbulent kinetic energy on
Boolean ICSTR	Flag switching the computation of the radiation model on
Boolean ICPR	Flag switching the computation of the surface layer model on
Boolean ICBR	Flag switching the computation of the boundary conditions on
Boolean ICTB	Flag switching the computation of the surface temperature on
Boolean ICGW	Flag switching the computation of the geostrophic winds from the large scale model when run in nesting mode on
Boolean reexist	Flag set when receptor points are set (file receptor_GRAMM.dat)
int Counter	= 0; Counter for writing "DispGRAMM.txt"
int TerminalOut	= 0; Counter for Terminal Output
int Na	Number of days since 1 Jan
double timeR	Number of days since 1 Jan
int KStMax	Vertical index of the highest surface cell
double Eps	= 0.00005; General lower threshold
int nOmega	= 40; Number of sectors used to compute the horizons of the surroundings (should be divisible by 4)
double GPhi	Variation of the solar flux over the year
double Mye	Sinus of sun height (He)
double My	Sinus of sun height (H)
double CosHe	Cosinus He(My)
double CosH	Cosinus He(My)
double He	Angle of sun height [rad]
double Eta	Azimuth of the sun [rad]
double GrRa	= 0.0174532925199; Conversion factor PI
double RaGr	= 57.2957795131; Conversion factor 180
double p0	= 1.01325E5; reference pressure
double eHoch1	= 2.7182818285; Eulerian constant
double IG0	= 1367; extraterrestrial solar radiation density [W
double cp	= 1004; heat capacity of air [Joule

Appendix A - Control files

double QCo2	= 0.00052; mass-concentration of CO2
double Wcrit	= 0.03; W-integral of black cloud [kg
double EpsTop	= 0.2; Epsilon at the top of the atmosphere
double TTop	= 216.65; Temperature at the top of the atmosphere
double AgNorm	= 0.6; Default surface albedo
double Sinle	sun angle above surface (M _{ye}) [rad]
double Sinl	sun angle above surface (M _y) [rad]

9.4 Control files

Below, the file formats for the necessary and optional input and output files to operate the GRAMM model are described. A graphical user interface (GUI) facilitates generating the input files and provides several features to analyse the output of GRAMM. For more details about the GUI the reader is referred to the manual, which is included in the GRAMM/GRAL package that can be downloaded from the website: <http://lampx.tugraz.at/~gral/>

9.4.1 Input files

9.4.1.1 GRAMM.geb

Used by: GRAMM.exe, GUI, GRAL.exe

It defines the number of grid cells in the horizontal and vertical directions as well as the minimum and maximum coordinates of the model domain in west-east and south-north directions. This information is further used in the GRAMM model to retrieve the horizontal grid sizes. The file has the following format (the exclamation marks are used as column separators when reading the file):

1	399	!number of cells in x-direction
2	399	!number of cells in y-direction
3	30	!number of cells in z-direction
4	-15000	!West border of GRAMM model domain [m]
5	14925	!East border of GRAMM model domain [m]
6	-15000	!South border of GRAMM model domain [m]
7	14925	!North border of GRAMM model domain [m]

9.4.1.2 GEOM.in

Used by: GUI

It defines the path for the input orography file, the cell-height of the first grid layer, and the vertical stretching factor used to compute cell-heights with increasing elevation.

```

geom.in x
1 S:\Benutzer\Oettl\Programme\GRAMM\Validierung\E3_Datensatz_3DRuecken\3D-Ruecken-Topographie.txt
2 20
3 1.16

```

9.4.1.3 GGEOM.asc

Used by: GRAMM.exe, GRAL.exe

It contains most of the information needed to establish the GRAMM grid. Although it is written as ASCII textfile, the format is not really designed to retrieve information using a text editor. The following parameters and arrays are included in subsequent order:

NX, NY, NZ, X, Y, Z, AH, VOL, AREAX, AREAY, AREAZX, AREAZY, AREAZ, ZSP, DDX, DDY, ZAX, ZAY, IKOOA, JKOOA, WINKEL, AHE

WINKEL is a historic parameter equal to zero (not used anymore).

An example code in C# is presented to generate the file ggeom.asc:

```

StreamWriter writer = new StreamWriter(Path.Combine(form1.projectname, @"Computation\ggeom.asc"),
false);
writer.Write(Convert.ToString(NX) + " " + Convert.ToString(NY) + " " + Convert.ToString(NZ) + " ");
for (int i = 1; i < NX + 2; i++)
{
    writer.Write(Convert.ToString(Math.Round(X[i], 2)).Replace(decsep, ".") + " ");
}
for (int i = 1; i < NY + 2; i++)
{
    writer.Write(Convert.ToString(Math.Round(Y[i], 2)).Replace(decsep, ".") + " ");
}
for (int i = 1; i < NZ + 2; i++)
{
    writer.Write(Convert.ToString(Math.Round(Z[i], 2)).Replace(decsep, ".") + " ");
}
writer.WriteLine();
for (int j = 1; j < NY + 1; j++)
{
    for (int i = 1; i < NX + 1; i++)
    {
        writer.Write(Convert.ToString(Math.Round(AH[i, j], 2)).Replace(decsep, ".") + " ");
    }
}
writer.WriteLine();
for (int k = 1; k < NZ + 1; k++)
{
    for (int j = 1; j < NY + 1; j++)
    {
        for (int i = 1; i < NX + 1; i++)
        {
            writer.Write(Convert.ToString(Math.Round(VOL[i, j, k], 2)).Replace(decsep, ".") + " ");
        }
    }
}
writer.WriteLine();
for (int k = 1; k < NZ + 1; k++)
{
    for (int j = 1; j < NY + 1; j++)
    {
        for (int i = 1; i < NX + 2; i++)
        {
            writer.Write(Convert.ToString(Math.Round(AREAX[i, j, k], 2)).Replace(decsep, ".") + " ");
        }
    }
}

