Issue
003

Reports on ICON

DOI: 10.5676/DWD_pub/nwv/icon_003
ISSN: 2628-4898

New features with YAC 1.5.0

Moritz Hanke and René Redler

April 2019

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

Max-Planck-Institut
für Meteorologie

DOI: 10.5676/DWD_pub/nwv/icon_003
ISSN: 2628-4898

Publisher

Deutscher Wetterdienst
Business Area “Research and Development”
Frankfurter Straße 135
63067 Offenbach
www.dwd.de

Editors

Sebastian Rast, MPI-M
sebastian.rast@mpimet.mpg.de
René Redler, MPI-M
rene.redler@mpimet.mpg.de
Daniel Reinert, DWD,
daniel.reinert@dwd.de
Daniel Rieger, DWD,
daniel.rieger@dwd.de
Florian Prill, DWD,
florian.prill@dwd.de

New features with YAC 1.5.0

Moritz Hanke[1] and René Redler[2]

(1) Deutsches Klimarechenzentrum, (2) Max-Planck-Institut für Meteorologie

This document refers to release icon-2.5.0.

Abstract

The YAC library has been developed as a joint initiative between the German Climate Computing Center (DKRZ) and the Max Planck Institute for Meteorology (MPI-M) to realise the coupling of Earth system model components. While targeting ICON in the first place the library can be used to exchange any 2-dimensional field between a pair of source and target grids defined on a sphere. The software provides a parallel 2-dimensional neighbourhood search, interpolation, and communication for the coupling between any two model components and offers flexible coupling of physical fields defined on regular and irregular grids on the sphere without a priory assumptions about the particular grid structure or grid element types.

In this paper, we describe several new aspects of the YAC library that have been developed since it was first published in 2016. It is currently used for the coupling of model components in two global configurations of the ICON model, a version with 5 km horizontal resolution in both atmosphere and ocean and a coarser resolution using roughly 160 km in the atmosphere and 40 km in the ocean.

Keywords: Coupling, ICON-ESM

Correspondence to: Moritz Hanke (hanke@dkrz.de)

1. From YAC 1.2 to YAC 1.5

For a general introduction to coupling in the context of Earth system modelling we refer the reader to Hanke et al. (2016) and the literature cited therein. In this note, we concentrate on the development of the YAC software after it was first published by Hanke et al. (2016), and we only briefly recapture some basic features of YAC here.

YAC is programmed as a library and provides a parallelised on-line neighbourhood search, interpolation and communication for the exchange of physical 2-dimensional fields which are defined on regular or irregular grids on the sphere. For ICON users YAC is available as part of the ICON git repository. While YAC was programmed with the ICON software in mind, it is sufficiently general in functionality and design to allow for the coupling outside of ICON. The coupling (or interpolation) is performed on a per-field basis. The interpolation methods and the exchange periods for all fields are defined in an Extendable Markup Language (XML) configuration file, which is read during the initialisation phase of a model run.

With version 1.2.0 we provide for the first time a fully functional implementation of YAC suitable for the needs of Earth system models, and the first version that was used in the coupled ICON-based Earth System Model (ICON-ESM). In particular version 1.2.0 provides

- 1^{st} order conservative remapping
- patch recovery (Zienkiewicz and Zhu, 1992, Gu et al., 2003, Khoei and Gharehbaghi, 2007)
- n-nearest-neighbour
- average
- file
- fixed-value

as interpolation methods. A unique feature compared to all other currently available coupling solutions is the option to combine interpolation methods into the interpolation stack. Needless to say that for each field a different interpolation stack can be provided.

All interpolation methods work only on the overlapping part between source and target grids. If a source grid contains holes - geographical regions where no grid cells are provided - any target cells that are located in such regions will not receive any values.

Since the release of version 1.2.0 we did a lot of the usual code clean up and bug fixing and added more functionality. Most visible to the user, with YAC version 1.5 we now provide

- 2^{nd} order conservative remapping (Kritsikis et al., 2017)
- hybrid cubic spherical Bernstein-Bézier patch (Alfeld et al., 1996, Liu and Schumaker, 1996)
- source to target mapping
- radial basis functions (Reinheimer, 2018)

as additional interpolation methods, which we describe in more detail in section 2.

To improve the internal functionality and user flexibility, we

- redesigned the n-nearest-neighbour interpolation method,
- significantly improved the performance of the sphere part algorithm,
- improved support for various coupling component to process configurations,
- provide optional mapping of interpolated data on the target instead of source processes,
- handle coordinates internally in the Cartesian space to avoid unnecessary forward- and backward conversion,

all of which is described further in Sec. 3. We revisit the performance of YAC in Sec. 4 and compare it to the numbers that are published by Hanke et al. (2016) for YAC 1.2.0, and briefly touch upon the optimisation of the code since this first official release. We finish this report with a few remarks about our plans in Sec. 5 and provide links and hints to further information in Sec. 6.

