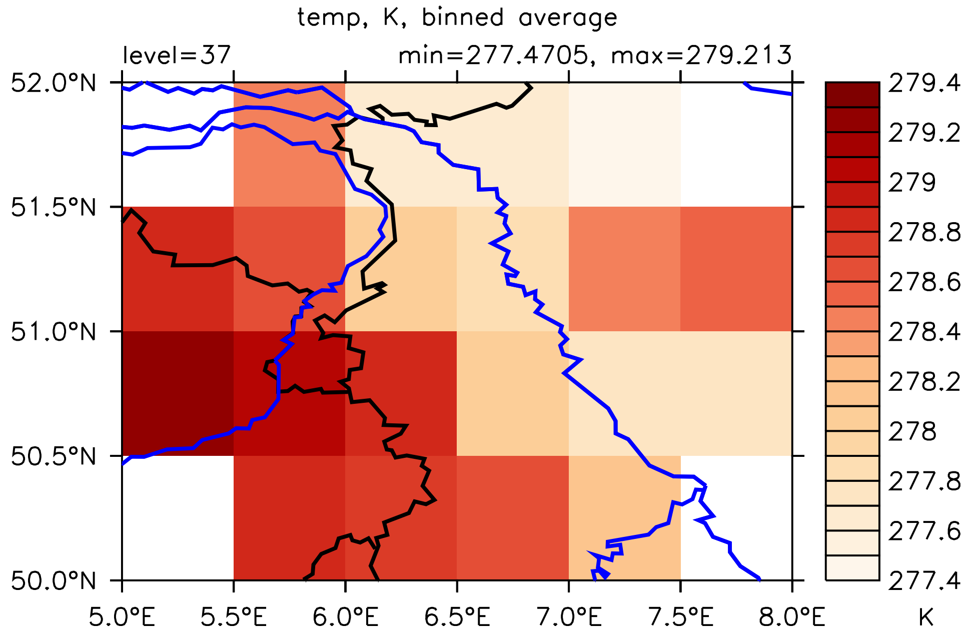
# GRAGG User Manual

*for the GRid AGGregation submodel*




**Bastian Kern and Patrick Jöckel**

Deutsches Zentrum für Luft- und Raumfahrt, Institut für Physik der Atmosphäre,
Oberpfaffenhofen, 82234 Wessling, Germany
`bastian.kern@dlr.de`

This manual is part of the electronic supplement of
B. Kern and P. Jöckel, "The Modular Earth Submodel System (MESSy, 2.50_extended) as diagnostic interface of the ICOsahedral Non-hydrostatic (ICON) modelling framework", Geosci. Model Dev. (2016), available at: `http://www.geoscientific-model-development.net`

Date: April 20, 2016

# Contents

# 1 Introduction

The GRid AGGregation (GRAGG) submodel is an advanced on-line diagnostic tool, which can be used to aggregate variables of the basemodel or any submodel on a user-defined (coarse) regular grid. The submodel's aggregation routines work on *channel objects* and provide their results as *channel objects* (see the CHANNEL user manual in this supplement and Jöckel et al., 2010). GRAGG supports the calculation of (area weighted) sums and averages, and discretised probability density functions (PDFs) and joint-PDFs (jPDFs) of two variables. The user interface is implemented via Fortran namelists and follows the MESSy standard. The user can define regular grids and assign them to aggregation operations on selected variables. Furthermore, for basemodels supporting multiple nested domains, aggregation operations can be restricted to specific domains.

# 2 GRAGG namelist user interface

The user interface of GRAGG is implemented via Fortran namelists provided in the `gragg.nml` file. Figure 1 shows an example of the GRAGG namelists. According to the MESSy standard (Jöckel et al., 2005, 2010), the namelist file contains a `CTRL` and a `CPL` namelist. At the moment, there are no namelist parameters in the `CTRL` namelist for GRAGG. The `CPL` namelist contains the actual setup for GRAGG. The user can define regular grids, mappings of different aggregation types to variables and grids, and control the domains of calculation for basemodels supporting multiple nested domains (*patches*).

```
&CTRL
/
&CPL
reg_lon_def(1)  = 5.25 , .5, 7.75
reg_lat_def(1)  = 50.25, .5, 51.75
reg_def_mode(1) = 1

map(1)  = 'qt_pdf', 'jpdf', 1, 10, 10, 'nh_state_prog:qv+qc+qi', 'nh_state_diag:temp'
map(2)  = 't_pdf' , 'pdf' , 1, 10,   , 'nh_state_diag:temp', ''
map(3)  = 'q_pdf' , 'pdf' , 1, 10,   , 'nh_state_prog:qv+qc+qi', ''
map(4)  = 't_ave' , 'ave' , 1,   ,   , 'nh_state_diag:temp', ''
map(5)  = 't_sum' , 'sum' , 1,   ,   , 'nh_state_diag:temp', ''

patches(1) = 3
patches(2) = 3
patches(3) = 3
patches(4) = 1,2,3
patches(5) = 1,2,3
/
```

Figure 1: Example namelist of the GRAGG submodel. For explanations, see text.