```

Appendix A - Control files

```
    }
}
writer.WriteLine();
for (int k = 1; k < NZ + 1; k++)
{
    for (int j = 1; j < NY + 2; j++)
    {
        for (int i = 1; i < NX + 1; i++)
        {
            writer.Write(Convert.ToString(Math.Round(AREAY[i, j, k], 2)).Replace(decsep, ".") + " ");
        }
    }
}
writer.WriteLine();
for (int k = 1; k < NZ + 2; k++)
{
    for (int j = 1; j < NY + 1; j++)
    {
        for (int i = 1; i < NX + 1; i++)
        {
            writer.Write(Convert.ToString(Math.Round(AREAZX[i, j, k], 2)).Replace(decsep, ".") + " ");
        }
    }
}
writer.WriteLine();
for (int k = 1; k < NZ + 2; k++)
{
    for (int j = 1; j < NY + 1; j++)
    {
        for (int i = 1; i < NX + 1; i++)
        {
            writer.Write(Convert.ToString(Math.Round(AREAZY[i, j, k], 2)).Replace(decsep, ".") + " ");
        }
    }
}
writer.WriteLine();
for (int k = 1; k < NZ + 2; k++)
{
    for (int j = 1; j < NY + 1; j++)
    {
        for (int i = 1; i < NX + 1; i++)
        {
            writer.Write(Convert.ToString(Math.Round(AREAZ[i, j, k], 2)).Replace(decsep, ".") + " ");
        }
    }
}
writer.WriteLine();
for (int k = 1; k < NZ + 1; k++)
{
    for (int j = 1; j < NY + 1; j++)
    {
        for (int i = 1; i < NX + 1; i++)
        {
            writer.Write(Convert.ToString(Math.Round(ZSP[i, j, k], 2)).Replace(decsep, ".") + " ");
        }
    }
}
writer.WriteLine();
for (int i = 1; i < NX + 1; i++)
{
    writer.Write(Convert.ToString(Math.Round(DDX[i], 2)).Replace(decsep, ".") + " ");
}
writer.WriteLine();
for (int i = 1; i < NY + 1; i++)
{
    writer.Write(Convert.ToString(Math.Round(DDY[i], 2)).Replace(decsep, ".") + " ");
}
writer.WriteLine();
for (int i = 1; i < NX + 1; i++)
{
    writer.Write(Convert.ToString(Math.Round(ZAX[i], 2)).Replace(decsep, ".") + " ");
}
writer.WriteLine();
for (int i = 1; i < NY + 1; i++)
{
    writer.Write(Convert.ToString(Math.Round(ZAY[i], 2)).Replace(decsep, ".") + " ");
}
}
```

```

writer.WriteLine();
writer.Write(Convert.ToString(IK00A).Replace(decsep, ".") + " ");
writer.Write(Convert.ToString(JK00A).Replace(decsep, ".") + " ");
writer.Write(Convert.ToString(winkel).Replace(decsep, ".") + " ");
writer.WriteLine();
for (int k = 1; k < NZ + 2; k++)
{
    for (int j = 1; j < NY + 2; j++)
    {
        for (int i = 1; i < NX + 2; i++)
        {
            writer.Write(Convert.ToString(Math.Round(AHE[i, j, k], 2)).Replace(decsep, ".") + " ");
        }
    }
}

writer.Close();

*****

```

9.4.1.4 Landuse.asc

Used by: GRAMM.exe, GRAL.exe

This is an optional file containing soil parameters as well as surface properties in dependence on the land-use categorization. By default CORINE categorization is utilized (see Table 1), though, a user may define own categories as well. In the absence of the file landuse.asc, GRAMM uses homogenous properties if the model is driven by categorized meteorological data stored in the file meteopgt.all. These default parameters are as follows: albedo 0.1, soil moisture 0.15, emissivity 0.9, heat conductivity 2.0, thermal conductivity 1E-6. Roughness length is taken from the file GRAMMin.dat.

If GRAMM is driven by detailed meteorological observations, the user is allowed to specify homogenous soil- and surface properties.

Landuse.asc is an ASCII text-file not designed to retrieve information using a text editor. The following parameters and arrays are included in subsequent order:

RHOB, ALAMBDA, Z0, FW, EPSG, ALBEDO

An example code in C# is presented to generate the file landuse.asc:

```

*****

StreamWriter writer = new StreamWriter(Path.Combine(form1.projectname, @"Computation\landuse.asc"));
for (int j = 1; j < NY + 1; j++)
{
    for (int i = 1; i < NX + 1; i++)
    {
        writer.Write(Convert.ToString(Math.Round(RHOB[i, j], 0)).Replace(decsep, ".") + " ");
    }
}
writer.WriteLine();
for (int j = 1; j < NY + 1; j++)
{
    for (int i = 1; i < NX + 1; i++)
    {
        writer.Write(Convert.ToString(Math.Round(ALAMBDA[i, j], 3)).Replace(decsep, ".") + "
");
    }
}
}

