ICON

Deutscher Wetterdienst
Wetter und Klima aus einer Hand
DWD

Wrap-Up and Best Practice

DWD ICON Course | July 2025 | Florian Prill, DWD

Deutscher Wetterdienst
Wetter und Klima aus einer Hand
DWD

DKRZ
DEUTSCHES
KLIMARECHENZENTRUM

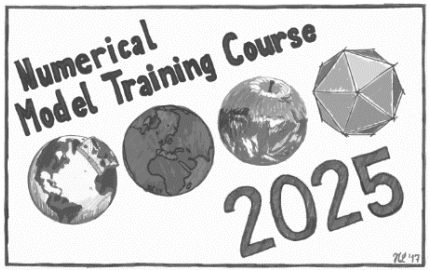
natESM

KIT
Karlsruher Institut für Technologie

Federal Ministry
of Education
and Research

Deutscher Wetterdienst
Wetter und Klima aus einer Hand
DWD

Before you start your journey

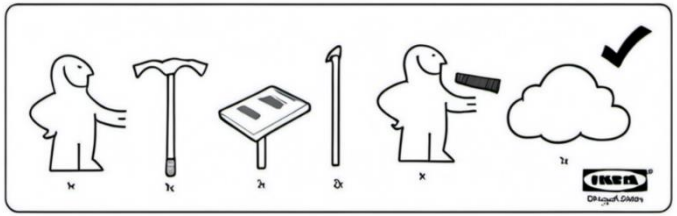


Recap of this week's lectures and exercises …

ICON

Before you start your journey

- Using an **HPC** makes sense, when it later comes to NWP use cases with realistic, high resolutions; running a “standard” x86 CPU cluster will simplify matters. Ideally, we recommend using the HPC platform of an ICON partner, e.g. Levante (DKRZ).
- Find out the essential **hardware parameters** (number of CPU cores, RAM). Know some details about your **compiler** and the **MPI installation**, eg. the version and compiler name.
- Necessary **libraries like Eccodes** are often available as modules, find out more!
- Explore your **file systems** (info often provided in the HPC documentation[1]). Choose a work partition for your experiments, ie. high-throughput, large-scale data, no backup.



[1] eg. https://docs.dkrz.de/doc/levante/file-systems.html

Install the ICON code

- Best practice: choose the **latest Open Source release version**[1] of ICON.
- You can make it easier to create the build setup by using a pre-configured **configure wrapper**[2]. Note that you can call these with additional arguments.
- Configuration: decide whether to activate non-default sub-packages, ie. set configure switches with the prefix `--enable-xxx` or omit others. You might, for example, consider enabling the Community Interface and its Python adapter right away[3].

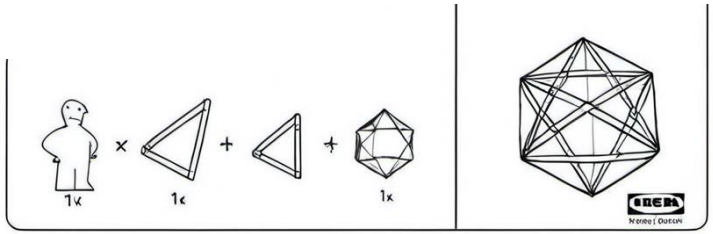
- Compiler options: make sure that you **generate debugging information** for proper stack traces (at least for the first time, `-g` option).
- When using the command `make -jX`, avoid placing too much load on the front-end node by selecting an appropriate value for `X`.

[1] https://gitlab.dkrz.de/icon/icon-model
[2] release-2025.04-public/config
[3] `--enable-comin --enable-bundled-python=comin`, see exercises

Make some estimates for your simulation

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

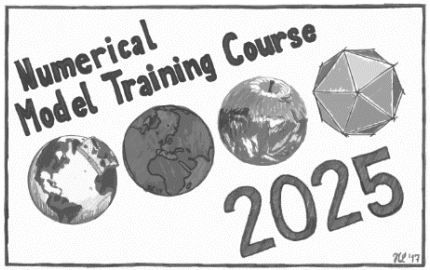
- Do you aim at a global **computational grid**, a nested grid hierarchy or a regional scenario?
- What is the desired **spatial resolution**? First, estimate the grid size. The “RnBk Finder Tool” on the Zonda Web service[1] is a helpful resource for finding the desired grid size/bisection value.
- Derive the **time step**, `dtime`, as described in Sec 3.7 of the ICON tutorial.
- What are the **queue limits for your HPC**? For example, what is the maximum number of nodes per job and the maximum wall clock time (Slurm: `sinfo`)? ICON scales well with the number of MPI processes. Therefore, we will need these numbers later to determine the optimal setup for your simulation and hardware.
- Take your time and make a few notes! 📝



