DKRZ
DEUTSCHES
KLIMARECHENZENTRUM

Max-Planck-Institut
für Meteorologie

The Coupling Library YAC

Moritz Hanke (DKRZ), René Redler (MPI-M), and Nils-Arne Dreier(DKRZ)

YAC-Team

- Main developer:
  - Moritz Hanke (DKRZ)
- With contributions from:
  - René Redler (MPI-M)
  - Nils-Arne Dreier (DKRZ)
  - Teresa Holfeld (MPI-M, student assistant)
  - Maxim Yastremsky (MPI-M, student assistant)
  - Thomas Jahns (DKRZ)
  - Uwe Schulzweida (MPI-M)
  - Hendryk Bockelmann (DKRZ)
  - Jörg Behrens (DKRZ)
  - Sergey Kosukhin (MPI-M)

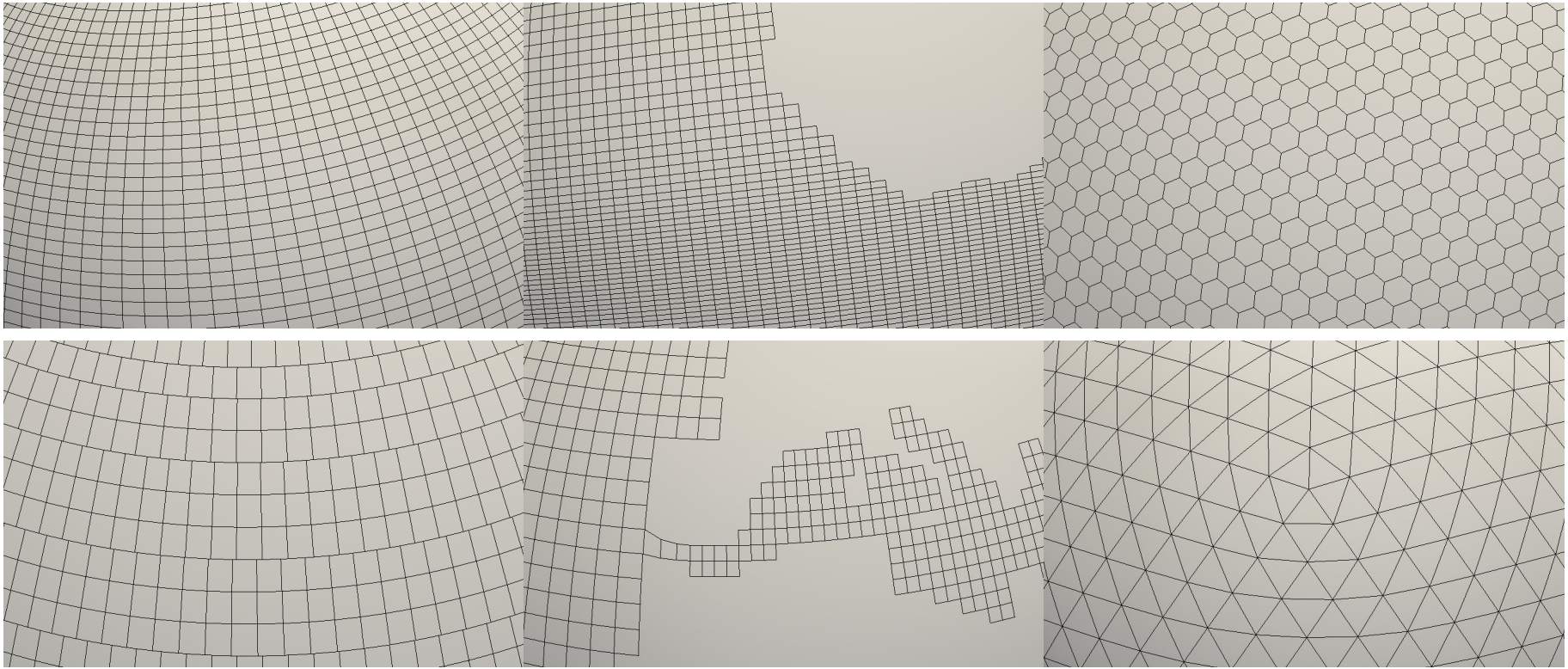
Introduction

- What does a coupler do?
  - exchange of data between independent components (e.g. atmosphere and ocean) at predefined time intervals
  - components can use different grids
    - coupler takes care of regridding between both grids
  - components can have different decompositions
    - coupler takes care of data redistribution
  - components can have different exchange periods
    - coupler takes care of matching of the exchanges and data aggregation if necessary