```

Appendix A - Control files

```
writer.WriteLine();
for (int j = 1; j < NY + 1; j++)
{
    for (int i = 1; i < NX + 1; i++)
    {
        writer.Write(Convert.ToString(Math.Round(Z0[i, j], 4)).Replace(decsep, ".") + " ");
    }
}
writer.WriteLine();
for (int j = 1; j < NY + 1; j++)
{
    for (int i = 1; i < NX + 1; i++)
    {
        writer.Write(Convert.ToString(Math.Round(FW[i, j], 4)).Replace(decsep, ".") + " ");
    }
}
writer.WriteLine();
for (int j = 1; j < NY + 1; j++)
{
    for (int i = 1; i < NX + 1; i++)
    {
        writer.Write(Convert.ToString(Math.Round(EPSG[i, j], 4)).Replace(decsep, ".") + " ");
    }
}
writer.WriteLine();
for (int j = 1; j < NY + 1; j++)
{
    for (int i = 1; i < NX + 1; i++)
    {
        writer.Write(Convert.ToString(Math.Round(ALBEDO[i, j], 3)).Replace(decsep, ".") + "
");
    }
}
writer.WriteLine();
writer.Close();
```

9.4.1.5 GRAMMin.dat

Used by: GRAMM.exe, GUI

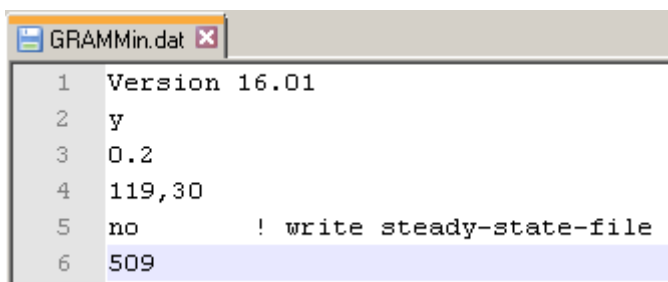
The first line provides information about the model version and is not used further.

The letter in the second line defines whether the GRAMM model is driven by categorized meteorological data as stored in the file meteopgt.all ("y") or if the model is initialized by specific meteorological observations ("n") (point- and profile measurements of wind and temperature). More information about the two ways to initialize the model is provided later in this document.

The number in the third line is the roughness length in [m], and the first number in the fourth line defines the weather situation to start with (only applicable and used if meteopgt.all is used for providing meteorological input data), while the second number in the fourth line provides information about the number of cells at the lateral boundaries over which orographical data is smoothed. This information is not used in GRAMM.

Line 5 defines whether files named "xxxxx_steady_state.txt" will be written by GRAMM at the end of each simulation. These files contain information about the fulfilment of the steady-state requirement of the VDI guideline 3783-7 (see chapt. 5 and for further descriptions of these files chapt. 9.4.2.5).

Line 6 triggers the storage of intermediate flow-field files to be used in the match-to-observation algorithm in the Graphical User Interface. If larger than zero, intermediate files will be written. Furthermore, global radiation is computed transient.



```

1 Version 16.01
2 y
3 0.2
4 119,30
5 no      ! write steady-state-file
6 509

```

9.4.1.6 Meteopgt.all

Used by: GRAMM.exe, GRAL.exe

It provides the simplest way of meteorological input data to run the GRAMM model. Numbers in the first line denote the anemometer height in [m], the way in which meteorological data is provided (0 = categorized; 1 = raw data), and the sector width used in the wind-direction categorization (only applicable, if the data represent categorized meteorological data).

Beginning from the third line, each line represents a specific weather situation. GRAMM uses the data in each line for initialization and subsequently computes a quasi steady-state flow field. GRAMM automatically computes flow fields one after each other.

The first number represents either the wind-direction sector if the second number in the first line is set to zero, or the observed wind-direction if the second number in the first line is set equal one.

In the first case the wind-direction is randomly computed in GRAMM out of the range for each wind-sector defined by

$$WD_{\min} = WD * SW - 0.5 * SW$$

$$WD_{\max} = WD * SW + 0.5 * SW$$

Taking the numbers from the third line of the example file below would give:

$$WD = 0.5 * 10 = 5 \text{ deg.}$$

$$SW = 10$$

Thus, the wind-direction can be any number between 0 and 10 deg. The motivation for this treatment of wind-direction is to avoid finger-like structures in computed concentration maps with the GRAL model.

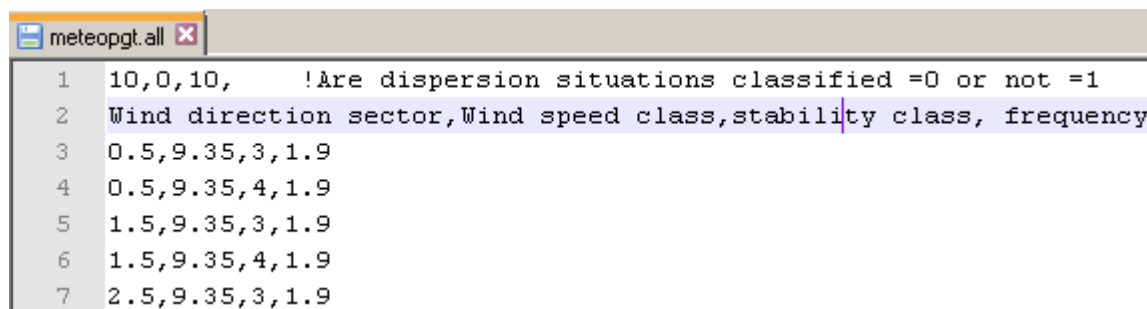
In the second case the wind-direction is obtained by

Appendix A - Control files

$WD = 0.5 * 10 = 5 \text{ deg.}$

As no categorization was carried out in this case, this wind-direction is used in GRAMM.