[1] https://docs.icon-model.org/tools/tools.html#ref-tools-gridextpargui

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

Gather your data



ICON

Generate a suitable grid

- Use the **Zonda Web service**[1] to generate the grids and external parameter sets – all you need is a Github account. Grids on the same level are always (automatically) disjoint! Save the log files and Fortran namelist parameters for later, as these will document your setup.
- You might also make use of the list of pre-defined grids[2].
- Consider introducing a **reduced radiation grid** (domain 0), i.e., using a coarser horizontal grid for radiation than for dynamics. See Section 3.10 of the ICON tutorial for details and related Fortran namelist settings.
- In many cases, you can use a regional grid from a nested grid hierarchy for limited-area simulations, and vice versa. Note, however, that this is no longer compatible when using a reduced radiation grid.
- Advanced tip if you have access to the DWD ICON Tools: Post-process the grid with the “icon_delaunay” tool (barycentric interpolation, but ICON also supports on-the-fly calculation).

[1] https://docs.icon-model.org/tools/tools.html#ref-tools-gridextpargui
[2] http://icon-downloads.zmaw.de/dwd_grids.xml

Initial and boundary data from DWD

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

- The simplest option is the **Initialized Analysis** product (single file only, containing the analyzed state). This data can be retrieved using DWD's web-based PAMORE service[1]. The downside: Access to PAMORE comes with a small fee and some bureaucracy.
- A list of mandatory and optional **input fields** can be found in Section 11.4, “Analysis Products”, and Section 2.3, “Boundary Data Preparation”, of the ICON Tutorial. Remember that it makes no sense to remap tiled surface datasets!
- Appropriate temporal frequency of **boundary data** is important. For most of our real data NWP simulations, we choose a frequency of between 3 and 6 hours.
- Make a test with `grib_ls` and `cdo sinfov`, checking if you have the GRIB2 definitions set up properly. For example, when your surface pressure is labelled as `sp`... then you missed some environment setting (should be `PS`).

[1] https://www.dwd.de/DE/leistungen/pamore/pamore.html

Interpolate the data onto the computational grid

- As in our practical exercises, we recommend choosing **NetCDF** as the target file format for interpolation.
- We recommend the well-maintained **cdo tools** (Climate Data Operators). See, eg., the script `exercise_prepare_lam/scripts/remap_inidata_iconremap.ipynb` from the practical exercises.
- By the way: This would be a good point where you could make a first visualization (we suggest to use Python and `cartopy`). The simple visual check clarifies whether we have made any technical errors in the previous steps.

Interpolate the data onto the computational grid

Be careful: The greater the (spatial) **resolution jump** between the input data and the computational grid, the greater the potential spin-ups / spin-downs in different variables.

