Sprint documentation #10

QUINCY in ICON-Land Sprint

Sergey Sukov[1], Jan Engel[2] and Julia Nabel[2]

[1]Jülich Supercomputing Center, Forschungszentrum Jülich GmbH, Jülich, Germany
[2] Max Planck Institute for Biogeochemistry, Jena, Germany

Contact: info@nat-esm.de

Published on 07.01.2025 on https://www.nat-esm.de/services/accepted-sprint

1 Summary

The main aim of the sprint was the stepwise, model-mode based porting of the land surface model QUINCY[1] code to GPUs, in line with the ICON OpenACC style and implementation guide. To achieve this goal, the ICON-Land Standalone Driver was first ported to GPUs, followed by running test cases where the Driver called existing GPU-implementations of JSBACH4 scientific processes.

The most important practical outcome of the sprint was the preliminary porting of the QUINCY software modules used for simulations in CANOPY model-mode, in which only biogeophysical processes are enabled. This included setting up host-device data synchronization in the infrastructure subroutines, which are called before and after restart data file I/O, regardless of the model mode.

The correctness of the work performed was confirmed by matching results when running the scientific subroutines QUINCY CANOPY model-mode with the ICON-Land Standalone Driver on both CPU and GPU.

Based on the experience gained during the sprint, the RSE additionally developed a porting guide, and a module for comparing host and device variable values using OpenACC features. This documentation and software can be used in the subsequent porting of routines required for the CANOPY model-mode which currently are still partially calculated on CPU, as well as the code for the PLANT and LAND model-modes.

2 General information

Start and end date:	January 2024 – August 2024
Intended period:	6 months
Responsible RSE:	Sergey Sukov (JSC)
Responsible scientists:	Jan Engel and Julia Nabel (MPI-BGC)

The land surface model QUINCY (Quantifying Interactions between terrestrial Nutrient CYcles and the climate System) is a state-of-the-art terrestrial biosphere model with fully coupled carbon,

[1] QUINCY (Quantifying Interactions between terrestrial Nutrient CYcles and the climate System) is a state-of-the-art terrestrial biosphere model (Thum et al., 2019). Its software implementation for CPU, written in FORTRAN with ICON-Land Domain-Specific Language macros, has been integrated into the ICON-Land framework as an alternative land surface scheme to ICON, in addition to JSBACH4.

nitrogen, phosphorus, water, and energy cycles (Thum et al., 2019). QUINCY has been developed as a standalone site-level model written in FORTRAN, following the structure, modularization design goals and particular Domain-Specific Language (DSL) extensions of the ICON-Land Framework (ICON-Land) [10, 11] as well as the ICON Programming Standard [4]. From the standpoint of the numerical algorithm structure, QUINCY is modularized to a degree that allows for the computation of schemes with different complexity:

- CANOPY mode - only biogeophysical processes are calculated, while the vegetation structure is prescribed, and vegetation dynamics are not represented.
- PLANT mode - adds vegetation biogeochemistry and dynamics processes to the CANOPY mode, but without dependence on soil biogeochemical processes.
- LAND mode (full QUINCY model) - in addition to PLANT mode includes soil biogeochemical processes.

Simulations in the LAND mode can be run with two alternative soil biogeochemistry schemes: a simple (state-of-the-art) soil biogeochemistry scheme for standard applications, or an advanced microbiologically explicit soil biogeochemistry model (Jena Soil Model; JSM) for future scientific developments.

All QUINCY scientific processes have been fully integrated into the ICON-Land (IQ: ICON-Land with QUINCY) as an alternative to the initially implemented land surface scheme JSBACH4 (IJ4: ICON-Land with JSBACH – the Jena Scheme for Biosphere Atmosphere Coupling in Hamburg; Nabel et al. 2020, Schneck et al. 2022). The ICON-Land framework supports coupled to the ICON-atmosphere model or standalone simulations (driven by re-analysis forcing). Software implementations of the land surface schemes share ICON-Land infrastructure and can be run both in coupled mode and with the ICON-Land Standalone Driver: ISQ (Standalone Driver with QUINCY) and ISJ4 (Standalone Driver with JSBACH4) configurations. Whereby QUINCY had not been tested in coupled mode when starting the sprint.