The second number in each weather situation is the wind-speed in [m/s], the third number is the stability class, and the fourth number is the frequency of the specific weather situation in [1/1000]. The recommended way to get stability classes is outlined in the GRAL-recommendation guideline, which can be downloaded from the GRAL website (<http://lampx.tugraz.at/~gral/>). Note that the frequency information in each line is not used in GRAMM or GRAL, but is only required in post-processing routines implemented in the GUI.



```
1 10,0,10,      !Are dispersion situations classified =0 or not =1
2 Wind direction sector, Wind speed class, stability class, frequency
3 0.5,9.35,3,1.9
4 0.5,9.35,4,1.9
5 1.5,9.35,3,1.9
6 1.5,9.35,4,1.9
7 2.5,9.35,3,1.9
```

9.4.1.7 Other meteorological data files

If the model is to be initialized with specific observations, the letter in the first line of the file GRAMMin.dat needs to be "n". The specific meteorological observations can be a combination of profile- and/or point measurements of wind- and temperature. Below is an example for the file format in case that one vertical profile of wind- and temperature measurements is available. The file name itself is arbitrary. The data for each meteostation is enclosed between "**". The sign "***" marks the end of the file. The first number in the brackets at the end of each line denotes the number of the meteorological station starting with 1 in ascending order, and the second number stands for the observational point within a vertical profile starting with 1 in ascending order.

```

Input_E3a.txt
1 DATENSATZ E3b
2 190989
3 21:00
4 NUMBER OF MONITORING STATIONS: 01
5 *
6 MONITORING STATION : Radiosonde (01)
7 DISTANCE FROM W : 10.00(01)
8 DISTANCE FROM S : 10.00(01)
9 HEIGHT ABOVE GROUND : 1.00(01)
10 HEIGHT ABOVE SEA LEV: 10.00(01,01)
11 TEMPERATURE [C]: 290.00(01,01)
12 U-COMPONENT OF WIND : 1.89(01,01)
13 V-COMPONENT OF WIND : 1.13(01,01)
14 HEIGHT ABOVE SEA LEV: 30.00(01,02)
15 TEMPERATURE [C]: 289.86(01,02)
16 U-COMPONENT OF WIND : 2.47(01,02)
17 V-COMPONENT OF WIND : 1.31(01,02)
18 HEIGHT ABOVE SEA LEV: 50.00(01,03)
19 TEMPERATURE [C]: 289.72(01,03)
20 U-COMPONENT OF WIND : 3.22(01,03)
21 V-COMPONENT OF WIND : 1.37(01,03)
22 HEIGHT ABOVE SEA LEV: 70.00(01,04)
23 TEMPERATURE [C]: 289.59(01,04)
24 U-COMPONENT OF WIND : 3.71(01,04)
25 V-COMPONENT OF WIND : 1.21(01,04)
26 HEIGHT ABOVE SEA LEV: 90.00(01,05)
27 TEMPERATURE [C]: 289.45(01,05)
28 U-COMPONENT OF WIND : 4.22(01,05)
29 V-COMPONENT OF WIND : 0.82(01,05)
30 HEIGHT ABOVE SEA LEV: 120.00(01,06)
31 TEMPERATURE [C]: 289.24(01,06)
32 U-COMPONENT OF WIND : 4.39(01,06)
33 V-COMPONENT OF WIND : 0.23(01,06)
34 HEIGHT ABOVE SEA LEV: 150.00(01,07)
35 TEMPERATURE [C]: 289.03(01,07)
36 U-COMPONENT OF WIND : 4.10(01,07)
37 V-COMPONENT OF WIND : -0.07(01,07)
38 HEIGHT ABOVE SEA LEV: 190.00(01,08)
39 TEMPERATURE [C]: 288.76(01,08)
40 U-COMPONENT OF WIND : 4.00(01,08)
41 V-COMPONENT OF WIND : 0.00(01,08)
42 HEIGHT ABOVE SEA LEV: 10000.00(01,09)
43 TEMPERATURE [C]: 221.00(01,09)
44 U-COMPONENT OF WIND : 4.00(01,09)
45 V-COMPONENT OF WIND : 0.00(01,09)
46 **

```

The specific file format as shown before is generated by the GRAMM model automatically at the end of an interactive questionnaire, which is invoked whenever “n” is written in the first line of the file GRAMMin.dat:

Prompts		Meaning
Should a stable stratification be provided starting from a certain height (Y)es (N)o:?		Press Y(es) or N(o). In case that no temperature observations are available above a certain height, it is recommended to provide the model with a specific temperature gradient above this height.
If Y(es):	starting at which height:	Provide the height above sea level from which the temperature gradient is to be used.