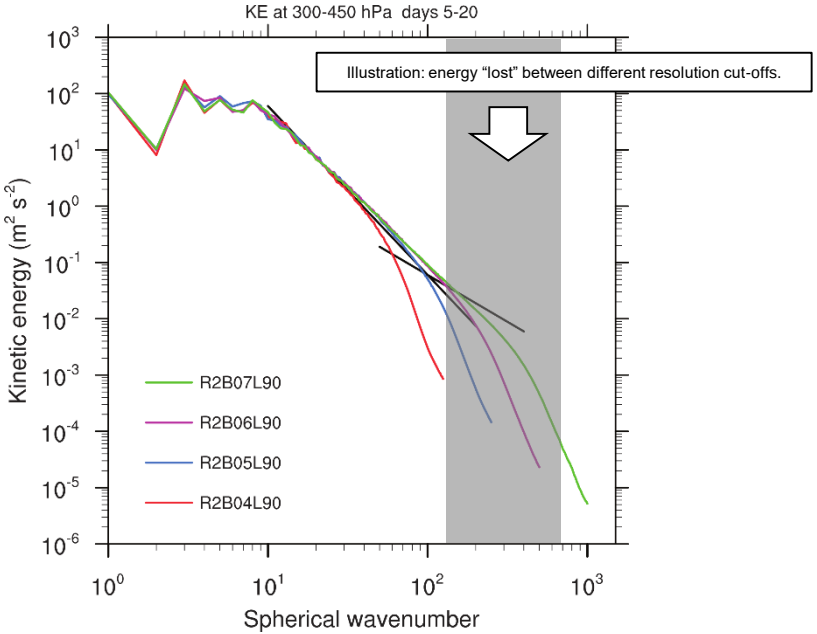
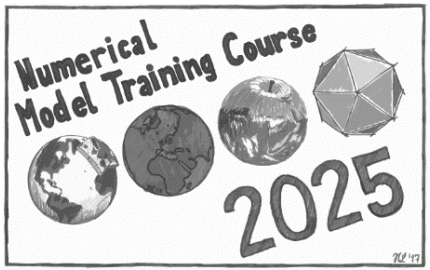


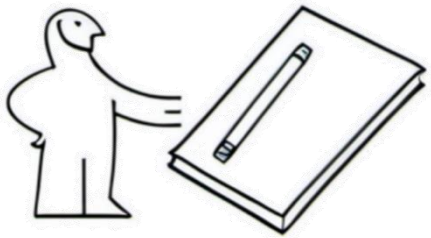
Fig. 7 from: Zängl, G., Reinert, D., Rípodas, P. and Baldauf, M. (2015), The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core. Q.J.R. Meteorol. Soc., 141: 563-579. https://doi.org/10.1002/qj.2378

Namelist setup and test run



Construct your Fortran namelist setup

- As a starting point: Begin with one of ICON's run script tools[1]. A **run-script** sets up the working directory, populates it with all required input files (grid files, namelists, etc.), sets environment variables, and runs the model.
- Here, we suggest to use the `make_runscript` tool, which is usually used to generate regression tests from the `run` sub-directory. Alternatively, you might start from one of the test setups from this ICON course.
- For **specific Fortran namelist settings** consider the table in `doc/Namelist_overview/Namelist_overview.pdf`, the documentation on https://docs.icon-model.org and the ICON tutorial!



[1] https://docs.icon-model.org/buildrun/buildrun_running.html#using-make-runscript-to-prepare-icon-experiments

Construct your Fortran namelist setup (cont'd)

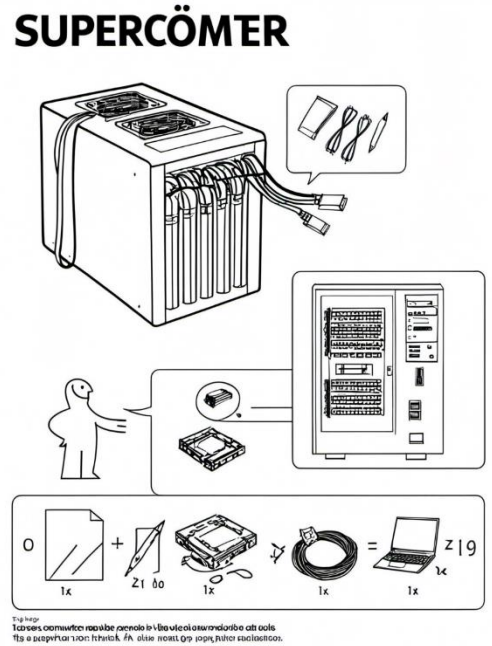
From the perspective of a computational scientist with a focus on dynamics and infrastructure:

Namelist Name	Purpose
`master_nml`	Controls restart information
`master_model_nml`	Contains model-specific information, including model type, and associated namelist files
`master_time_control_nml`	Manages calendar information, start/stop dates, and restarts
`run_nml`	Specifies time integration settings, model processes, and tracer configuration
`grid_nml`	Contains grid configuration details
`extpar_nml`	Provides external parameter settings
`sleve_nml`	Sets vertical coordinate information
`initicon_nml`	Determines the mode of initialization
`nonhydrostatic_nml`	Configures parameters for the nonhydrostatic dynamic core
`parallel_nml`	Handles parallel computing and vectorization settings
`transport_nml`	Manages transport scheme parameters

But you shouldn't forget about `physics_nml`, `tuning_nml`, `soil_nml`, `turbulence_nml` ...!