Introduction

- Library linked to components
  - Written in C
  - Unit tests cover 99.5% of the code
  - C-, Fortran-, and Python-Interface
- Supports all common grid types
- Provides various 2D-interpolation schemes
  - All computation on the unit sphere using cartesian coordinates
- Parallel online weight computation
- Licenced under BSD 3-Clause License
- Used in official ICON-release (but developed independently)
- runs on Piz Daint[1], JUWELS[2], Levante[3], LUMI[4], MAC OS, and Linux systems with little to no porting effort

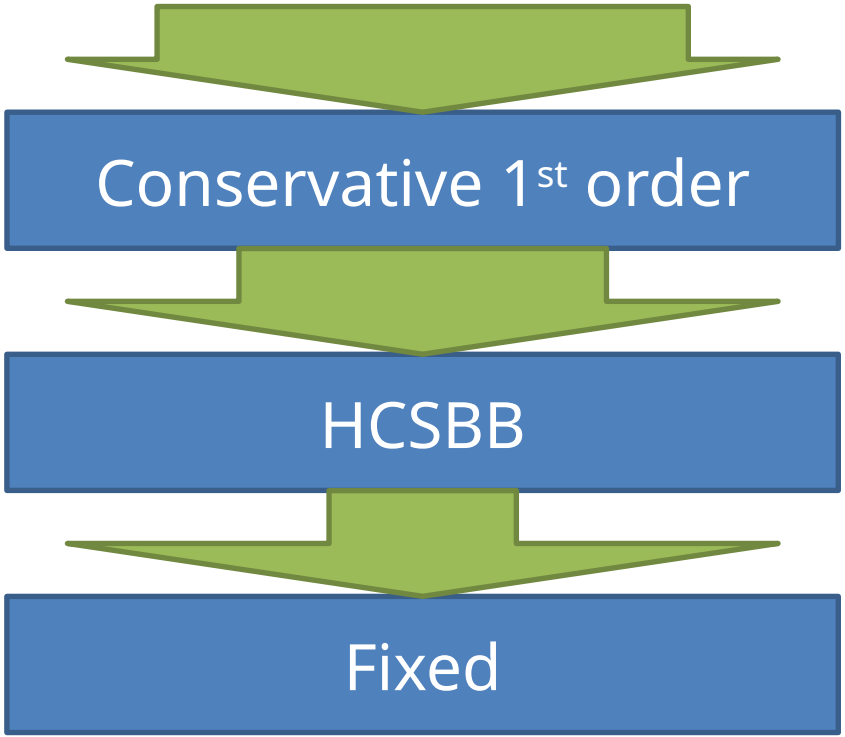
Supported grids

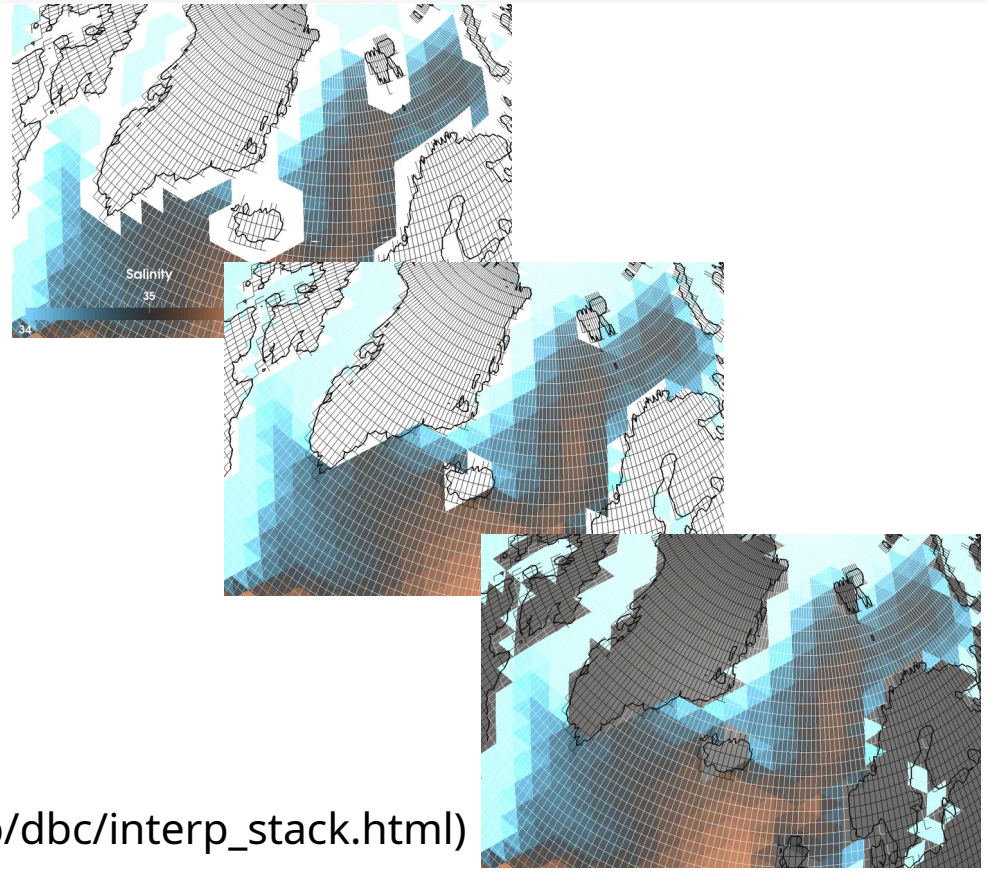


Interpolations

- 2D inter- and extrapolation
  - Various methods with different properties
  - All grid combinations are supported
- Fields can be defined on cells, vertices, or edges

Interpolation Stack



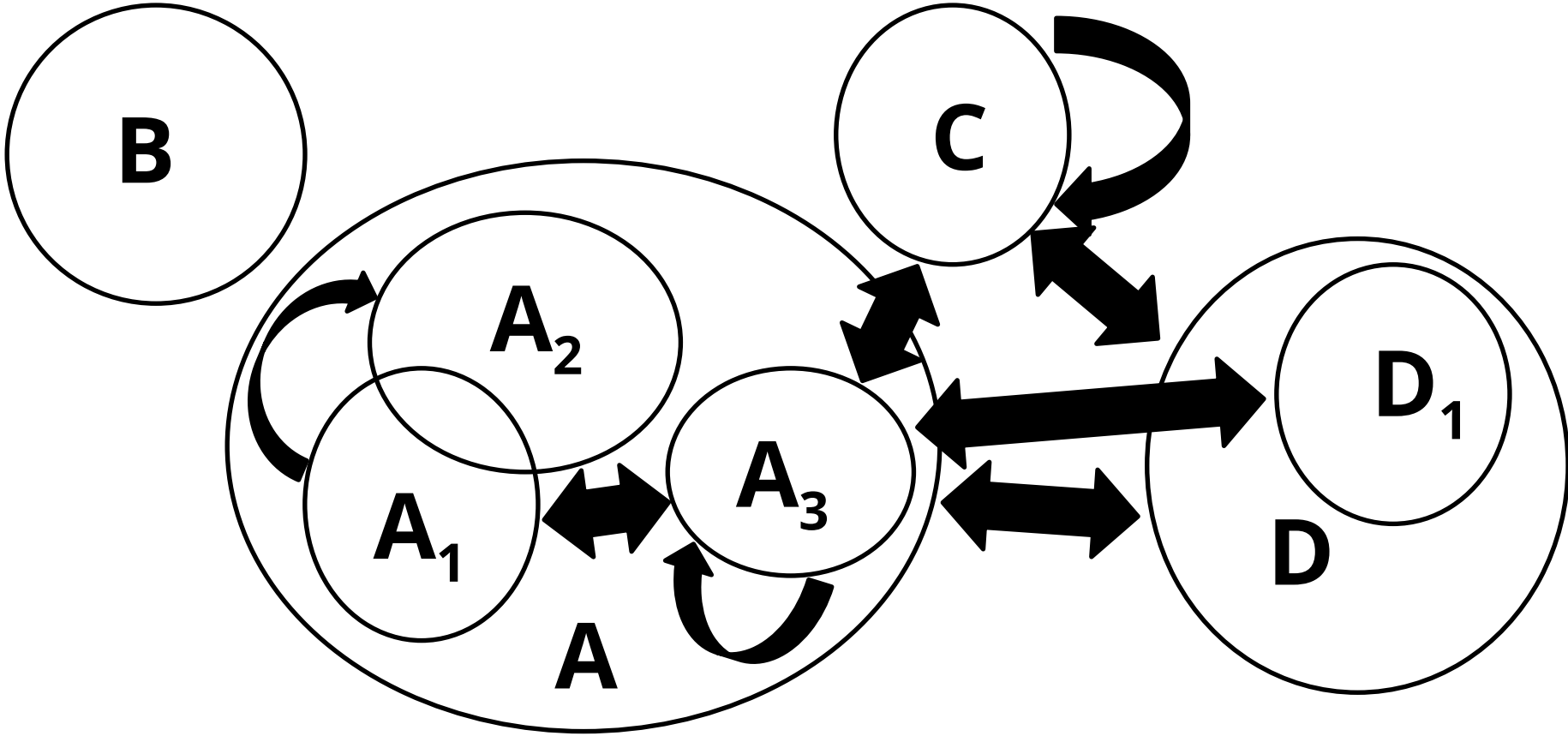


(https://dkrz-sw.gitlab-pages.dkrz.de/yac/db/dbc/interp_stack.html)