Before the sprint, ICON-Land infrastructure subroutines and JSBACH4 scientific processes had been partially ported to GPU using OpenACC and tested within coupled simulations. The ICON-Land Standalone Driver and scientific processes of QUINCY had CPU-only implementations.

3 Sprint objectives

The main aim of the sprint was the stepwise (model-mode based) porting of the IQ code to GPUs, following the established workflow used for the ICON GPU implementation with OpenACC [5]. Porting code integrated into ICON-Land is significantly simplified when debugging is based on standalone runs, furthermore, when starting the sprint, no running coupled configuration was available with the QUINCY code. Therefore, porting the Standalone Driver using the already ported scientific processes of JSBACH4 has been defined as a preparatory step. The next and most important task within the sprint was to port the CANOPY mode, where only biogeophysical processes are active, and run it on GPUs using the ICON-Land Standalone Driver. It was assumed that addressing this task will include a discussion on the optimal order of dimensions in the 4D to 5D matrices that carry QUINCY's biogeochemical pools (state variables) during runtime, as well as the necessary GPU-CPU synchronization of this data to enable restart reading and output writing. If time allowed, the plan was to proceed with the offloading of vegetation biogeochemistry and dynamics processes (PLANT mode) and the soil biogeochemical scheme (full QUINCY model or LAND mode) to GPUs.

4 Procedure and insights

4.1 Technical approach / procedure

The actual timeline of completed work is divided into four steps:

1. Porting the ICON-Land Standalone Driver to the GPU.
2. Performance testing and code profiling based on the runs of ISJ4 (CPU and GPU executables) and ISQ (CPU executable).

3. Preliminary porting of QUINCY infrastructure subroutines and scientific processes related to CANOPY mode.
4. Development of accompanying documentation and writing of the final report.

Porting the ICON-Land Standalone Driver involved addressing three tasks:

- Porting the forcing subroutines.
- Allocating device memory for global data arrays and synchronizing variable values between the CPU and GPU.
- Configuring parameters for file I/O subroutines (completed by the main developers).

The correctness of the code porting was verified by comparing the computation results obtained on the CPU and GPU (output and restart NetCDF data files). The acceptable deviation in variable values was determined based on the level of deviations in results generated by different CPU executable files (built with different compilers using different optimization flags). Program runs were repeated multiple times using different numbers of MPI processes and block sizes. During the testing and profiling step, NVTX (Nvidia Tools Extension) markers were added to the QUINCY code. Subsequently, based on graphical data and reports generated by Nvidia Nsight System tools, a list of 24 scientific processes was composed. Following this, subroutines related to the ICON-Land infrastructure, which had already been ported to the GPU, were identified and excluded from the list.

In line with the ICON code development strategy, the choice of platform (CPU or GPU) and the parallelization approach (enabling or disabling OpenMP threads) is determined at the configure stage. As a result, the built executable can only be run on a single type of device. The main idea of the third step of the sprint was to generate a MPI+OpenACC GPU-configured executable that would produce results identical to those of the CPU-executable.

The procedure for porting scientific processes related to the CANOPY mode can be formalized as follows:

1. Setting up CPU-GPU data synchronization in functions responsible for copying QUINCY specific data structures (used to represent biogeochemical pools) to multidimensional ICON arrays which are already subject to CPU-GPU data synchronization and vice versa.
2. Stepwise porting of QUINCY scientific processes:
   - adding data synchronization directives in the subroutines for setting the initial values of variables;
   - transforming the code: replacing access to structure members with direct pointer access to array cells, converting FORTRAN array operations into loops, and so forth;
   - adding OpenACC compute constructs (PARALLEL, PARALLEL LOOP, KERNELS);
   - compilation, CPU and GPU executable runs, and results comparison.