2. New Interpolation methods

While the provided interpolation methods of version 1.2.0 enabled us already to use YAC in the ICON-ESM, additional interpolation methods were requested. As a consequence we now provide a 2^{nd} order conservative remapping and the Hybrid cubic spherical Bernstein-Bézier patch interpolation. Especially the hydrological discharge is difficult to handle with the interpolation methods provided in version 1.2.0 and it only worked under special conditions. Source to Target mapping was implemented as response to this issue. In order to allow for a better interpolation of vector quantities like wind stress we now provide a basic implementation of the Radial Basis Function method.

2.1. 2^{nd} order conservative remapping

In addition to the existing 1^{st} order conservative remapping scheme, we implemented a 2^{nd} order conservative remapping as described by Kritsikis et al. (2017). To be able to compute the information required for the higher order method, we use a larger stencil. Therefore, this method fails for source cells that are located directly next to masked source cells or boundaries of the source grid.

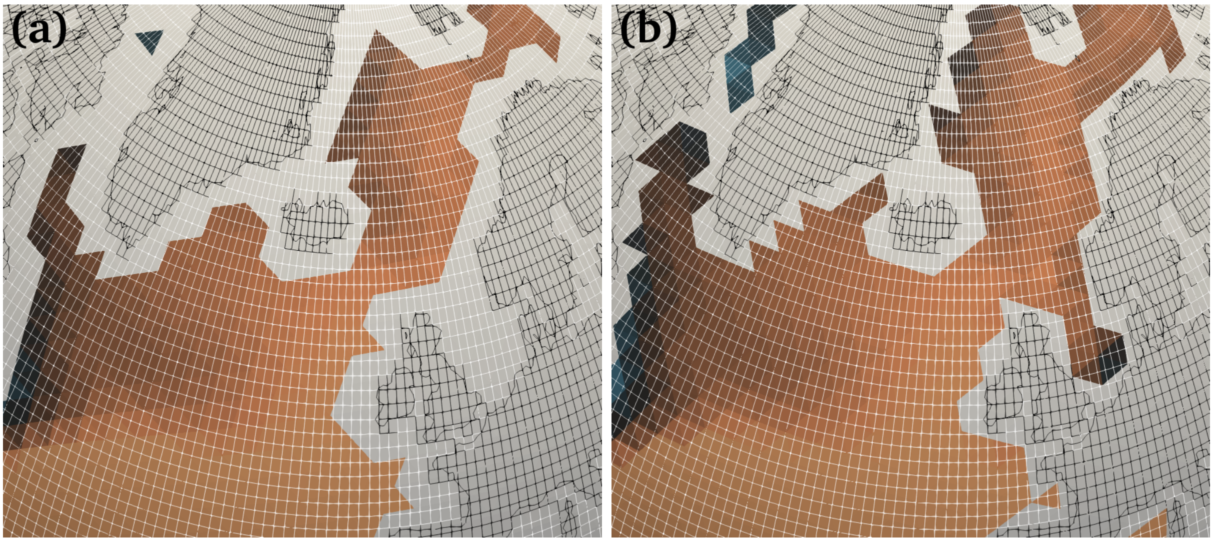

Figure 1: WOA09 January monthly mean sea surface salinity given on the white rectangle interpolated to an ICON grid (triangles) using the a YAC interpolation stack with 2^{nd} order conservative remapping **(a)** plus 1^{st} order conservative remapping **(b)**.

With Fig. 1 we reproduce Fig. 1 of Hanke et al. (2016). Sea surface salinity from the World Ocean Atlas 2009 (WOA09; Antonov et al., 2010), visible as rectangles is interpolated onto the triangular cells of an ICON R2B4 grid. Our Fig. 1 (a) here shows the interpolated results for a pure 2^{nd} order conservative remapping. For Fig. 1 (b) we extend the interpolation stack by a 1^{st} order conservative remapping and get the same coverage of interpolated cells as in Fig. 1 (a) of Hanke et al. (2016). Of course the interpolation stack can be extended further to cover all remaining target cells in this example as well (not shown). When compared to 1^{st} order conservative remapping the reduced domain coverage of the 2^{nd} order scheme is clearly visible all along the coastline, in particular in the Labrador and North Sea. Details of the picture will change depending on the source and target grids.