## 2.1 Regular grid definition

The specification of user-defined grids follows the notation used in ICON (Zängl et al., 2015). The definitions of the longitudinal (`reg_lon_def`) and latitudinal (`reg_lat_def`) range both consist of three entries:

- the value of the first grid box centre in degree
- the increment in degree or the number of grid boxes (see `reg_def_mode`)
- the value of the last grid box centre in degree

The `reg_def_mode` selects, whether the second value in `reg_lon_def` and `reg_lat_def` is treated as increment in degree or as number of grid points:

- 0: automatic selection, values > 5.0 are treated as number of grid points
- 1: increment in degree
- 2: number of grid points

As an example, the namelist entry

```
reg_lon_def(1)  = 5.25 , .5, 7.75
reg_lat_def(1)  = 50.25, .5, 51.75
reg_def_mode(1) = 1
```

defines a non rotated regular grid number 1 with a grid spacing of $0.5° \times 0.5°$ and grid centres ranging from 5.25° E to 7.75° E and from 50.25° N to 51.75° N.

In the current implementation a maximum of 10 different grids can be defined by the user (`MAX_GRIDS` in `messy_gragg_si.f90`).

## 2.2 Aggregation operation mapping

Via the mapping, the user defines aggregation operations and assigns the grids on which these operations are calculated. The `map` parameter consists of seven values, which not necessarily have to be defined, depending on the aggregation operation. The seven entries are:

1. the name of the output variable
2. the aggregation operation, which is one of `sum`, `wsum`, `ave`, `wave`, `pdf`, or `jpdf` (see Sect. 3)
3. the number of the assigned user-defined grid
4. for (j)PDF: the number of bins (in x-direction)
5. for jPDF: the number of bins in y-direction
6. the *channel object(s)* for calculation (jPDF: x-direction)
7. for jPDF: the *channel object(s)* for calculation (y-direction)

For entry 6 and 7, the submodel can apply an automatic summation of different *channel objects*, the notation follows the rule: `<channel>:<object>[+[<channel>:]<object>][+...]`. This allows for summation of different *channel objects* in the same or different *channels*.
Example:

```
map(1)  = 'qt_pdf', 'jpdf', 1, 10, 10, 'nh_state_prog:qv+qc+qi', 'nh_state_diag:temp'
map(4)  = 't_ave' , 'ave' , 1,   ,   , 'nh_state_diag:temp', ''
```

The first namelist entry defines a calculation of the jPDF, which is stored in the *channel object* '`qt_pdf`'. The calculation uses grid 1 and 10 bins in both directions. The variable for the x-direction is the sum of the *channel objects* '`qv`', '`qc`', and '`qi`' in the *channel* '`nh_state_prog`' and the variable in y-direction is the *channel object* '`temp`' in *channel* '`nh_state_diag`'.
The second namelist entry defines an averaging operation on grid 1 of the *channel object* '`temp`' in *channel* '`nh_state_diag`'. The results are stored in the *channel object* '`t_ave`'.
In the current implementation a maximum of 100 different mappings can be defined by the user (`MAX_MAPS` in `messy_gragg_si.f90`).

### 2.3 Multiple nested domains

For each defined mapping (see Sect. 2.2), the user can specify the domains, on which the calculation is performed. For each mapping `patches` contains a list of domains. The default, or the entry -1, specifies a calculation on all domains. Example:

```
patches(1) = 3
patches(4) = 1,2,3
```

The first namelist entry restricts the calculation of mapping 1 (the jPDF calculation from above) to the nested domain number 3.
The second entry sets the calculation of mapping 4 (the temperature averaging on grid 1) to be calculated for the nested domains 1, 2, and 3.

## 3 Aggregation types

At the moment, six aggregation operations (or aggregation types) are implemented in GRAGG. These aggregation operations are the (area weighted) summation (`(w)sum`), the (area weighted) averaging (`(w)ave`), and the (joint) probability density function (`(j)pdf`). The following subsections provide some implementation details on the aggregation operations.

