



Electronic Communications of the EASST
Volume 86 (1) Year 2026

deRSE26 - Selected Contributions of the 5th Conference for Research Software Engineering in Germany (Part 1)

Edited by Bernd Flemisch, René Caspart, Bernadette Fritzsch, Florian Goth, Jean-Noël Grad, Florian Thiery

Workflow automation of ICON model runs using auto-icon

Andreas Baer, Rumeng Li, Noel M. Chawang, René Caspart, Martina Klose

DOI: 10.14279/eceasst.v86.1.2803

License: This article is licensed under a CC-BY 4.0 License.

Electronic Communications of the EASST (https://eceasst.org).
Published by **Berlin Universities Publishing**
(https://www.berlin-universities-publishing.de/)

Workflow automation of ICON model runs using auto-icon

Andreas Baer [1,2], **Rumeng Li** [2], **Noel M. Chawang** [2], **René Caspart** [1], **Martina Klose** [2]

Scientific Computing Center[1] and Institute of Meteorology and Climate Research Troposphere Research[2], Karlrsuhe Institute of Technology

Abstract: Weather and climate modeling workflows comprise not only the numerical simulation itself, but also extensive pre-processing, data management, and post-processing tasks. Many of these tasks are repetitive across experiments and are therefore well suited for automation. Despite this potential, no widely adopted workflow management solution currently exists within the community of the ICOsahedral Nonhydrostatic (ICON) weather and climate model. To address this gap, auto-icon was developed as an extension to the workflow manager Autosubmit, enabling the orchestration and execution of ICON-based modeling workflows. In this work, we show several representative use cases that demonstrate the applicability of auto-icon to a range of modeling scenarios. The presented applications highlight improvements in workflow throughput, usability, reproducibility, and sustainability achieved through automated workflow management. More generally, this work illustrates the broader benefits that workflow management systems can provide for scientific modeling and computational research.

Keywords: Workflow Management, Sustainability, Climate Simulation, ICON model, Autosubmit

1 Introduction

Numerical weather prediction (NWP) and climate simulations predict the state of the atmosphere by solving a set of discretized dynamical and thermodynamical equations. These are complemented by equations for the explicit treatment or parametrization of additional physical processes like radiation, surface exchange, cloud microphysics, and aerosol transport. Earth system models further extend this framework by incorporating representations of land-surface processes, including vegetation dynamics and hydrological cycles, as well as ocean components that simulate the evolution of the ocean state analogously to atmospheric models. These modeling systems are computationally demanding and require complex workflows. Besides compilation, configuration, and execution of the actual model on an HPC system, the workflows include the retrieval of input data from distributed repositories, extensive pre-processing thereof, and post-processing steps to enable comparisons with observational datasets or reference simulations. As a consequence, each experiment involves a large number of repetitive and largely non-interactive tasks that require minimal user intervention and are therefore well suited for automation [JGP23].

Despite the potential of workflow automation for Earth system modeling, there is no commonly established tool for automating the entire workflow within the community of the ICOsahedral Nonhydrostatic (ICON) model [ZRRB14]. Existing tools are typically tailored to specific

applications, such as BaCy [PRS+14] for data assimilation, Spice [RG23] for ICON in Limited Area Mode (LAM) configurations, or ESMtools [GAB+21], which support model compilation and execution, but provide only limited functionality for comprehensive pre- and post-processing workflows. Consequently, many users still rely on so-called *runscripts*, i.e., self-contained shell scripts that execute basic model preparation and runtime procedures, but generally exclude broader workflow orchestration. This situation highlights a clear need for commonly accepted solutions that improve automation, reproducibility, and thereby sustainability throughout the entire modeling workflow [JGP23].

Although no broadly adopted workflow management solution currently exists for ICON, several mature workflow managers are widely used in the Earth system modeling community, with prominent examples including ecFlow [Bah11], Cylc [OSS18, OSM+19], and Autosubmit [MVP+16]. Among these, Autosubmit is particularly focused on improving workflow sustainability and experiment throughput, both of which are essential for research applications. A detailed comparison of these tools is provided in Section 2, motivating our choice of Autosubmit as the workflow management framework best suited to our objectives. Since Autosubmit itself is model agnostic, an additional integration layer is required to support ICON workflows. To address this need, auto-icon [BTJ+, BTJ+25] is developed as an extension that enables the execution and management of ICON experiments using Autosubmit as backend infrastructure. An overview on the design and implementation of auto-icon is provided in Section 3. The main part of this manuscript presents use cases of auto-icon for ICON simulations in Section 4, demonstrating both the flexibility of the framework and the practical benefits it provides to users. Finally, Section 5 summarizes the contributions of Autosubmit and auto-icon, and discusses how workflow management solutions of this kind can serve as a model for improving sustainability, reproducibility, and automation in other computational research domains, thereby offering valuable perspectives for the Research Software Engineering (RSE) community.

2 Workflow Management Tools in Earth System Modeling

Workflow managers are designed to execute computational workflows composed of multiple interdependent tasks that together form a directed task graph. These tasks are executed with the highest possible degree of parallelism while respecting intrinsic dependencies. In the context of High Performance Computing (HPC), workflow managers additionally handle the submission and monitoring of tasks with the batch system. By maintaining an internal representation of the workflow graph, they enable *meta-scheduling*, i.e. the optimization of task submission strategies with respect to the underlying batch scheduler and the structure of the workflow itself [SO18]. A wide range of workflow managers exists, including general-purpose solutions such as Airflow [Air], Jenkins [Jen], and Nextflow [DCF+17]. Within the communities of NWP and climate modeling, however, three workflow managers are particularly prominent: Autosubmit [MVP+16], ecFlow [Bah11], and Cylc [OSS18, OSM+19], that are briefly reviewed in the following.