Appendix A - Control files

	Gradient [K/m]:	Provide the temperature gradient of the absolute temperature (not potential temperature). Neutral and stable gradients are recommended.
Read from existing file?		If there exists already a file that can be used press "Y". The file can be amended using the interactive communication of GRAMM. Otherwise "N".
Project Name		Provide any suitable name for the observations.
Input Date		Provide the date of the observations. No specific format required.
Input Time		Provide the time of the observations. No specific format required.
Input of point measurements = 1 Input of vertical profiles = 2 Select number		Press „1“ or „2“ to provide point- or profile measurements of wind and temperature.
In case of point measurements GRAMM will prompt for the following information: Note that at least one vertical profile is necessary for the interpolation procedure.	Monitoring station	Provide the name of the station
	Distance from western border	Provide the distance of the observational site from the western border of the model domain in [m].
	Distance from southern border	Provide the distance of the observational site from the southern border of the model domain in [m].
	Height above sea level	Provide the height of the observational site above sea level in [m].
	Height above ground level	Provide the height of the observational site above ground level in [m].
	Temperature	Provide the observed temperature in Celsius. If there is no temperature observation available use the value -273.15.
	Windvelocity	Provide the observed wind speed in [m/s]. If there is no wind speed observation available use the value 0.
	Winddirection	Provide the observed wind direction in deg. North = 0, South = 180 deg.
	Add to vertical profiles	Y(es) or N(o). In case of Y(es) the observed temperature is treated as being representative for the undisturbed atmosphere and the value is used throughout the domain, while in case of N(o) the value decreases with increasing distance to the observational site.
In case of profile measurements GRAMM will prompt for the following information:	Monitoring station	Provide the name of the station
	Distance from western border	Provide the distance of the observational site from the western border of the model domain in [m].
	Distance from southern border	Provide the distance of the observational site from the southern border of the model domain in [m].
	Height above ground level for closest point to the ground	Provide the height of the lowest measurement point of the profile above ground level in [m].
	Height above sea level	Provide the height of the observational site above sea level in [m].
	Temperature	Provide the observed temperature in Celsius.

		If there is no temperature observation available use the value –273.15.
	Windvelocity	Provide the observed wind speed in [m/s]. If there is no wind speed observation available use the value 0.
	Winddirection	Provide the observed wind direction in deg. North = 0, South = 180 deg.
	Input correct? End – Continue - Abort	Provide E(nd) – C(ontinue) – A(bort). E(nd) finalizes this profile. Other profiles or surface stations can still be added. C(ontinue) enables to add further measurements to this profile. A(bort) deletes the last input.
More input		Y(es) or N(o). N(o) finalizes the whole input procedure. Y(es) enables to add more monitoring data, i.e. the routine turns back to the stage “Input of point measurements = 1 Input of vertical profiles = 2 Select number”
Save input to file		Y(es) or N(o).
Filename		Provide a file name with extension to store all the data in ASCII file format.

9.4.1.8 IIN.dat

Used by: GRAMM.exe, GUI

IIN.dat is the main control file for the GRAMM execution. It is in ASCII file format and is generated by the GUI. However, in non-standard applications of the GRAMM model, it is necessary to amend the file using any sort of text-editor. The file format is shown in the next figure. Standard application of GRAMM means that it is driven by meteorological data stored in meteopgt.all.

Appendix A - Control files

1	COMPUTATION DATE (YYMMDD)	: 080210	!No influence when using meteopgt.all
2	BEGINNING OF COMPUTATION (hhmm)	: 0000	!No influence when using meteopgt.all
3	MAXIMUM ALLOWED TIME STEP DT [s]	: 10	!Range 1-100
4	MODELLING TIME (FOR VALUES >1(s) AND <1[%])	: 3600	!Range 0.01-1% or 2-infinite sec.
5	BUFFERING AFTER TIMESTEPS	: 3600	!No influence when using meteopgt.all
6	MAX. PERMISSIBLE W-DEVIATION ABOVE < 1 [mm/s]	: 0.01	!Not used
7	RELATIVE INITIAL HUMIDITY [%] GT.0	: 0.5	!No influence when using meteopgt.all
8	HEIGHT OF THE LOWEST COMPUTATION HEIGHT [m]	: 330.	!Not used
9	AIR TEMPERATURE AT GROUND [K]	: 280.0	!No influence when using meteopgt.all
10	GRADIENT OF THE POTENTIAL TEMPERATURE [K/100m]	: 0.01	!Not used
11	NEUTRAL LAYERING UP TO THE HEIGHT ABOVE GROUND:	: 10000	!Not used
12	SURFACE TEMPERATURE [K]	: 275.0	!No influence when using meteopgt.all
13	TEMPERATURE OF THE SOIL IN 1 M DEPTH [K]	: 283.0	!No influence when using meteopgt.all
14	LATITUDE	: 47	!Range -90 - +90 deg.
15	UPDATE OF RADIATION (TIMESTEPS)	: 10000	!No influence when using meteopgt.all
16	DEBUG LEVEL 0 none, 3 highest	: 0	!not used
17	COMPUTE U V W PN T GW FO YES=1	: 1111100	!GW and FO not used
18	COMPUTE BR PR QU PSI TE TB STR YES=1	: 1100011	!BR, PSI not used, TE not recommended
19	DIAGNOSTIC INITIAL CONDITIONS YES=1	: 1	!Not used
20	EXPLICIT/IMPLICIT TIME INTEGRATION I=1/E=0	: 1	!Not used
21	RELAXATION VELOCITY	: 0.1	!Range 0.01-1.0
22	RELAXATION SCALARS	: 0.1	!Range 0.01-1.0
23	Force catab flows -40/-25/-15 W/m ² AKLA 7/6/5	: 0	!1=ON suitable with steady state meteo.all
24	NESTING (=1) OR LARGE SCALE FORCING (=0)	: 0	!Not used
25	BOUNDARY CONDITION (1,2,3,4,5,6)	: 6	!1= homogenous Von Neumann, 6=Nudging (standard)
26	NON-STEADYSTATE/FORCING W. 'METEO.ALL' YES=1	: 0	!Should not be changed
27	NON-STEADYSTATE/FORCING W. IFU-MMS DATA YES=1	: 0	!Not used
28	GRAMM IN GRAMM NESTING YES=1	: 0	!Not used
29	Flowfield Output Format	: 0	!0=binary stream +header, 3=binary stream

The meaning of each line is outlined in the following table. Note that each line containing the information “!No influence when using [meteopgt.all](#)” indicates that the information provided in this line is not used if GRAMM is driven by the file [meteopgt.all](#), which is the standard GRAMM application. Furthermore, the information “!Not used” indicates, that the information in the corresponding line is not used at all (these are relicts resulting from historical model developments).