### 3.1 Initialisation

During the initialisation phase, the GRAGG submodel processes user namelist and initialises the memory. For all user-defined regular grids, GRAGG has to find the triangles of the original ICON grid, which are part of a regular grid box. This is determined in a double loop over all user-defined grids and all triangles. The respective longitude $\lambda$ and latitude $\phi$ bin (or grid box indices in x-/y-direction) in the user-defined grid $G$ can be calculated as:

$$i_\lambda = \left\lfloor \frac{(\lambda_c - \lambda^G_{start} + \frac{1}{2}\Delta^G_\lambda)}{\Delta^G_\lambda} \right\rfloor + 1 \tag{1}$$

$$i_\phi = \left\lfloor \frac{(\phi_c - \phi^G_{start} + \frac{1}{2}\Delta^G_\phi)}{\Delta^G_\phi} \right\rfloor + 1, \tag{2}$$

with $\lambda_c$ and $\phi_c$ the longitude and latitude at the triangle centre, respectively. $\lambda^G_{start}$ and $\phi^G_{start}$ denote the user-defined starting longitude and latitude, respectively. $\Delta^G_\lambda$ and $\Delta^G_\phi$ are the longitudinal and latitudinal grid spacings, respectively. $\lfloor \ldots \rfloor$ denotes the mathematical floor function.
For each grid box, the local block and index, as well as the area of the triangle put into a list. Furthermore, the local number and local area of all triangles per grid box is calculated. These numbers are communicated over all processes and combined to the total number of triangles and total triangle area per grid box in the user-defined grid. For each triangle, the area weight is calculated from the triangle's area and the total area of all triangles in the respective grid box.

### 3.2 Aggregated summation

For the summation (`sum`) we sum over all values at the respective triangle centres per grid box in the time integration loop. In case of the area weighting (`wsum`), the area weight of the respective triangle is multiplied in the summation step. This gives partial results on each process, which are combined to the total sum in the output time-step via inter-task communication.

### 3.3 Aggregated average

For the averaging (`ave`) we sum over all values at the respective triangle centres per grid box in the time integration loop. In case of the area weighting (`wave`), the area weight of the respective triangle is multiplied in the summation step. In a second step, the values are divided by the total number of triangles per grid box. This gives partial results on each process, which are summed up over all processes in the output time-step via inter-task communication.

### 3.4 Discrete (joint) probability density function

Discretised PDF and jPDF calculation can be selected by the user via namelist using the aggregation types `pdf` and `jpdf`, respectively. During the time integration loop for all variables the minimum and maximum values per grid box over all processes are determined. This step involves inter-task communication and is done for the efficient storage of the discretised (j)PDFs. From the minimum and maximum values, the bin width $\Delta_\zeta$ of the (j)PDF in each grid box $m$ is calculated from the user-defined number of bins $n_\zeta$ for the respective variable ($\zeta = x$ for PDF, $\zeta = (x, y)$ for jPDF):

$$\Delta\zeta^m = (max(Z^m) - min(Z^m))/n_\zeta. \tag{3}$$

$Z^m = (\zeta_1^m, \ldots, \zeta_N^m)$ are the values of $\zeta$ at the centre of the $N = N(m)$ triangles in the grid box $m$.
In a loop $i = 1 \ldots N$ over all triangles in each grid box, GRAGG calculates for each value $\zeta$ the bin $b_{\zeta i}^m$ of the discrete (j)PDF ($\zeta = x$ for PDF, $\zeta = (x, y)$ for jPDF):

$$b_{\zeta i}^m = \lfloor (\zeta_i^m - min(Z^m))/\Delta\zeta^m \rfloor + 1. \tag{4}$$

Here $\lfloor \ldots \rfloor$ is the mathematical floor function.

For jPDFs, the total bin is determined as follows, as we have to store the jPDF as 1-D vector for each grid box in our output file:

$$b_{xyi}^m = (b_{yi}^m - 1) * n_x) + b_{xi}^m. \tag{5}$$

For each of the $N$ triangles in the grid box $m$, GRAGG increases the $b_{\zeta i}^m$-th or the $b_{xyi}^m$-th component of the bin vector (of length $\prod n_\zeta$) by 1 for the PDF and jPDF, respectively. In this way, the number of values fitting in each bin of the discrete (j)PDF is counted. After the loop over all triangles on one process, the bin vector is normalised by division of the total number of triangles in the respective grid box. GRAGG just has calculated local results per process. In the final step, these local results are combined (summed up over all processes) in the output time-step via inter-task communication.

## References

Jöckel, P., Sander, R., Kerkweg, A., Tost, H., and Lelieveld, J.: Technical Note: The Modular Earth Submodel System (MESSy) - a new approach towards Earth System Modeling, Atmos. Chem. Phys., 5, 433–444, doi:10.5194/acp-5-433-2005, 2005.

Jöckel, P., Kerkweg, A., Pozzer, A., Sander, R., Tost, H., Riede, H., Baumgaertner, A. J. G., Gromov, S., and Kern, B.: Development cycle 2 of the Modular Earth Submodel System (MESSy2), Geoscientific Model Development, 3, 717–752, doi:10.5194/gmd-3-717-2010, 2010.

Zängl, G., Reinert, D., Rípodas, P., and Baldauf, M.: The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core, Quarterly Journal of the Royal Meteorological Society, 141, 563–579, doi:10.1002/qj.2378, 2015.