Cylc [OSS18, OSM+19], originally developed at the National Institute of Water and Atmospheric Research (NIWA) in New Zealand and later further advanced in cooperation with the UK Met Office, is presumably the most feature-rich among the three systems. Although initially

designed for operational and research-oriented NWP forecasting, Cylc is not restricted to meteorological applications and provides a highly flexible workflow framework. It supports both cyclic and non-cyclic directed workflow graphs, as well as complex dependency structures involving support for multiple partially parallel and partially dependent workflows. For example, a task may depend on the successful completion of the corresponding task in a previous workflow cycle while the remainders of the two workflows are independent of each other. Combined with hierarchical workflow organization and decentralized workflow management, this design enables efficient scaling from small workflows to large, distributed workflow ensembles. Cylc further supports execution across multiple HPC systems within a single workflow, advanced event handling, configurable automatic retries, and a comprehensive graphical user interface (GUI) [OSS18].

The ecFlow workflow manager [Bah11], developed at the European Centre for Medium-Range Weather Forecasts (ECMWF), has the longest history among the three aforementioned tools. In contrast to Autosubmit and Cylc, which are implemented in Python 3, ecFlow is written in C++, providing excellent performance for the management of very large workflows. Similar to Cylc, ecFlow allows tasks to be grouped into families to simplify dependency management and triggering mechanisms. It also supports cyclic workflows, repetitions of complete workflows or workflow subsets, and cron-like scheduling functionality. ecFlow follows a server–client architecture in which a server executes and monitors workflows submitted by clients. The system provides three client interfaces: a command-line interface (CLI), a GUI, and a Python Application Programming Interface (API), each exposing most of the available functionality. As a result, ecFlow offers the most comprehensive GUI among the considered tools, enabling workflow submission, monitoring, and control directly through the GUI.

Autosubmit [MVP+16], developed at the Barcelona Supercomputing Center (BSC), differs from Cylc and ecFlow in several important aspects. While it provides common workflow management functionality such as automatic retries and support for multiple HPC platforms within the same workflow, it is specifically designed for non-cyclic workflows. Although cyclic behavior can partially be emulated through dependencies between separate workflows, dedicated support for cyclic execution is comparatively limited. Instead, Autosubmit is particularly well suited for ensemble simulations, sensitivity studies, and other multi-member experiment configurations. When combined with a time-based trigger, e.g. a cron job, this can also be used for quasi-operational forecasting as it is done at the BSC for their dust forecast [BGJ+20]. A distinguishing feature of Autosubmit is its integrated experiment management capability. In addition to orchestrating workflow execution, Autosubmit maintains a database of experiments together with their associated configuration files and metadata. This information can be exported in accordance with the RO-Crate standard, thereby improving provenance tracking and reproducibility, which are essential requirements for computational research [LCR+24]. Furthermore, Autosubmit provides advanced meta-scheduling capabilities by combining multiple workflow tasks into a single batch-system job allocation, thereby improving throughput and reducing scheduling overhead for large-scale workflows. For an exemplary use case, similar to the one described in Subsection 4.1, Marciani *et al.* [MCU+25] showed a reduction in total time to solution of 7 %. Finally, Autosubmit offers a comparatively lightweight and user-friendly interface, resulting in a shallow learning curve for new users.

All three presented tools have a large feature set and great flexibility regarding orchestrated

applications, which would allow implementation of the target ICON model workflows. Due to its combination of integrated experiment management, advanced task wrapping capabilities, and ease of use, we ultimately decided to adopt Autosubmit as the workflow management framework and developed auto-icon to enable application of Autosubmit with the ICON model.

3 auto-icon

As described in Section 2, Autosubmit was selected as the backend workflow manager due to its advanced support for workflow throughput optimization through task wrapping, as well as its comprehensive capabilities for experiment archiving and publication. Similar to other workflow managers discussed in Section 2, Autosubmit is model agnostic and therefore requires a dedicated interface for executing ICON workflows. auto-icon provides this integration layer by implementing the model-specific functionality required for preparing, executing, pre- and post-processing ICON simulations.

A detailed description of auto-icon is provided in [BTJ+, BTJ+25]. It consists of a collection of modular jobs, each representing a dedicated task within the workflow, such as building the model, executing simulations, or interpolating input datasets to the target model grid (cf. Figure 1). Within Autosubmit, each job is represented as an individual node in the workflow graph and configured through lightweight YAML configuration files provided by auto-icon. The executable logic of each task is implemented through Bash or Python scripts that interface directly with Autosubmit to retrieve and process the corresponding configuration. In addition, auto-icon includes a collection of reusable Python modules that encapsulate functionality shared across multiple tasks.

The overall workflow configuration is defined through a set of YAML files that are merged by Autosubmit into a unified experiment configuration. These configuration files specify the structure of the workflow, the tasks to be executed, the location and handling of input datasets, general experiment parameters such as simulation period and model resolution, and details of the target HPC system. Figure 2 shows a small snippet of the configuration, highlighting grid and timing information. The ICON model itself is configured via a Fortran namelist, which is a Fortran feature to read variable values directly from a text file in a key-value structure. The namelist file can either be provided directly in its native format or provided rewritten in YAML format. In both cases, auto-icon automatically patches selected configuration parameters in order to seamlessly integrate workflow-specific functionality such as the names of input and grid files created during workflow execution.