Line	Meaning
Computation date (YYMMDD)	Year, month, and day at the beginning of the simulation. This information is used for the sun's angle and azimuth.
Beginning of the computation (hhmm)	Hour and minute at the beginning of the simulation (UTC). This information is used for the sun's angle and azimuth.
Maximum allowed time step DT [s]	Each simulation starts with the minimum time step of 1.5 s. It increases in dependency on the convergence of the model up to the limit set here. If the upper limit set here is too high, the model might become numerically unstable. Guidance on reasonable upper limits are given in the GRAL recommendation guide.
Modelling time (for values >1(s) and <1[%])	Total modelling time in seconds. Note that if GRAMM is driven by meteopgt.all the modelling time is automatically adjusted by GRAMM in dependence on the stability class: in neutral (SC 4) conditions it is fixed with 600s, in convective (SC 1-3) conditions the modelling time is halved, and in stable (SC 5-7) conditions it is doubled. If the modelling time is set to values below 1, it is

	automatically interpreted as percentage of the time it takes the flow for the given wind speed to pass the given percentage of the model domain diagonally. If the value is above 1, the modelling time is taken as seconds.
Buffering after timesteps	Time in seconds after which an output wind-field file is written.
Relative initial humidity [%] GT.0	Relative humidity in the range 0.01 - 100, used for initialization in the total domain.
Air temperature at ground [K]	= Tatm; Used to define the initial soil- and surface temperatures. See next field.
Surface temperature [K]	= Tsini; Used to define the initial soil- and surface temperatures: Tsurf = Tair – (Tatm-Tsini) Thus, if the chosen “Air temperature at ground” is 280 K and the “Surface temperature” is 278 K, then, the surface temperature in each grid cell of the model is 1 K lower than the initial temperature of air in the lowest grid cell. Hence, if for example the air temperature decreases with height, so does the surface temperature.
Temperature of the soil in 1m depth	The temperature in this depth is assumed to be constant over the entire model simulation time. The soil temperatures between the deep soil temperature and the surface is obtained via linear interpolation. Automatically, GRAMM assumes a vertical gradient of -0.005 K/m for the deep soil temperature. Therefore, the value in this field is representative for the lowest elevation in the model domain.
Latitude	e.g.: 47. Influences the Coriolis force. Negative values are eligible.
Update of radiation (timesteps)	It is recommend to use values around 100. If the actual time step is 2 s, then, every 200 s the radiation model is invoked. Between these updates, radiation values are kept constant.
Compute U V W PN T GW FO YES=1	Set flags either 0 or 1, in order to invoke the computation of the following quantities: U: Wind component in west-east direction V: Wind component in south-north direction W: Vertical wind component PN: Non-hydrostatic pressure T: Potential temperature GW and FO are not used anymore.
Compute BR PR QU PSI TE TB STR YES=1	Set flags either 0 or 1, in order to invoke the computation of the following quantities: PR: Invokes the computation of Obukhov-length, Richardson number, friction velocity, etc. QU: Specific humidity TE: Turbulent kinetic energy TB: Invokes that the atmospheric temperature at the lowest grid level is influenced by the

Appendix A - Control files

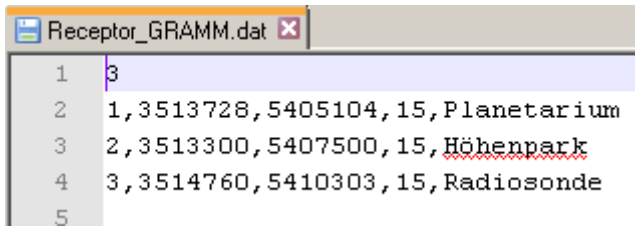
	surface temperature STR: Radiation model. BR and PSI are not used anymore.
Relaxation velocity	Value between 0 and 1. Applied for the three wind components. The lower the value the better the numerical stability, but the lower the rate of convergence. Recommended values can be found in the GRAL recommendation guide.
Relaxation scalars	Value between 0 and 1. Applied for temperature, humidity, turbulent kinetic energy, and dissipation. The lower the value the better the numerical stability, but the lower the rate of convergence. Recommended values can be found in the GRAL recommendation guide.
Force catab. flows -40/-25/-15 W/m ² AKLA 7/6/5	Sets fixed values for the sensible heat flux in dependence on stability class. Land-use has no influence on computed drainage flows anymore.
Boundary condition (1 or 6)	Sets the type of boundary condition at the lateral boundaries: 1: Zero gradient for all quantities. 6: Zero gradients for all quantities except for the horizontal wind-components, which are forced towards large-scale values using six lateral damping layers.
Non-steady state/forcing w. 'meteo.all' Yes=1	If set 1, GRAMM uses the meteorological data stored in meteo.pgt.all to define the geostrophic wind, which is linearly interpolated between the updates. In this way a simple non-steady-state forcing can be realized.
Flowfield Output format	0: standard binary stream + header (GUI) 3: same as 0 but without header

9.4.1.9 Receptor_GRAMM.dat

Used by: GRAMM.exe

Optionally, receptor points can be defined. Wind speed, direction, and temperature are written to the file GRAMM.dat after the update interval provided in the file IIN.dat in the field "buffering after time steps". This routine is only invoked if the model is not driven by the file meteo.pgt.all.