The code was ported to the GPU using OpenACC API tools, closely following the ICON OpenACC style and implementation guide. However, to reduce porting time and simplify debugging, a few style requirements were not followed in the preliminary port[2]: the ASYNC clause was not added to the compute constructs, and in some places, the KERNEL directive was used to offload computations on arrays. For the same reasons, three scientific processes were only partially ported to the GPU. One of them is expected to be replaced by another scientific process implementation in the near future. In these cases, directives for host-device data synchronization were added before and after specific subroutine calls on the CPU within the scientific processes. To simplify code debugging, an additional FORTRAN module was developed with subroutines for comparing the values of 1D to 5D array elements on the CPU and GPU.

[2] The final version is expected to be fully compliant with ICON standards.

During the porting process, special interrupt functions were added to code branches that were inactive for the test runs. If the GPU-executable reaches a non-ported code block, the program stops and outputs an error message. These functions are ignored during CPU executions.

As a result of the sprint being extended to include the porting of the ICON-Land Standalone Driver, and the RSE taking longer than expected to get to know the ICON, ICON-Land, JSBACH and QUINCY codes structure, the time available for working on QUINCY scientific subroutines was less than the originally accepted schedule. Therefore, requests for support regarding the porting of scientific processes for LAND and PLANT modes and the investigations into the optimal ordering of dimensions in QUINCY's matrices were not addressed. Based on the experience gained in the third step of the sprint, a brief guide for porting the code has been developed, which should be useful during the porting of the remaining scientific processes.

Discussions of the sprint technical details mainly took place on the DKRZ GitLab platform (JSBACH repository: feature branches, issues, as well as merge requests for code development). In addition, weekly video conferences were organized, and two personal meetings occurred between the RSE and the scientists.

4.2 General insights

After completing the first step of the sprint, the RSE conducted preliminary strong scaling and performance testing of the JSBACH4 and QUINCY software modules using ISQ on CPU and ISJ4 on CPU and GPU. All experiments were run on an R2B4 grid containing 20480 surface cells, and only the execution time of scientific processes was measured.

Strong scaling tests of ICQ and ISJ4 CPU executables (using 1 - 8 MPI processes) showed nearly linear scaling. The execution time for a single simulation step using QUINCY scientific processes was three times longer than with JSBACH4.

When comparing ISJ4 execution time on single devices --an AMD 7763 CPU (64 cores) versus an NVIDIA A100 GPU (80 GB) on the HPC Levante system at DKRZ-- the GPU exhibited a performance ratio of approximately 1:4 compared to the CPU. This means that a single GPU performed at the level of about 16 CPU cores. The relatively modest performance of the GPU runs can be attributed to the small size of the test problem, which led to idle time for some streaming multiprocessors.

Block size (nproma)	Time (seconds)
2048	0.25
4096	0.14
20480	0.04

Table 1: Execution time of JSBACH4 scientific processes on GPU: one simulation step, runscript land_jsbach_R2B4_test (disabling lakes, hd and carbon).

Table 1 shows the execution times of JSBACH4 scientific subroutines on GPU for different block sizes. The nproma parameter allows for an implicit adjustment of the GPU's occupancy level. The data shows that as the block size increases, computation time decreases, as previously idle hardware units become involved in executing the kernels. However, due to the complexity and heterogeneity of the algorithm (with many kernels that have varying levels of register usage), it is not possible to extrapolate these results to cases where the GPU is fully loaded.

Once the porting of the QUINCY model is completed, there are plans to prepare specialized scripts and experimental input data to run simulations on more detailed grids. This will enable an accurate comparison of computation times across different devices.

Further code analysis indicated that at least some QUINCY subroutines contain data-independent computational kernels, which can be executed in asynchronous streams. In theory, concurrent kernel execution will improve the efficiency of utilizing massively parallel accelerators for simulations on small grids. It is preferable if the option to launch kernels in different streams with adjustment of the optimal gang numbers and vector lengths is enabled only in case of low GPU load. Otherwise, all kernels will be launched in the same stream with the ASYNC clause to reduce overhead. Additionally, some speed-up of computations can be achieved by restructuring code blocks that currently use temporary arrays sized nproma along one of the dimensions.

Since most QUINCY subroutines roughly follow a similar structure, it was possible to formalize the porting procedure, which led to the porting guide developed during this sprint that can be found here: https://www.nat-esm.de/services/support-through-sprints/accepted-sprints.