3.1 Development Process and User Onboarding

The development of auto-icon initially focused on core functionality, most notably building and executing the ICON model, accompanied by a wiki-based documentation. Once a minimal but usable feature set was established, pilot users were introduced to the framework through a dedicated workshop in order to maintain close interaction between users and developers. Subsequent feature development was guided by feedback from these early adopters and other community members, thereby prioritizing functionality with broad applicability across the user community.

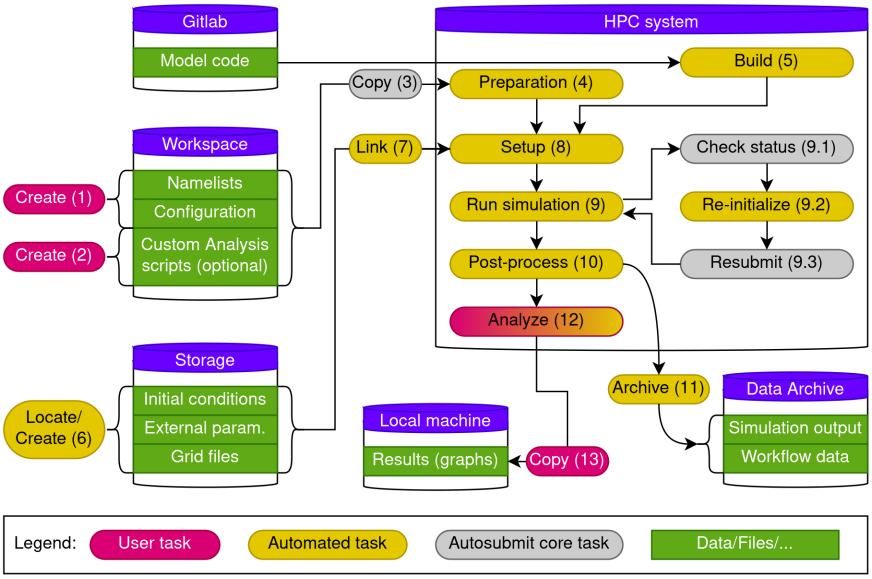


Figure 1: Overview of a workflow for an ICON simulation with auto-icon, comparable to the one describen in Subsection 4.1. Depicted are the steps executed by auto-icon, those are part of the core functionality of Autosubmit and those requiring user interaction. The Analysis task can (partially) be automated, but is in practice often done interactively by the user.

```yaml
# -- ----- Grid info ----- -- #
GRID: # Information on all domain-specific files
  FILELIST:          # List of input file types
    - GRID
    - IFS
    - HD_MASK
    - BC_SIC
...
  RADGRID: True      # Use a corase radiation grid
  DOM1:              # Domain #1
    TYPE: G          # Global domain
    R: 2             # Resolution specification (R)
    B: 4             # Resolution specification (B)
    GRID_NUMBER: 41  # Official grid ID
...
# -- ----- Basic experiment info ----- -- #
EXPERIMENT:
  # -- ----- Timing info ----- -- #
  DATELIST: 20111201      # List of startdates
  MEMBERS: "default var1 var2 var3 ..."
  CHUNKSIZEUNIT: month    # Chunking info:
  CHUNKSIZE: 1            # Run 13 chunks with
  NUMCHUNKS: 13           # 1 month length each
```

```yaml
NAMELIST:  # Common namelist for all members
  run_nml:                  # Main run configuration
    num_lev: 47             # number of full levels
    modeltimestep: 'PT4M'
    ltestcase: False        # run testcase
    ldynamics: True         # dynamics
    ltransport: True        # transport
...
  ham_lite_dustctl_nml:     # namelist section
    drag_part: 4            # default value
    cthr: 0.6               # default value
    smccor: 2               # default value
    vegopt: 1               # default value
...
MEMBERS:  # Modify namelist per member
  default:                  # member name
    ham_main_list_nml:      # namelist section
      nhamlite_dustemi: 3   # overwrite value
  var1:                     # member name
    ham_main_list_nml:      # namelist section
      nhamlite_dustemi: 2   # overwrite value
    ham_lite_dustctl_nml:   # namelist section
      drag_part: 2          # overwrite value
      vegopt: 0             # overwrite value
```

Figure 2: Snippets of the configuration, displaying the settings for grid and timing of the model run (left) and from the ICON namelist in YAML format (right). The configuration corresponds to the ICON-HAM-lite case described in Subsection 4.2.

This iterative development process led to the implementation of additional features, including automated retrieval of input data from commonly used repositories, pre-processing routines for preparing datasets for ICON simulations, and a generic interface to the Climate Data Operators (CDO) toolkit [KMSe] for post-processing tasks. Throughout the development process, two primary design goals were emphasized: maintaining simplicity and usability for end users while preserving sufficient flexibility to support a wide range of ICON model configurations without imposing restrictive workflow assumptions. To foster adoption by the local and wider communities, auto-icon was regularly presented within the developers' institution, at meetings of various research projects and groups, and at the ICON community meeting ICCARUS [Deu]. An early open-source release ensured broad accessibility and enabled external contributions and feedback. At a later stage, a hands-on training workshop was organized to foster community engagement and lowering the entry barrier for adoption.