Construct your Fortran namelist setup (cont'd)

- To make things easier, it's best to keep all the input files and the ICON binary in one place, linked together (use the `ln -s` command) or as a copy. Make sure to have plenty of space available in your **experiment/output directory**.
- Initial data files are typically required for each nested domain. This can be overcome with only a slight loss of forecast skill by starting the nested domain(s) shortly after the global domain. See the Fortran namelist parameters `start_time` and `end_time`.
- Modify your Fortran namelist to have a **short run-time**.
- Having launched `cd ${RUNDIR} && sbatch icon.sbatch`, keep your fingers crossed and get yourself a ☕ coffee!



Check the results for technical plausibility

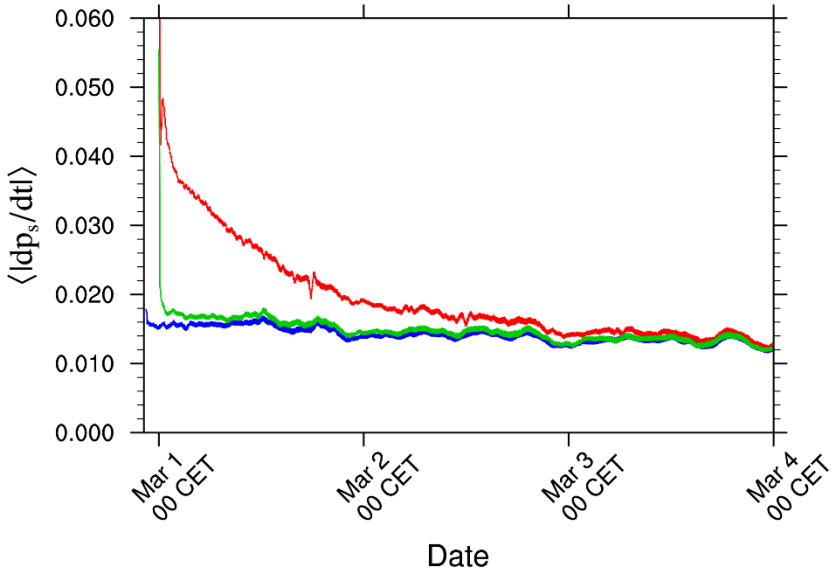
- Check the **log files**, as you have two: `stderr` and `stdout`. However, ICON intentionally writes its output to the standard error stream. Messages sent to `stderr` are less likely to be buffered and appear immediately in terminal windows or job logs.
- Take a look at the average cells/process: `Information on domain decomposition ... prognostic cells: max/min/avg`. The ratio of cells to halo cells is an important factor in assessing the efficiency of MPI communication.
- As before, make a quick visual check of your data (visualization script).
- Examine the table of ICON internal **sub-process timers** located at the end of the log file. Are there any dominant sub-processes, or is there an imbalance between them? Additionally, the MPI runtime often concludes with an “MPI Program Information” message containing helpful job information.

Check the results for technical plausibility

Monitor the area-averaged absolute **surface pressure** tendency $\left|\frac{dp_s}{dt}\right|$. This should decrease quickly during the start-up phase of your experiment and should not increase significantly at later lead times.

- Note: requires namelist setting `run_nml::msg_level ≥ 11`

Source: ICON Tutorial, Section “Necessary Input Data/Initial Conditions”, initialized analysis



Refine your experiment setup

- Enable/change the settings for the **physical parameterizations**. However, be careful: Not all Namelist options can be combined at random. For example
  - Convection is switched on in the parent grid but switched off in the nested child grid: This does not make much sense (see the lecture on Tuesday)
  - Only use the 2-mom microphysics scheme at a resolution below 3 km: This scheme is most suitable at convection-permitting or convection-resolving scales.
  - Do not change the microphysics between domains (the 2-mom microphysics scheme requires different tracers).
- Note: Changing the resolution and parameterisations will most probably affect the settings in the `tuning_nml`.