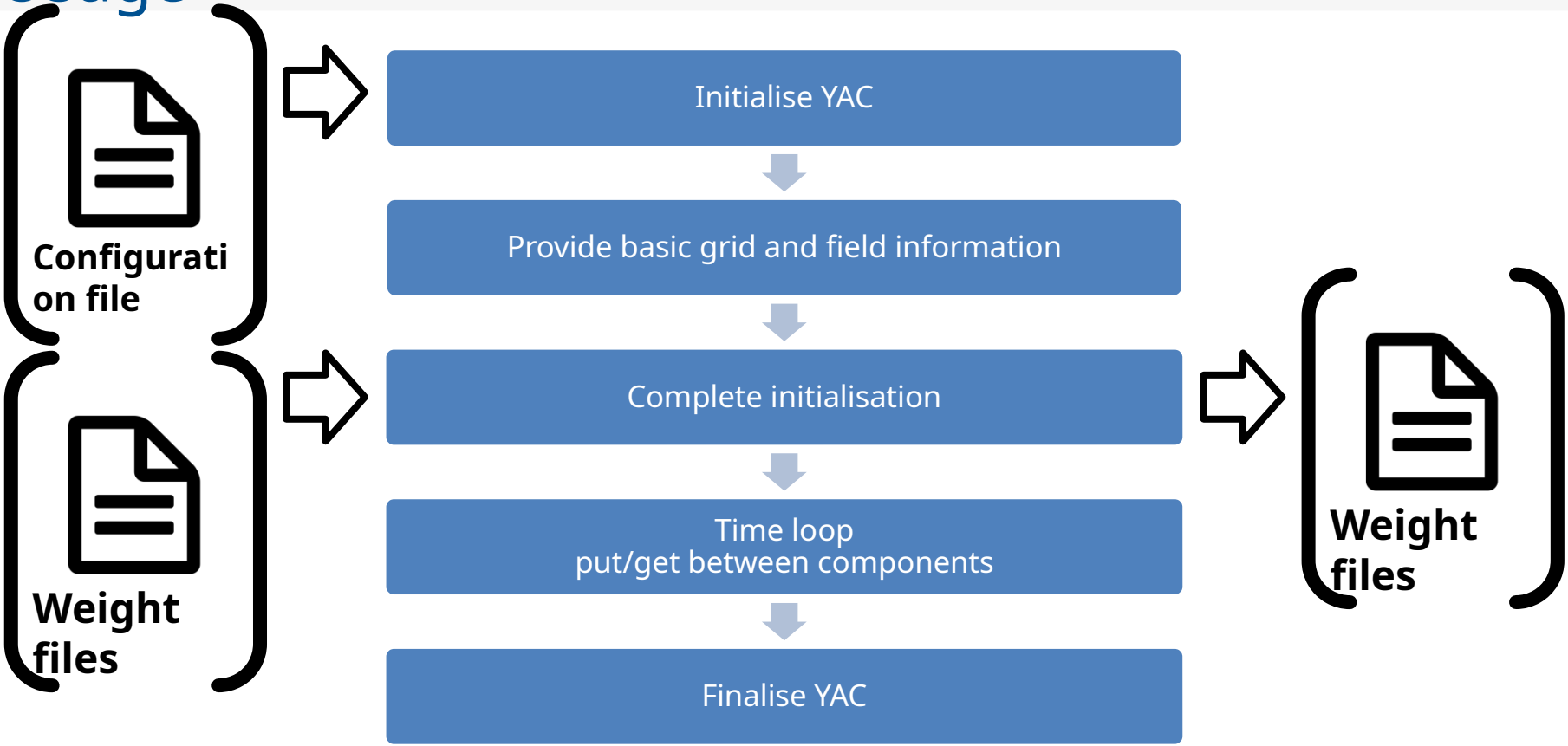
Interpolation Quality

- Benchmark on regridding quality by CERFACS in the frame of ISENES-Project
  - “This benchmark leads us to conclude that YAC, ESMF, and XIOS can all three be considered as high-quality regridding libraries [...]”
  - Valcke, S.; Piacentini, A.; Jonville, G. Benchmarking Regridding Libraries Used in Earth System Modelling. Math. Comput. Appl. 2022, 27, 31. https://doi.org/10.3390/mca27020031