2.2. Hybrid cubic spherical Bernstein-Bézier patch

The Hybrid cubic spherical Bernstein-Bézier (HCSBB) patch method was implemented as response to problems observed with the Patch recovery method. It is based on Alfeld et al. (1996) and Liu and Schumaker (1996). In contrast to the Patch recovery method, the HCSBB method always produces an interpolation that has a contiguous first derivative. To achieve this, first the source grid is triangulated. Then the derivatives of the source field across the edges of the triangles are estimated. Using these, triangular patches from a blend of spherical Bernstein-Bézier polynomials are constructed, which are used for the interpolation of the target points. Compared to the Patch recovery interpolation method, this method uses a bigger stencil to compute each target point but the coverage of source cells is similar (see Fig. 2) when compared to Fig. 2 (b) of Hanke et al. (2016). Again, details of the coverage depend on the source and target grids and have to be investigated on a case by case basis.

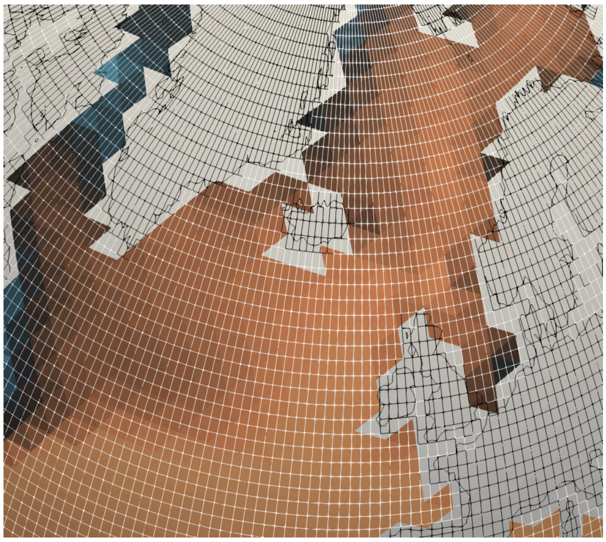

Figure 2: as Fig. 1 (a) but with the Hybrid cubic spherical Bernstein-Bézier patch interpolation method.

2.3. Source to Target mapping

The Source to Target mapping has been implemented in particular to cover the mapping of the hydrological discharge. This quantity is provided on selected cells at the coastline. In ICON, the discharge is assembled at coastal land cells and then stored on the nearest ocean cells on the donor grid. The goal of all other interpolation schemes is to generate an interpolation for target cells. In contrast to that and in order to be mass conserving and not to loose any water, the Source to Target mapping identifies the closest target cell for each donor cell. In case multiple donor cells are assigned to a single target cell, their contributions are summed up.

2.4. Radial Basis Functions

Recently we added a Radial Basis Function interpolation method, which was implemented with Reinheimer (2018) as a reference. This method uses a nearest neighbour scheme to determine N source points for each target point that is to be interpolated (default value for N is 9). We use a Gaussian kernel as the radial basis function to generate a patch for each target point using the aforementioned selected source points.

In this method source points that are closer to the target point have more impact on the interpolation result than source points that are farther away. The dependency between the influence of a source point on a target point and the distance between the two points can be adjusted by a scaling factor. A lower factor increases the influence of source points closest to that target point. We apply a normalisation to the generated patch to reduce the dependency between the scaling factor and the resolution of the source grid. By conducting a number of experiments, we determined a good default value for the scaling factor. Changing this value is only advised for experienced users. A new value should be verified by a number of tests with the relevant grid configuration. In cases where the N source points of a patch are not evenly distributed within a round area around the target point, the accuracy of this method can be significantly impeded. Such situations occur for example with Gaussian grids close the the poles.

3. Further Extensions

There are various options for mapping coupling components to the available processes of a run. For example each component can have its own processes, or components can share processes. In the OASIS_MCT paper (see Fig. 1 a in Craig et al., 2017) a complex coupling setup containing various potential component-process-placements is presented. Since YAC 1.2.0_p16, we can fully replicate this coupling setup.

Craig et al. (2017) also describe an option of the OASIS_MCT coupler, which allows the user to choose whether the interpolation weights shall be applied on the source or target processes. Inspired by the presented performance results, we also introduced this feature in YAC 1.2.0_p16.

The original implementation of the n-nearest-neighbour interpolation method contained a complex and inefficient communication scheme. Especially the scaling behaviour was poor. After a major rework a new version was introduced in YAC 1.3.0_p3, the performance and scalability of this interpolation method is now on a par with the other methods. In addition, it greatly reduces complexity of the respective code. This is especially important since the new Hybrid cubic spherical Bernstein-Bézier patch and Radial Basis Function interpolation make use of the same algorithm.