As the code base expanded, the importance of software engineering practices such as ensuring code quality standards, continuous integration, and automated testing became increasingly apparent. These aspects were addressed through systematic refactoring efforts and the introduction of unit tests, complementing the automated end-to-end tests. An important observation throughout the development process was that users with little or no prior ICON experience were often the easiest to onboard, provided that comprehensive introductory documentation and tutorials were available. Users without experience with ICON modeling start with as little as three Autosubmit commands and one for auto-icon to run the simplest example. Using the possibilities and introductory guides offered by auto-icon, those users can then gradually learn how the ICON model and auto-icon work, how to build the model, obtain input data and adapt auto-icon to their needs. Users already familiar with ICON can also benefit from all the options auto-icon offers, although some initial effort may be required to integrate or migrate their existing scripts, tools, and workflows. Furthermore, early and continuous involvement of users proved essential for identifying the most relevant feature requirements and effectively guiding development priorities.

4 Use Cases

auto-icon has been successfully applied to a wide range of ICON experiments, demonstrating its flexibility in orchestrating various different simulation setups. Among the first applications are the default test cases of the Aerosol and Reactive Trace gases (ART) submodule [HBB+26], which cover fundamental features of ART using global, regional, and nested grids with distinct input datasets. While these cases can be executed directly from a pre-packaged input archive and thus benefit little from workflow automation, they serve as important templates for new ICON users, covering also many important features of the host model. auto-icon manages the complete workflow for these cases, including building the model, retrieving and extracting input data, and executing the simulations. More advanced applications target research and development objectives. Two use cases are highlighted in more detail: the evaluation of a newly implemented feature against a published model reference [KJA+21] and a sensitivity study for dust emission schemes in the ICON-HAM-lite model [WHS25, LGAK]. Further cases are shortly outlined to demonstrate the flexibility and variety of use cases.

4.1 Model validation with extensive input data generation and post-processing

A use case highlighting extensive pre- and post-processing is the validation of a newly implemented radiation multiple call scheme in ICON-ART, which allows the radiation routine to be invoked multiple times per radiation timestep, each time including a different set of aerosols to quantify their radiative effects [HBB+26]. The experiment compares ICON results against reference simulations with the MONARCH model [KJA+21], evaluates runtime implications, and tests three distinct configurations of the multiple call scheme. The configuration of the corresponding run presented in [HBB+26] is publicly available [BK].

The experiment comprised four members, all of which were global simulations: one reference run and three variants corresponding to the different multiple call configurations. The experiment spans the full year of 2012, including a one-month spin-up period. Daily global nudging to the ERA5 reanalysis [Cop23] requires generation of nearly 400 input files. As ERA5 is stored in one file per variable, several thousand raw files are remapped and merged to generate the necessary initial and boundary conditions. Simulations were partitioned into monthly chunks to comply with HPC wallclock limits. Autosubmit's native job-wrapping capabilities were leveraged: 13 remapping jobs and 13 simulation jobs were created, and pairs of monthly jobs were combined into single Slurm allocations.

Post-processing and visualization were automated using CDO [KMSe] to extract variables of interest, followed by their visualization to create publication-ready figures of annual averages of dust load and radiative effects. During model development, these were compared against the MONARCH reference. The full workflow involved over one hundred jobs, remapped thousands of input files, ran four simulations across 13 chunks each, and produced numerous figures. As the workflow itself is rather linear and conceptually simple, the greatest benefit of using auto-icon is the easy configuration of all four members with a simple YAML file and execution of the workflow by running a few Autosubmit commands. The wrapper functionality further enhanced throughput and storing the entire configuration in the auto-icon Git repository supports sharing and re-use.

4.2 Sensitivity study with automated configuration adjustments in a coupled model

A second use case addresses a sensitivity study of newly implemented physics-based mineral dust emission schemes within the ICON-HAM-lite model, a computationally efficient configuration of ICON together with the simplified aerosol module HAM-lite [WHS25], designed for global aerosol-climate simulations at kilometer-scale resolutions. This use case further entails coupling of the atmospheric component of ICON with the land-surface model JSBACH [RGG+21], a common setup for climate simulations. Two emission parameterizations, based on [Sha04] and [KMF+14], were incorporated to enable a physically grounded representation of dust emission [KJA+21, LGAK]. Several tuning parameters required optimization for each scheme, necessitating multiple simulation members with adjusted ICON namelist parameters for each.

In total, 73 experiments were executed, each representing a unique combination of emission scheme, vegetation representation, soil moisture correction, drag partition parameterization, and tuning factors. Simulations span from 20 January 2020 to 1 February 2021, with an 11-day spin-up, initialized from ERA5 reanalysis data [Cop23], ocean boundary conditions from AMIP II

[TWZ00], anthropogenic emissions from the AeroCom-II ACCMIP inventory following RCP4.5 ([HSG22]), and biogenic emissions from [GHE+95]. Potential dust source regions were defined using the climatological (2003–2015) frequency of occurrence of MODIS Deep Blue dust optical depth exceeding 0.2 [HTKH04, GPG+12, KJA+21].

The auto-icon workflow incorporated the preparation, configuration, and execution of all 73 simulations split into 40-day chunks to comply with HPC wallclock limits. The namelist for each simulation was automatically adjusted to the configured sensitivity parameters, minimizing the risk for inconsistencies or user-errors and maximizing reproducibility. To do so, auto-icon was used to create the namelist for each run from a template with all common settings and a separate section for each member with the member-specific entries. An example snippet how this is resembled in the configuration is depicted for two of the members in Figure 2. Outputs were consistently formatted and organized into per-member subdirectories, enabling efficient post-processing. This use case demonstrates the ability of auto-icon to execute large ensembles with minimal manual intervention, enhancing user friendliness, sustainability and traceability.