Refine your experiment setup (cont'd)

- Set the actual **start and end date** for the experiment in the `time_nml` namelist (`ini_datetime_string`, `end_datetime_string`).
- Using the information about the HPC, extrapolate the actual simulation times from the measured runtimes of your short test run.
- Adjust the **wallclock limit** and the **number of nodes** in your run-script, the number of tasks per node, and (implicitly) the total number of MPI tasks (Slurm: `#SBATCH --nodes=xxx`, `#SBATCH --ntasks-per-node=yyy`).
- You can control how many threads OpenMP creates by setting the environment variable `OMP_NUM_THREADS`.
- You can tune the memory consumption per node in a hybrid MPI and OpenMP job by adjusting the ratio between MPI tasks and OpenMP threads: using fewer MPI tasks per node and more OpenMP threads per task typically reduces total memory usage.

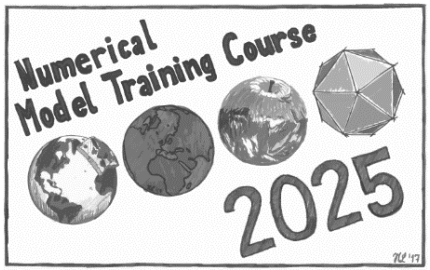
Refine your experiment setup (cont'd)

- Based on the extrapolation of your test run and the queue limits on your HPC, define the **restart checkpointing**. See Section 7.2 of the ICON tutorial for details on checkpointing and restart.
- Regarding the available **output products**: See the ICON tutorial appendix or the ICON Database Reference Manual[1]; check if it makes sense for you to enable whole variable groups with the `group:` keyword.
- Is the asynchronous output working for you? The file `output_schedule.ps` contains a rudimentary visualization of write and idle times for the different output processes.
- Consider other output mechanisms, such as output via the YAC coupler[2].
- If you need to understand advanced topics such as heterogeneous execution and GPU accelerators: Seek out an expert!

[1] https://www.dwd.de/SharedDocs/downloads/DE/modelldokumentationen/nwv/icon/icon_dbbeschr_aktuell.html
[2] see https://www.nat-esm.de/services/workshops-and-trainings/technical-trainings/20240717_natesm_a_pythonic_way_to_use_yac_dreier.pdf

Deutscher Wetterdienst
Wetter und Klima aus einer Hand
DWD

Follow your own path!



ICON

Follow your own path!

- The **ICON Community Interface**, ComIn), has been designed as a stable interface for third-party plugins[1]. Is ComIn sufficient for your needs?
- Otherwise: Get ready for the big adventure! Make sure that you are familiar with the `jc`, `jb` blocked layout of ICON arrays. Find out about how grids are stored, how parallel communication works in ICON. Etc.
- Write your own feature for the ICON model! 💪

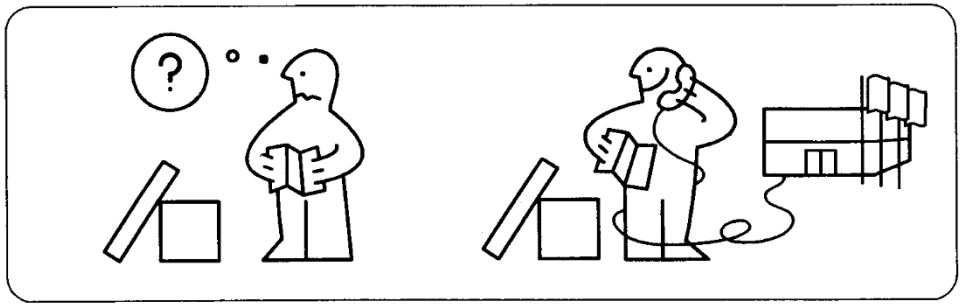
[1] https://gitlab.dkrz.de/icon-comin/comin

If something went wrong …

- If the build or configuration process fails, check the individual build sub-processes of the external libraries and build them one by one (always check the final paragraphs of the respective `config.log` file).
- Grids, constant data etc. are associated by `uuidOfHGrid` metadata. Check for consistency!
- If your program aborts, do not stop reading when you see a “SegFault” error message. This may only be a secondary error. Try to find the actual error message and stack trace.
- Log messages on maximum wind speed (`MAXABS VN, W`): how quickly do these values grow? Which domain and model level? This could, for example, indicate insufficient Rayleigh damping at TOA.
- Check the time-stepping output log for messages about the calling of physical modules (e.g. `downscaling of radiation output fields`). Does the calling frequency correspond to the intended namelist settings? In case of an error: Does the log message indicate where the error occurred?

