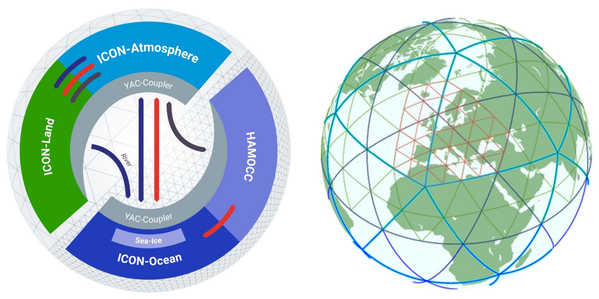
Simulating Aquaplanet Using ICON With A GT4Py DSL Dynamical Core

C. Müller[3], N. Burgdorfer[3], E. Cossevin[3], T. Ehrengruber[2] , N. Farabullini[1], A. Gopal[1], L. Groner[2], R. Häuselmann[2], D. Hupp[3], J. Jucker[1], P. Kardos[1], S. Kellerhals[1], D. Leutwyler[3], M. Luz[1], C. Osuna[3], E.G. Paredes[1], P. Pothapakula[1], M. Röthlin[3], F. Thaler[2], H. Vogt[2], B. Weber[3], C. Zeman[1], M. Bianco[2], A. Dipankar[1] , X. Lapillonne[3]

[1] Institute for Atmospheric and Climate Science (IAC), ETH Zurich
[2] Swiss National Supercomputing Center (CSCS)
[3] Federal Office for Meteorology and Climatology, MeteoSwiss

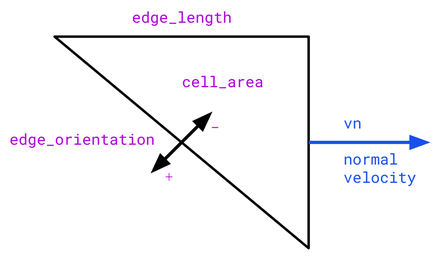
ICON EXCLAIM

- The ICON (ICOsahedral Non-hydrostatic) modeling framework[1] has been developed by the Deutscher Wetterdienst (DWD) and the Max Planck Institute for Meteorology (MPI-M)
- Current ICON partners: DWD, MPI-M, KIT, DKRZ, C2SM
- EXCLAIM - an open ETH Zurich project that aims to develop a Python framework based on the ICON model, that is capable of running kilometer-scale climate simulations at both regional and global scales[2]



GT4Py - A Domain Specific Language (DSL)

GT4Py[3] is a DSL compiler framework for weather and climate modeling. Stencil code is written in an abstracted Python based language and executed with high performance on different machine architectures, e.g. CPUs and GPUs.



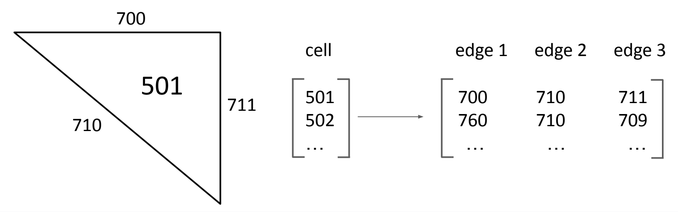

```
geofac_div(cell_idx,1:3) =

edge_length(1:3) *
edge_orientation(cell_idx,1:3) /
cell_area(cell_idx)
```

Divergence operator in GT4Py:

```
@field_operator
def divergence(
 vn:Field[[EdgeDim,KDim], float], geofac_div:Field[[CellDim,C2EDim], float],
) -> Field[[CellDim, KDim], float]:
   return neighbor_sum(vn(C2E) * geofac_div, axis=C2EDim)
```

```
neighbor sum is shorthand for:   vn(C2E(1)) * geofac div(1) +
                                 vn(C2E(2)) * geofac div(2) +
                                 vn(C2E(3)) * geofac_div(3)
```

C2E (CellToEdge) maps each cell to its neighboring edges:



GT4Py vs Fortran Source Code

Divergence in GT4Py:

```
neighbor_sum(vn(CellToEdge) * geofac_div, axis=CellToEdgeDim)
```