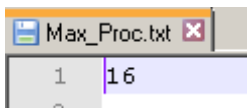
The first line is the number of receptors followed by lines, each containing the number of the receptor in ascending order, the x-, and y-coordinates in [m], the height above ground in [m], and the receptor's name.



9.4.1.10 Max_Proc.txt

Used by: GRAMM.exe, GRAL.exe

Sets the maximum number of threads to be used for parallel computing. The number can be larger than the actual number of processors available on the computer (in this case simply all available processors are used). The file contains only one line with the corresponding figure.



9.4.1.11 CustomInit.txt

The individual GRAMM initialization allows to adapt the initial parameters for GRAMM individually for user defined conditions. It should be noted that some initialization parameters must be adapted to the given meteorology (especially the stability classes). For this reason, it is suggested to run an individual quality control on the wind situations calculated in this way. However, very good results for single meteorological situations can be achieved by using the optional individual initialization.

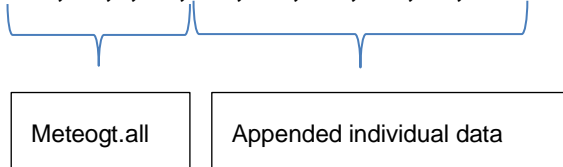
1. Copy the "meteopgt.all" file
2. Rename the file to "CustomInit.txt"
3. Append additional data to the desired situations, with the following content:
 - a. Altitude above snow conditions should be taken into account (>0)
 - i. Snow reduces the roughness length to 0.01 m if the original roughness is below 0.4 m
 - ii. Snow sets the ground albedo to 1
 - iii. Snow sets the albedo (for the radiation model) to 0.6
 - iv. Snow changes the density of the soil
 - v. Snow changes the soil moisture
 - vi. Snow changes the soil emissivity
 - b. Initial air temperature at sea-level in 2m height above ground (>0)
 - c. Initial soil temperature at sea-level (>0)
 - d. Initial soil core temperature at sea-level in 1 m depth (>0)

Appendix A - Control files

- e. Difference temperature to the ground for water bodies (0 ignored)
- f. Relative air humidity (> 0)
- g. Inversion height above lowest terrain cell within the GRAMM domain (> 0)
- h. Gradient of the air temperature – default value -0.0065 K (0 ignored)
- i. Gradient of the air temperature below the inversion height - default $+0.01$ K (0 ignored)
- j. Gradient of the soil temperature - default value -0.005 K (0 ignored)

Example

27.5,0.3,7,0.4,100,265,272,269,0.8,150



User defined initialization in this example:

Snow height from an altitude of 100 m above sea level

Air temperature at sea level 265 K

Ground temperature at sea level 272 K

Ground core temperature 269 K

Relative humidity 80%

Inversion height at 150 m altitude above the lowest cell in the GRAMM domain

9.4.2 Output files

9.4.2.1 *.wnd files

Used by: GUI, GRAL.exe

Computed wind fields are stored in binary files with the file extension „wnd“. If the files are to be used for postprocessing with the GUI, they need to have a header text (Option 0 in the last line of the input file IIN.dat). A code excerpt written in C# is given in the following to see how the files are generated. Note that in case of Option 0 (standard option), the wind components are written as integer values in [cm/s]. The filename itself contains the weather situation utilizing 5 digits (e.g. 00001.wnd corresponds to the flow field of the first weather

situation). The header contains information about the number of grid cells in each direction (NI , NJ , NK), and the horizontal grid size in [m].

```
string wndfilename = (Convert.ToString(Program.IWETTER).PadLeft(5, '0') + ".wnd");

BinaryWriter writer = new BinaryWriter(File.Open(wndfilename, FileMode.Create));
int header = -1;
Int16 dummy;
float GRAMMhorgridsize = (float)Program.DDX[1];

//there are two different formats: IOOUTPUT = 0 (standard output for GRAL-GUI users)
//and IOOUTPUT = 3 for SOUNDPLAN USERS
if (Program.IOOUT == 0)
{
    writer.Write(header);
    writer.Write(NI);
    writer.Write(NJ);
    writer.Write(NK);
    writer.Write(GRAMMhorgridsize);
    for (int i = 1; i <= NI; i++)
        for (int j = 1; j <= NJ; j++)
            for (int k = 1; k <= NK; k++)
            {
                dummy = Convert.ToInt16(Program.U[i][j][k] * 100);
                writer.Write(dummy);
                dummy = Convert.ToInt16(Program.V[i][j][k] * 100);
                writer.Write(dummy);
                dummy = Convert.ToInt16(Program.W[i][j][k] * 100);
                writer.Write(dummy);
            }
}
if (Program.IOOUT == 3)
{
    for (int i = 1; i <= NI; i++)
        for (int j = 1; j <= NJ; j++)
            for (int k = 1; k <= NK; k++)
            {
                writer.Write((float)Program.U[i][j][k]);
                writer.Write((float)Program.V[i][j][k]);
                writer.Write((float)Program.W[i][j][k]);
            }
}

writer.Close();
```

9.4.2.2 *scl, *obl, *ust files

Used by: GUI, GRAL.exe

Computed fields for the stability class, the Obukhov length, and the friction velocity are stored in a zipped archive (*scl-extension) containing three binary files with the file extension „scl“, „obl“, and „ust“. If the files are to be used for postprocessing with the GUI (e.g. match-to-observation functionality), they need to have a header text. A code excerpt written in C# is given in the following to see how the files are generated. The filenames themselves contain the weather situation utilizing 5 digits (e.g. 00001.scl corresponds to the first weather situation).