4.3 Various other applications

Further use cases are shortly outlined in the following. The first case uses the ICON-SmART configuration [Han], where the ART submodule for aerosol modeling is integrated into the seamless configuration of ICON [FPM+22, MLP+25]. Hereby, the atmosphere component of ICON was coupled both to the JSBACH land-surface model [RGG+21] and to the ICON ocean model [Kor18, KBJ+22], creating a full Earth system modeling configuration. A major challenge hereby was the handling of multiple components, each requiring their own namelist, input data and a distinct section of the global Message Passing Interface (MPI) communicator, complicating MPI task assignment. Furthermore, incorporating the ESMValTool [ABM+26] directly into the workflow, enabled automatic post-processing and evaluation of simulation results. The evaluation step ran immediately after model completion, guaranteeing reproducible quality checks for every experiment in this configuration.

Another use case features ICON in LAM configuration, where the grid and external parameter data was created with the community tool Zonda [JSJ] and the initial and boundary conditions were remapped by auto-icon (as described in Subsection 4.1). In this case, multiple experiments were conducted on different HPC systems, at different horizontal model resolutions, thus with different computational grids, and with multiple ICON versions. auto-icon proved valuable hereby as the entire workflow configuration, captured all common settings and was simply stored in a dedicated Git branch. It was easily distributed to all experiments and only a few fields needed adjustment when switching between model version, grid resolution, or target system.

Furthermore, the Large Eddy Simulation (LES) mode of ICON uses entirely different input datasets – provided by the user – and a distinct physics configuration via the ICON namelist. Despite these differences, *auto-icon* automates the build, configuration, and execution of the model, along with fetching of the provided input data, delivering a smooth, reproducible and easily configurable LES workflow.

Finally, current exploratory work is ongoing to support data assimilation in auto-icon by integrating the BaCy data assimilation tool [PRS+14] into an auto-icon workflow. This requires significantly more complex, cyclic workflows that have been successfully implemented with Au-

tosubmit for data assimilation with the MONARCH model [DEB+22].

5 Conclusions

The ICON community currently lacks a commonly established tool for workflow automation to orchestrate entire modeling workflows. In this paper, we demonstrate the capabilities of auto-icon to address this need by building on Autosubmit, a mature, powerful, and user-friendly workflow management tool for climate and Earth system modeling applications. As auto-icon is described in detail elsewhere [BTJ+, BTJ+25], we focus on exemplary use cases to show how auto-icon improves both workflow throughput and usability for a broad range of simulation setups.

Two major advantages of auto-icon are particularly evident. First, it automates the extensive pre-processing and post-processing tasks associated with climate and weather simulations, including data retrieval, interpolation, visualization, and evaluation workflows. Second, it enables the efficient management of large ensembles of closely related simulations, such as sensitivity studies. Although similar functionality can in principle be achieved with custom shell scripts, such scripts often require experiment-specific modifications to individual namelist entries and submission commands. auto-icon provides a standardized and reusable framework for defining ensemble members, modifying configurations, submitting jobs, and managing outputs, thereby reducing the need for each user to develop and maintain separate orchestration scripts for each experiment. Furthermore, integrating all workflow components into a unified and reproducible configuration promotes sharing, reuse, and long-term sustainability of computational experiments.

Owing to the comparatively simple and flexible interface provided by Autosubmit, establishing the connection between Autosubmit and ICON through auto-icon proved straightforward, enabling rapid development of the first functional versions of auto-icon. While the implementation of additional functionality could be achieved efficiently, building and maintaining an active user community emerged as an equally important but considerably more challenging task. In this context, open-source availability was a necessary prerequisite, but by itself insufficient to ensure adoption. Comprehensive documentation, introductory tutorials, and hands-on workshops proved essential for successful onboarding.

Finally, auto-icon may serve as a blueprint for similar developments in other scientific modeling communities. Although the tool itself is specifically tailored to the ICON model, many of its key concepts rely on the model-independent capabilities provided by Autosubmit. Consequently, the project not only contributes to broader adoption of Autosubmit within the Earth system modeling community, but also provides an openly available example of how generic workflow managers can be extended towards domain-specific scientific applications. More generally, the present work highlights the broader opportunities offered by workflow management systems for computational science. Their benefits extend beyond reducing manual effort and improving computational efficiency to enhancing the sustainability, traceability, and reproducibility of scientific results, which are central objectives of Research Software Engineering.

Acknowledgements: This work has received funding through the Helmholtz Association's Initiative and Networking Fund (grant agreement no. VH-NG-1533). This work was partly supported by the European Union's Horizon 2020 research and innovation program under Grant Agreement 101003470 - project Next Generation Earth Modelling Systems (NextGEMS). Part of the work has been funded in the scope of the TESSA project, awarded by the National High Performance Computing Center at KIT (NHR@KIT). This center is jointly supported by the Federal Ministry of Research, Technology and Space and the Ministry of Science, Research and Arts of Baden-Württemberg, as part of the National High-Performance Computing (NHR) joint funding program (https://www.nhr-verein.de/en/our-partners). Parts of this work was performed on the HoreKa supercomputer funded by the Ministry of Science, Research and Arts Baden-Württemberg and by the Federal Ministry of Research, Technology and Space. This work used resources of the Deutsches Klimarechenzentrum (DKRZ) granted by its Scientific Steering Committee (WLA) under project IDs 1215 and 1368. Datasets provided by DKRZ data management department via the DKRZ data pool were used.