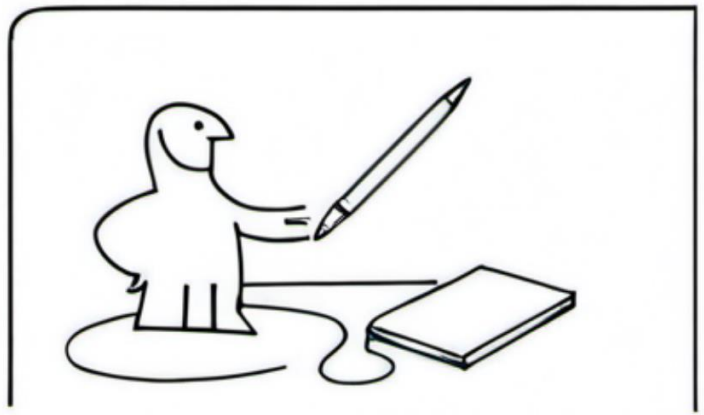
If something went wrong: Seek for help!

- Carefully re-read what you can find about your case on https://docs.icon-model.org/ and in the ICON tutorial.
- Who's the ICONist in your department? Get on his nerves!
- Double-check the time step `dtime` and decrease it if in doubt
- Build the ICON executable binary with enhanced debugging options, eg. enable array bounds checking for your compiler (GNU compiler: `-fcheck=bounds`).

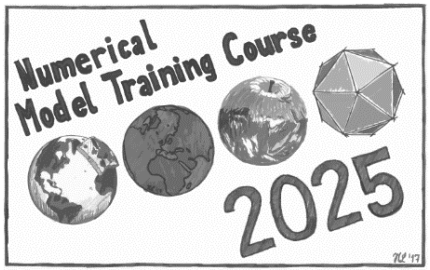


In the unlikely event of success

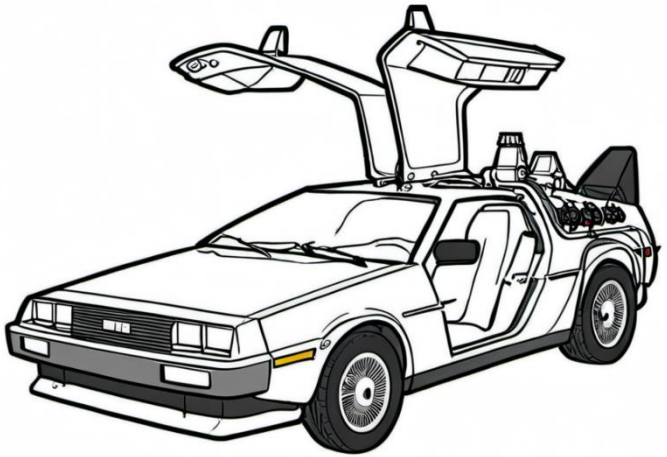
- Don't hide your findings, suggestions, or observed bugs!
- Be persistent! When your are interested in a collaboration, try to find the right ICON institute and developers
- Whom to contact? Try to find out about author information (`git blame`).



Final words



Final words



Over the past few years, various machine learning surrogate models have been announced.

- They claim to produce better forecast scores while using fewer hardware resources.
- Almost certainly these ML-based models will become part of the operational NWP process chain, eg. for real-time impact forecasts.
- Nevertheless, they have limited interpretability and do not yet offer predictions on a comparable number of vertical levels. They also lack forecast variability and sometimes physical consistency.
- ML models are still based on reanalysis data sets of “classical” grid point models.

These arguments should make it clear that the ICON model will remain relevant in the future.

Evaluation of the training

- With these concluding remarks, this training course ends.
- We hope that we have given you some ideas and, most importantly, some things to get you thinking about what you want to do next with ICON.
- Please take part in our survey - for an improved ICON training next year!
- Fill out the form: https://forms.gle/UJTi5KVgTXrAinCa6

Florian Prill

Met. Analyse und Modellierung
Deutscher Wetterdienst

e-mail: Florian.Prill@dwd.de