Divergence in Fortran:

```
DO jb = i_startblk,i_endblk

  CALL get_indices_c(p_patch, jb, i_startblk, i_endblk, &
    i_startidx, i_endidx, rl_start, rl_end)

  DO jk = 1, nlev
    DO jc = i_startidx, i_endidx

      div(jc,jk) = p_nh_prog%vn(ieidx(jc,jb,1),jk,ieblk(jc,jb,1))*p_int%geofac_div(jc,1,jb) + &
                   p_nh_prog%vn(ieidx(jc,jb,2),jk,ieblk(jc,jb,2))*p_int%geofac_div(jc,2,jb) + &
                   p_nh_prog%vn(ieidx(jc,jb,3),jk,ieblk(jc,jb,3))*p_int%geofac_div(jc,3,jb)
    ENDDO
  ENDDO
ENDDO
```

Comparison:

- No loops or explicit indices
- No blocking (compiler handles this internally as optimization)
- No explicit iteration over neighbors (still possible in GT4Py)
- No `!ACC` or `!OMP` directives

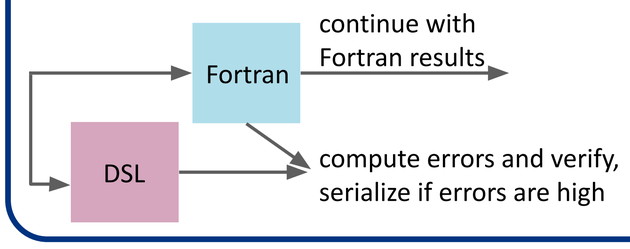
Fortran Integration and Verification with ICON Liskov

ICON Liskov is a directive-based preprocessor which parses comments and substitutes them with code, facilitating the integration of the GT4Py generated code into the ICON model.

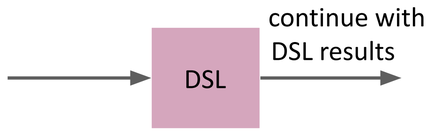
```
!$DSL START STENCIL(name=mo_nh_diffusion_stencil_10; w=p_nh_prog%w(:,:,1); diff_multfac_n2w=diff_multfac_n2w(:); &
!$DSL     cell_area=p_patch%cells%area(:,1); z_nabla2_c=z_nabla2_c(:,:,1); vertical_lower=2; &
!$DSL     vertical_upper=nrdmax(jg); horizontal_lower=i_startidx; horizontal_upper=i_endidx)
DO jk = 2, nrdmax(jg)
  DO jc = i_startidx, i_endidx
    p_nh_prog%w(jc,jk,jb) = p_nh_prog%w(jc,jk,jb) + diff_multfac_n2w(jk) * p_patch%cells%area(jc,jb) * z_nabla2_c(jc,jk,jb)
  ENDDO
ENDDO
!$DSL END STENCIL(name=mo_nh_diffusion_stencil_10)
```

ICON Liskov generates code in two modes, verification and substitution:

Verification mode: execute both DSL and Fortran versions

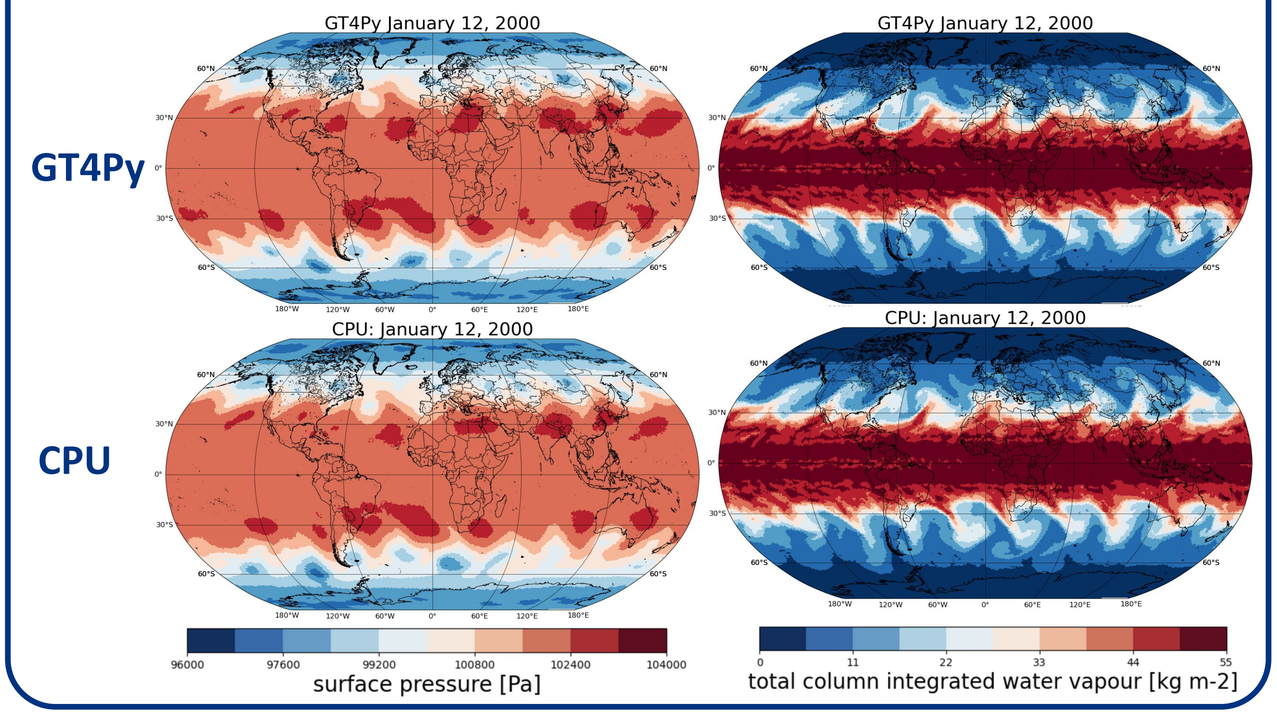


Substitution mode: only execute the DSL version



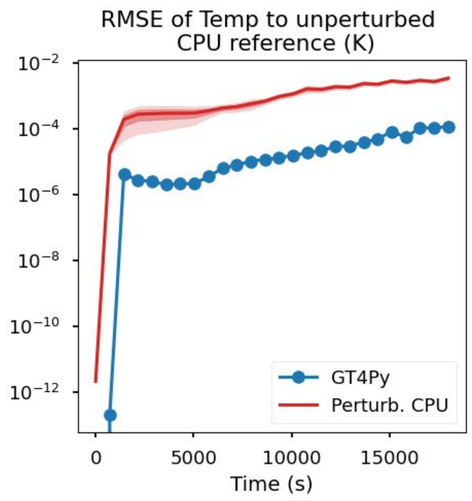
Preliminary Results for Aquaplanet

We compare preliminary results from the GT4Py dynamical core with the reference ICON model on the CSCS *Piz Daint*[4] cluster. Below we show the plots of surface pressure and total water vapor from two Aquaplanet experiments performed on a global domain with 80 km grid resolution (R02B05), and run for 12 days starting from January 01, 2000.



Model Verification

To check model correctness, the RMS error growth in the GT4Py Dynamical core (relative to the unperturbed CPU reference) is compared against 20 CPU ensembles where initial perturbations of $O(10^{-14})$ are introduced.



Ongoing and Future Efforts

Currently working on releasing the *alpha* version of the GT4Py dynamical core to EXCLAIM users and partners.

A more rigorous medium-term verification of the preliminary GT4Py dycore results, using an ensemble statistical methodology[5], is underway, alongside longer-term scientific validation.

Parallel efforts to build an experimental ICON driver written entirely in Python, as a replacement for the present FORTRAN driver, are currently a big focus.

References

[1] Zängl, Günther, et al., *Quarterly Journal of the Royal Meteorological Society* 141.687 (2015): 563-579
[2] EXCLAIM website - exclaim.ethz.ch
[3] GT4Py: GridTools for Python - github.com/GridTools/gt4py
[4] Piz Daint: a CSCS High-Performance Computing Cluster cscs.ch/computers/piz-daint/
[5] Zeman, C. and Schär, C. Geoscientific Model Development (2022)

C2SM Center for Climate Systems Modeling
CSCS
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Home Affairs FDHA
Federal Office of Meteorology and Climatology MeteoSwiss
MeteoSwiss
SDSC
Empa
Materials Science and Technology