Data availability: The software auto-icon is publicly available on Zenodo [BTJ+25] and in the public repository at https://gitlab.dkrz.de/auto-icon/auto-icon.

Declaration on the use of AI: Redaction of the text was aided by generative AI.

Bibliography

[ABM+26] B. Andela, B. Broetz, L. de Mora, N. Drost, V. Eyring, N. Koldunov, A. Lauer, B. Mueller, V. Predoi, M. Righi, M. Schlund, J. Vegas-Regidor, K. Zimmermann, K. Adeniyi, E. Arnone, O. Bellprat, P. Berg, C. Billows, E. Blockley, L. Bock, A. Bodas-Salcedo, L.-P. Caron, N. Carvalhais, I. Cionni, N. Cortesi, S. Corti, B. Crezee, E. L. Davin, P. Davini, C. Deser, F. Diblen, D. Docquier, L. Dreyer, C. Ehbrecht, P. Earnshaw, T. Geddes, B. Gier, G. Castellani, E. Gillett, N. Gonzalez-Reviriego, P. Goodman, S. Hagemann, C. Hardacre, J. von Hardenberg, B. Hassler, H. Heuer, E. Hogan, A. Hunter, C. Kadow, S. Kindermann, S. Koirala, B. Kuehbacher, L. Lledó, Q. Lejeune, V. Lembo, B. Little, S. Loosveldt-Tomas, R. Lorenz, T. Lovato, V. Lucarini, E. Malinina, F. Massonnet, C. W. Mohr, P. Amarjiit, N. Parsons, N. Pérez-Zanón, A. Phillips, M. Proft, J. Russell, M. Sandstad, A. Sellar, D. Senftleben, F. Serva, J. Sillmann, T. Stacke, R. Swaminathan, K. Tomkins, V. Torralba, K. Weigel, K. Schulze, E. Sarauer, C. Roberts, P. Kalverla, S. Alidoost, S. Verhoeven, B. Vreede, S. Smeets, A. Soares Siqueira, R. Kazeroni, J. Potter, F. Winterstein, R. Beucher, J. Kraft, L. Ruhe, P. Bonnet, G. Munday, F. Chun, H. Ellis. ESMValTool. Mar. 2026.
doi:10.5281/zenodo.3401363
https://github.com/ESMValGroup/ESMValTool/

[Air] Airflow contributors. Airflow.
https://archive.softwareheritage.org/swh:1:dir:6170b7b37e5c17a5368fb503e9c99c23c714017a

[Bah11] A. Bahra. Managing work flows with ecFlow. *ECMWF Newsletter* 129:30–32, Oct. 2011.
doi:10.21957/nr843dob
https://www.ecmwf.int/en/elibrary/80182-managing-work-flows-ecflow

[BGJ+20] S. Basart, C. P. García-Pando, O. Jorba, F. Benincasa, M. Olid, K. Serradell, G. Montanyé, E. Werner. Upgrading the monarchoperational forecast: deployment protocoland dust emission upgrades over namee. techreport, Barcelona Dust Regional Center, Nov. 2020. Accessed 10 August 2026.
https://dust.aemet.es/resolveuid/03dc3659f64d4dda9fa04e51cc4574f8

[BK] A. Baer, M. Klose. auto-icon configuration for the paper Hoshyaripour et al. 2026, GMD.
https://gitlab.dkrz.de/auto-icon/auto-icon/-/tags/Baer%2FGMD-Hoshyaripour-2026

[BTJ+] A. Baer, O. Tintó Prims, F. Jakub, M. Castrillo, R. Redl, R. Caspart, M. Klose. auto-icon: seamless management of ICON model runs with Autosubmit. Journal of Open Source Software: Under Review.
https://joss.theoj.org/papers/03249d4d9c3a812767e73f47beb3c8ff

[BTJ+25] A. Baer, O. Tintó Prims, F. Jakub, M. Castrillo, R. Redl, R. Caspart, M. Klose. auto-icon: seamless management of ICON model runs with Autosubmit. 2025.
doi:10.5281/ZENODO.11482663

[Cop23] Copernicus Climate Change Service. Complete ERA5 global atmospheric reanalysis. 2023.
doi:10.24381/CDS.143582CF

[DCF+17] P. Di Tommaso, M. Chatzou, E. Floden, P. Prieto Barja, E. Palumbo, C. Notredame. Nextflow enables reproducible computational workflows. Apr. 2017.
doi:10.1038/nbt.3820
https://nextflow.io

[DEB+22] E. Di Tomaso, J. Escribano, S. Basart, P. Ginoux, F. Macchia, F. Barnaba, F. Benincasa, P.-A. Bretonnière, A. Buñuel, M. Castrillo, E. Cuevas, P. Formenti, M. Gonçalves, O. Jorba, M. Klose, L. Mona, G. Montané Pinto, M. Mytilinaios, V. Obiso, M. Olid, N. Schutgens, A. Votsis, E. Werner, C. Pérez García-Pando. The MONARCH high-resolution reanalysis of desert dust aerosol over Northern Africa, the Middle East and Europe (2007–2016). *Earth System Science Data* 14(6):2785–2816, 2022.
doi:10.5194/essd-14-2785-2022

[Deu] Deutscher Wetterdienst. ICCARUS 2024. Accessed 04 August 2026. https://www.dwd.de/EN/specialusers/research_education/seminar/2024/iccarus2024/iccarus_2024_en_node.html