Appendix A - Control files

The header contains information about the number of grid cells in each direction (NI , NJ , NK), and the horizontal grid size in [m].

```
//output for friction velocity and Obukhov length
// write a Zip file
string stabclassfilename = (Convert.ToString(Program.IWETTER).PadLeft(5, '0') +
".scl");
try
{
using (FileStream zipToOpen = new FileStream(stabclassfilename, FileMode.Create))
{
using(ZipArchive archive = new ZipArchive(zipToOpen, ZipArchiveMode.Update))
{
string ustarfilename = (Convert.ToString(Program.IWETTER).PadLeft(5, '0') + ".ust");
ZipArchiveEntry write_entry1 = archive.CreateEntry(ustarfilename);
using (writer = new BinaryWriter(write_entry1.Open()))
{
writer.Write(header);
writer.Write(NI);
writer.Write(NJ);
writer.Write(NK);
writer.Write(GRAMMhorgridsize);
for (int i = 1; i <= NI; i++)
for (int j = 1; j <= NJ; j++)
{
dummy = Convert.ToInt16(Program.UST[i][j] * 1000);
writer.Write(dummy);
}
}

string obukhovfilename = (Convert.ToString(Program.IWETTER).PadLeft(5, '0') + ".obl");
ZipArchiveEntry write_entry2 = archive.CreateEntry(obukhovfilename);
using (writer = new BinaryWriter(write_entry2.Open()))
{
writer.Write(header);
writer.Write(NI);
writer.Write(NJ);
writer.Write(NK);
writer.Write(GRAMMhorgridsize);
for (int i = 1; i <= NI; i++)
for (int j = 1; j <= NJ; j++)
{
dummy = Convert.ToInt16(Program.OL[i][j] * 10);
writer.Write(dummy);
}
}

//computation and ouput of stability classes
string stabilityfile = (Convert.ToString(Program.IWETTER).PadLeft(5, '0') + ".scl");
ZipArchiveEntry write_entry3 = archive.CreateEntry(stabilityfile);
using (writer = new BinaryWriter(write_entry3.Open()))
{
writer.Write(header);
writer.Write(NI);
writer.Write(NJ);
writer.Write(NK);
writer.Write(GRAMMhorgridsize);
for (int i = 1; i <= NI; i++)
for (int j = 1; j <= NJ; j++)
{
writer.Write(Program.stabilityclass[i][j]);
}
}
}
```

```

} // archive
} // Zip File
} // catch
catch {}

```

9.4.2.3 Problemreport.txt

Used by: Infile for the user

The file contains information about possible numerical problems in the simulations expressed in terms of the total mass divergence. If the final mass divergence is larger than the initial one, a message is written to the file "problemreport.txt". In addition, the maximum wind speeds in all three coordinate directions are provided for facilitating the detection of possible numerical problems. The user is encouraged to carefully analyse flow fields not meeting the criterion for mass divergence.

It is recommended to delete the file whenever GRAMM is re-launched again from the beginning (weather situation 1) as GRAMM just puts the computed values at the end of the file.

```

3 Final mass divergence is larger than initial mass divergence. Possible numerical instabilities detected for flow field: 2
4 Final mass divergence is larger than initial mass divergence. Possible numerical instabilities detected for flow field: 3

```

9.4.2.4 GRAMM.dat

Used by: Infile for the user

In case that the file Receptor_GRAMM.dat is existent, and if GRAMM is not driven by meteopgt.all, then, GRAMM writes the file GRAMM.dat, which is in ASCII format as depicted below. Each line stands for one weather situation. GRAMM writes for each receptor the wind speed [m s^{-1}], wind direction [deg], and absolute temperature [K]. For instance, if there are two receptors defined in Receptor_GRAMM.dat, then, the file GRAMM.dat will contain 6 columns. It is recommended to delete the file whenever GRAMM is re-launched again from the beginning as GRAMM just puts the computed values at the end of the file.

GRAMM.dat										
1	1.2	210.	11.0	1.1	314.	12.4	1.1	284.	12.6	
2	0.11	331	14.51	0.27	346	14.11	0.20	348	14.25	
3	0.47	320	13.95	0.54	334	13.69	0.51	308	13.93	
4	0.28	238	12.85	0.76	313	13.06	0.88	289	13.43	
5	1.45	204	10.94	1.13	312	12.41	1.21	283	12.81	

Appendix A - Control files

9.4.2.5 “xxxxx_steady_state.txt”

Used by: Infile for the user, visualizable in the GUI

This file contains a value for every cell at the lowest grid level. The value describes if the steady state criterion according to VDI 3783-7 is reached (see chapt. 5).

Value	Steady-state criterion met		
	U	V	W
0	✗	✗	✗
1	✓	✗	✗
2	✗	✓	✗
3	✓	✓	✗
4	✗	✗	✓
5	✓	✗	✓
6	✗	✓	✓
7	✓	✓	✓