Supported coupling configurations



Usage


Configuration file
Weight files
Initialise YAC
Provide basic grid and field information
Complete initialisation
Time loop
put/get between components
Finalise YAC
Weight files

Fortran example

```
USE yac

CALL yac_finit()
CALL yac_fdef_calendar(&
   YAC_PROLEPTIC_GREGORIAN)
```

Fortran example

```
CALL yac_fread_config_yaml( &
   "coupling.yaml")
```

Fortran example

```
INTEGER :: comp_id
INTEGER :: comp_comm

CALL yac_fdef_comp( &
  “atmosphere”, comp_id)
CALL yac_fget_comp_comm( &
  comp_id, comp_comm)
```

Fortran example

```
INTEGER, PARAMETER :: nbr_vertices = …
INTEGER, PARAMETER :: nbr_cells = …
INTEGER, PARAMETER :: nbr_vertices_per_cell = …
REAL, ALLOCATABLE :: vertex_lon(nbr_vertices)
REAL, ALLOCATABLE :: vertex_lat(nbr_vertices)
INTEGER, ALLOCATABLE :: &
    cell_to_vertex(nbr_vertices_per_cell,nbr_cells)

INTEGER :: grid_id

CALL yac_fdef_grid( &
    “atmosphere_grid”, nbr_vertices, nbr_cells, &
    nbr_vertices_per_cell, vertex_lon, vertex_lat, &
    cell_to_vertex, grid_id)
```

Fortran example

```
REAL, ALLOCATABLE :: cell_lon(nbr_cells)
REAL, ALLOCATABLE :: cell_lat(nbr_cells)

INTEGER :: cell_point_id

CALL yac_fdef_points( &
   grid_id, nbr_cells, &
   YAC_LOCATION_CELL, &
   cell_lon, cell_lat, &
   cell_point_id)
```

Fortran example

```
INTEGER, PARAMETER :: collection_size = …

INTEGER :: taux_field_id

CALL yac_fdef_field( &
   “TAUX”, comp_id, &
   (/cell_point_id/), 1, &
   collection_size, “600”, &
   YAC_TIME_UNIT_SECOND, &
   taux_field_id)
```

Fortran example

```
CALL yac_fenddef()
```

Fortran example

```
REAL :: taux_buffer(nbr_cells, &
                            collection_size)
INTEGER :: info, error

DO t = 1, ntimes
   …
   CALL yac_fput( &
      taux_field_id, nbr_cells, &
      collection_size, taux_buffer, &
      info, error)
   …
END DO
```

Fortran example

```
REAL :: taux_buffer(nbr_cells, &
                            collection_size)
INTEGER :: info, error

DO t = 1, ntimes
   …
   CALL yac_fget( &
      taux_field_id, nbr_cells, &
      collection_size, taux_buffer, &
      info, error)
   …
END DO
```

Fortran example

```
CALL yac_ffinalize()
```

End

- Questions?
- Download: https://gitlab.dkrz.de/dkrz-sw/yac
- Documentation: https://dkrz-sw.gitlab-pages.dkrz.de/yac/
- References
  - M. Hanke, R. Redler, T. Holfeld und M. Yastremsky, 2016: YAC 1.2.0: new aspects for coupling software in Earth system modelling. Geoscientific Model Development, 9, 2755-2769, https://doi.org/10.5194/gmd-9-2755-2016
  - M. Hanke und R. Redler, 2019: New features with YAC 1.5.0. Reports on ICON, No 3. https://doi.org/10.5676/DWD_pub/nwv/icon_003
  - E. Kritsikis, M. Aechtner, Y. Meurdesoif, and T. Dubos: Conservative interpolation between general spherical meshes, Geosci. Model Dev., 10, 425–431, https://doi.org/10.5194/gmd-10-425-2017, 2017
  - Xiaoyu Liu, Larry L. Schumaker, Hybrid Bézier patches on sphere-like surfaces, Journal of Computational and Applied Mathematics, Volume 73, Issues 1–2, 1996, Pages 157-172, ISSN 0377-0427, https://doi.org/10.1016/0377-0427(96)00041-6