[FPM+22] B. Früh, R. Potthast, W. Müller, P. Korn, S. Brienen, K. Fröhlich, J. Helmert, M. Köhler, S. Lorenz, T. V. Pham, H. Pohlmann, L. Schlemmer, R. Schnur, J.-P. Schulz, C. Sgoff, B. Vogel, R. Wirth, G. Zängl. ICON-Seamless, the development of a novel Earth System Model based on ICON for time scales from weather to climate. *EMS Annual Meeting 2022*, June 2022. doi:10.5194/ems2022-292

[GAB+21] P. Gierz, M. Andrés-Martínez, D. Barbi, N. Wieters, L. Cristini, J. Streffing, S. Wahl, D. Ural, J. Kjellsson, F. Chegini, S. Khosravi. ESM-Tools. 2021. doi:10.5281/ZENODO.3737927

[GHE+95] A. Guenther, C. N. Hewitt, D. Erickson, R. Fall, C. Geron, T. Graedel, P. Harley, L. Klinger, M. Lerdau, W. A. Mckay, T. Pierce, B. Scholes, R. Steinbrecher, R. Tallamraju, J. Taylor, P. Zimmerman. A global model of natural volatile organic compound emissions. *Journal of Geophysical Research: Atmospheres* 100(D5):8873–8892, 1995. doi:https://doi.org/10.1029/94JD02950 https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/94JD02950

[GPG+12] P. Ginoux, J. M. Prospero, T. E. Gill, N. C. Hsu, M. Zhao. Global-scale attribution of anthropogenic and natural dust sources and their emission rates based on MODIS Deep Blue aerosol products. *Reviews of Geophysics* 50(3), 2012. doi:https://doi.org/10.1029/2012RG000388 https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2012RG000388

[Han] Hans-Ertel-Zentrum für Wetterforschung. ICON-SmART. Accessed 19 May 2026. https://www.hans-ertel-zentrum.de/Projekte/ICON-SmART.html

[HBB+26] G. A. Hoshyaripour, A. Baer, S. Bierbauer, J. Bruckert, D. Brunner, J. Förstner, A. Hamzehloo, V. Hanft, C. Keller, M. Klose, P. Kumar, P. Ludwig, E. Metzner, L. Muth, A. Pauling, N. Porz, M. Ramezani Ziarani, T. Reddmann, L. Reißig, R. Ruhnke, K. Satitkovitchai, A. Seifert, M. Sinnhuber, M. Steiner, S. Versick, H. Vogel, M. Weimer, S. Werchner, C. Hoose. The atmospheric composition component of the ICON modeling framework: ICON-ART version 2025.10. *Geoscientific Model Development* 19(4):1645–1681, Feb. 2026. doi:10.5194/gmd-19-1645-2026

[HSG22] A. Heil, M. Schultz, C. Granier. AeroCom II emission data. 2022. https://aerocom-classic.met.no/DATA/download/emissions/AEROCOM-II-ACCMIP/

[HTKH04] N. Hsu, S.-C. Tsay, M. King, J. Herman. Aerosol properties over bright-reflecting source regions. *IEEE Transactions on Geoscience and Remote Sensing* 42(3):557–569, 2004.
doi:10.1109/TGRS.2004.824067

[Jen] Jenkins contributors. Jenkins.
https://github.com/jenkinsci/jenkins

[JGP23] C. Jakob, A. Gettelman, A. Pitman. The need to operationalize climate modelling. *Nature Climate Change* 13(11):1158–1160, Nov. 2023.
doi:10.1038/s41558-023-01849-4

[JSJ] J. Jucker, M. Stellio, M. Jähn. Zonda Request.
https://github.com/C2SM/zonda-request

[KBJ+22] P. Korn, N. Brüggemann, J. H. Jungclaus, S. J. Lorenz, O. Gutjahr, H. Haak, L. Linardakis, C. Mehlmann, U. Mikolajewicz, D. Notz, D. A. Putrasahan, V. Singh, J. von Storch, X. Zhu, J. Marotzke. ICON-O: The Ocean Component of the ICON Earth System Model—Global Simulation Characteristics and Local Telescoping Capability. *Journal of Advances in Modeling Earth Systems* 14(10), Oct. 2022.
doi:10.1029/2021ms002952

[KJA+21] M. Klose, O. Jorba, M. G. Ageitos, J. Escribano, M. L. Dawson, V. Obiso, E. D. Tomaso, S. Basart, G. M. Pinto, F. Macchia, P. Ginoux, J. Guerschman, C. Prigent, Y. Huang, J. F. Kok, R. L. Miller, C. P. García-Pando. Mineral dust cycle in the Multiscale Online Nonhydrostatic AtmospheRe CHemistry model (MONARCH) Version 2.0. *Geoscientific Model Development* 14(10):6403–6444, oct 2021.
doi:10.5194/gmd-14-6403-2021

[KMF+14] J. F. Kok, N. M. Mahowald, G. Fratini, J. A. Gillies, M. Ishizuka, J. F. Leys, M. Mikami, M.-S. Park, S.-U. Park, R. S. Van Pelt, T. M. Zobeck. An improved dust emission model – Part 1: Model description and comparison against measurements. *Atmospheric Chemistry and Physics* 14(23):13023–13041, 2014.
doi:10.5194/acp-14-13023-2014
https://acp.copernicus.org/articles/14/13023/2014/

[KMSe] L. Kornblueh, R. Mueller, U. Schulzweida, et al. Climate Data Operators.
https://code.mpimet.mpg.de/projects/cdo

[Kor18] P. Korn. A structure-preserving discretization of ocean parametrizations on unstructured grids. *Ocean Modelling* 132:73–90, Dec. 2018.
doi:10.1016/j.ocemod.2018.10.002

[LCR+24] S. Leo, M. R. Crusoe, L. Rodríguez-Navas, R. Sirvent, A. Kanitz, P. De Geest, R. Wittner, L. Pireddu, D. Garijo, J. M. Fernández, I. Colonnelli, M. Gallo, T. Ohta, H. Suetake, S. Capella-Gutierrez, R. de Wit, B. P. Kinoshita, S. Soiland-Reyes. Recording provenance of workflow runs with RO-Crate. *PLOS ONE*

19(9):e0309210, Sept. 2024.
doi:10.1371/journal.pone.0309210

[LGAK] R. Li, C. P. García-Pando, M. G. Ageitos, M. Klose. In Preparation.