Using a Standalone Driver has significantly simplified porting, profiling, and debugging the code, but it has not provided any advantages in terms of reducing code compilation time. ICON-Land is partially built upon ICON data types and infrastructure subroutines. Therefore, compiling and running ISJ4 and ISQ is equivalent to building an ICON executable and then launching it with specific setups. As a result, compilation and recompilation times are the same as they would be for a coupled application (from 5 to 15 minutes, depending on the source files that have been modified, the compiler settings, and the target platform).

5 Results

The key results achieved during the sprint can be summarized as follows:

1. The ICON-Land Standalone Driver was ported to GPUs.
2. Strong scaling tests and performance comparison were carried out for the CPU and GPU implementations of JSBACH4 and the CPU-only implementation of QUINCY scientific processes.
3. Preliminary porting of the QUINCY modules used for simulations in CANOPY mode has been completed, either by introducing parallel regions or via host-device data synchronization directives. Modifying the code for execution on GPUs also involves setting up host-device data synchronization within the infrastructure subroutines, which are called regardless of the model mode.
4. To support the subsequent porting of further QUINCY scientific processes, a porting guide and an additional data comparison module were developed.

6 Conclusions and Outlook

The main result of the sprint is the preliminary porting of the QUINCY software modules used for simulations in CANOPY mode. This version of the code, together with the GPU-ported ICON-Land Standalone Driver, can serve as a platform for further porting of still missing scientific processes.

Future steps include:

- Porting scientific routines which are still missing for a full port of the CANOPY mode.
- Porting scientific routines for other modes (PLANT and LAND with simple soil).
- Bringing the code into full compliance with the ICON OpenACC style and implementation guide.
- Investigating and conducting necessary steps for a GPU port of the complex soil model (the Jena Soil Model; JSM).
- Assesing the performance of the ported QUINCY code, with a test case large enough to fully load the GPUs.
- Code optimization regarding GPU usage, including joining or separating parallel regions, and considering asynchronies, as well as investigating dimension orders.
- As this sprint made use of the ICON-Land Standalone Driver, another future task will also be to investigate the behavior of the QUINCY code when running coupled to the atmosphere on GPUs.

7 References

1 Nvidia Nsight Systems: https://developer.nvidia.com/nsight-systems

2 Nvidia NVTX: https://docs.nvidia.com/nvtx/index.html

3 OpenACC API: https://www.openacc.org/

4 ICON Programming Standard: https://gitlab.dkrz.de/icon/icon/-/blob/icon-2.6.5-rc/doc/style/icon_standard.pdf

5 ICON OpenAcc style and implementation guide: https://gitlab.dkrz.de/icon/wiki/-/wikis/GPU-development/ICON-OpenAcc-style-and-implementation-guide

6 Jungclaus, J., Lorenz, S., Schmidt, H., et al. (2022). The ICON Earth System Model Version 1.0. Journal of Advances in Modeling Earth Systems, 14: e2021MS002813. doi:10.1029/2021MS002813

7 Nabel, J., Naudts, K. & Pongratz, J. (2020). Accounting for forest age in the tile-based dynamic global vegetation model JSBACH4 (4.20p7; git feature/forests) – a land surface model for the ICON-ESM. Geoscientific Model Development, 13, 185-200. doi:10.5194/gmd-13-185-2020

8 Schneck, R., Gayler, V., Nabel, J., et al. (2022). Assessment of JSBACHv4.30 as a land component of ICON-ESM-V1 in comparison to its predecessor JSBACHv3.2 of MPI-ESM1.2. Geoscientific Model Development, 15, 8581-8611. doi:10.5194/gmd-15-8581-2022

9 Thum, T., Caldararu, S., Engel, J., et al. (2019): A new model of the coupled carbon, nitrogen, and phosphorus cycles in the terrestrial biosphere (QUINCY v1.0; revision 1996), Geosci. Model Dev., 12, 4781–4802, https://doi.org/10.5194/gmd-12-4781-2019

10 ICON-Land Framework and DSL extensions: https://gitlab.dkrz.de/jsbach/jsbach/-/wikis/Documentation/ICON-Land-framework/

11 https://icon-model-org.gitlab-pages.dkrz.de/website/land/land.html#ref-land