[MCU+25] M. G. Marciani, M. Castrillo, G. Utrera, M. C. Acosta, B. P. Kinoshita, F. Doblas-Reyes. Evaluating the impact of task aggregation in workflows with shared resource environments: use case for the MONARCH application. *Geoscientific Model Development* 18(23):9709–9721, Dec. 2025.
doi:10.5194/gmd-18-9709-2025

[MLP+25] W. A. Müller, S. Lorenz, T. V. Pham, A. Schneidereit, R. Brokopf, V. Brovkin, N. Brüggemann, F. Chegini, D. Dommenget, K. Fröhlich, B. Früh, V. Gayler, H. Haak, S. Hagemann, M. Hanke, T. Ilyina, J. Jungclaus, M. Köhler, P. Korn, L. Kornblueh, C. A. Kroll, J. Krüger, K. Castro-Morales, U. Niemeier, H. Pohlmann, I. Polkova, R. Potthast, T. Riddick, M. Schlund, T. Stacke, R. Wirth, D. Yu, J. Marotzke. The ICON-based Earth System Model for climate predictions and projections (ICON XPP v1.0). *Geoscientific Model Development* 18(23):9385–9415, Dec. 2025.
doi:10.5194/gmd-18-9385-2025

[MVP+16] D. Manubens-Gil, J. Vegas-Regidor, C. Prodhomme, O. Mula-Valls, F. J. Doblas-Reyes. Seamless management of ensemble climate prediction experiments on HPC platforms. In *2016 International Conference on High Performance Computing & Simulation (HPCS)*. Pp. 895–900. 2016.
doi:10.1109/HPCSim.2016.7568429

[OSM+19] H. Oliver, M. Shin, D. Matthews, O. Sanders, S. Bartholomew, A. Clark, B. Fitzpatrick, R. van Haren, R. Hut, N. Drost. Workflow Automation for Cycling Systems. *Computing in Science & Engineering* 21(4):7–21, July 2019.
doi:10.1109/mcse.2019.2906593

[OSS18] H. J. Oliver, M. Shin, O. Sanders. Cylc: A Workflow Engine for Cycling Systems. *Journal of Open Source Software* 3(27):737, July 2018.
doi:10.21105/joss.00737

[PRS+14] R. Potthast, H. Reich, C. Schraff, K. Stephan, A. Rhodin. BACY = Basic Cycling, A COSMO Data Assimilation Testbed for Research and Development. Technical report, Detuscher Wetterdienst, Mar. 2014. Accessed 11 August 2026.
download.dwd.de/pub/DWD/Forschung_und_Entwicklung/CUS2014_presentations_PDF/01_DA/05_Roland_Potthast_Hendrik_Reich/potthast_COSMO_USER_SEMINAR_2014.pdf

[RG23] B. Rockel, B. Geyer. SPICE (Starter Package for ICON-CLM Experiments). 2023.
doi:10.5281/ZENODO.5910888

[RGG+21] C. H. Reick, V. Gayler, D. Goll, S. Hagemann, M. Heidkamp, J. E. M. S. Nabel, T. Raddatz, E. Roeckner, R. Schnur, S. Wilkenskjeld. JSBACH 3 - The land component of the MPI Earth System Model: documentation of version 3.2. *Berichte zur Erdsystemforschung* 240, 2021.
doi:10.17617/2.3279802

[Sha04] Y. Shao. Simplification of a dust emission scheme and comparison with data. *Journal of Geophysical Research: Atmospheres* 109(D10), 2004.
doi:https://doi.org/10.1029/2003JD004372
https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2003JD004372

[SO18] B. Sorkhpour, R. Obermaisser. MeSViz: Visualizing Scenario-based Meta- Schedules for Adaptive Time-Triggered Systems. In *AmE 2018 - Automotive meets Electronics; 9th GMM-Symposium*. Pp. 1–6. 2018.

[TWZ00] K. E. Taylor, D. Williamson, F. Zwiers. The Sea Surface Temperature and Sea Ice Concentration Boundary Conditions for AMIP II Simulations. Technical report UCRL-MI-123395, Lawrence Livermore National Laboratory, 2000.
https://pcmdi.llnl.gov/report/ab60.html

[WHS25] P. Weiss, R. Herbert, P. Stier. ICON-HAM-lite 1.0: simulating the Earth system with interactive aerosols at kilometer scales. *Geoscientific Model Development* 18(12):3877–3894, June 2025.
doi:10.5194/gmd-18-3877-2025
https://gmd.copernicus.org/articles/18/3877/2025/

[ZRRB14] G. Zängl, D. Reinert, P. Rípodas, M. Baldauf. 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(687):563–579, jun 2014.
doi:10.1002/qj.2378