Originally, YAC internally uses spherical as well as Cartesian coordinates. This mixture requires a lot of conversions between both representations, which has an impact on performance and potentially on accuracy. Therefore, introduced with version 1.3.2, we internally only use Cartesian coordinates. The user input is still provided in spherical coordinates and is converted only once at the beginning.

4. Performance

As for the original YAC paper by Hanke et al. (2016), we repeated the performance measurements for the new interpolation methods. The measurements were again done on Mistral at the German Climate Computing Center (DKRZ). The parameters for our tests were as follows:

- Intel Xeon E5-2695V4 18-core nodes
- Intel compiler version 18.0.0
- OpenMPI version 2.0.2
- 36 MPI processes per node
- bidirectional exchange between an ICON R2B7 (983,040 cells) and a 810x810-cubed-sphere grid (3,936,600 cells)

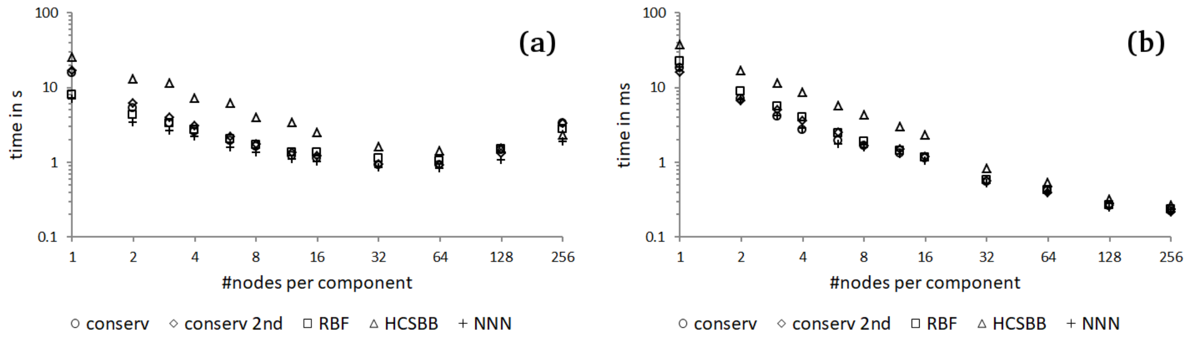

Figure 3: Time required for **(a)** weight generation and **(b)** ping-pong exchange

When compared to the result presented in Hanke et al. (2016) Fig. 3 (b) and Fig. 4 (b), the results in Fig. 3 are similar in respect to absolute values and scaling behaviour. Differences in absolute values can be attributed to differences in hardware, number of processes per node, used MPI library, and changes to the code itself. The new results show less variation between interpolation methods especially for higher node counts.

4.1. Coupling ICON with YAC

In order to provide some insight about how YAC behaves in the "real" world we briefly examine two ICON-based model configurations which make use of YAC for the coupling between the ocean and atmoshere component, here still with version 1.3.2. We recommend to keep in mind that the timing discussed below strongly depends on the quality of load balancing, the ratio and orientation of the grids and cells towards each other, and to some extent on the chosen interpolation stack(s) and stencils with its communications patterns.

In a first configuration both components run on horizontal grids with approx. 5 km horizontal (R2B9) resolution. As the ocean grid is derived from the atmosphere grids, corresponding cell centres and vertices are located at identical positions with the restriction that the ocean does not contain cells over land. Due to this special conditions we can employ a 1-nearest-neighbourhood search for all coupling fields and the data exchange can be viewed as a pure repartitioning of coupling fields. In a first set of model runs the atmosphere component is generally slower in the initialisation phase while it is considerably more compute intensive between coupling time steps. Therefore, the ocean component is waiting in both the initialisation phase (including the neighbourhood search) and during the coupling itself. The timing measured for the ocean processes mainly consists of MPI wait time while they are waiting for the atmosphere processes to participate in the coupling. In this configuration, the complete YAC initialisation takes 3.6 seconds for the atmosphere. The time measured for the ocean adds up to 45 seconds mainly, because the processes are waiting for the atmosphere to join the initialisation phase.

Figure 4 shows a sketch of the exchange of coupling fields. At the beginning of a run, the coupling fields are read from restart files. At each time step the coupling fields are delivered to the YAC library where they are accumulated (summed up). At coupling events, data are averaged, and interpolated onto the remote grid. The interpolated data are sent to the remote processes. Here it is received and copied back to user space. At the end of a job, accumulated data are once again averaged, interpolated and sent to the remote processes where they are written to restart files. As shown in the figure on each side data are first put to the library before attempting to receive.

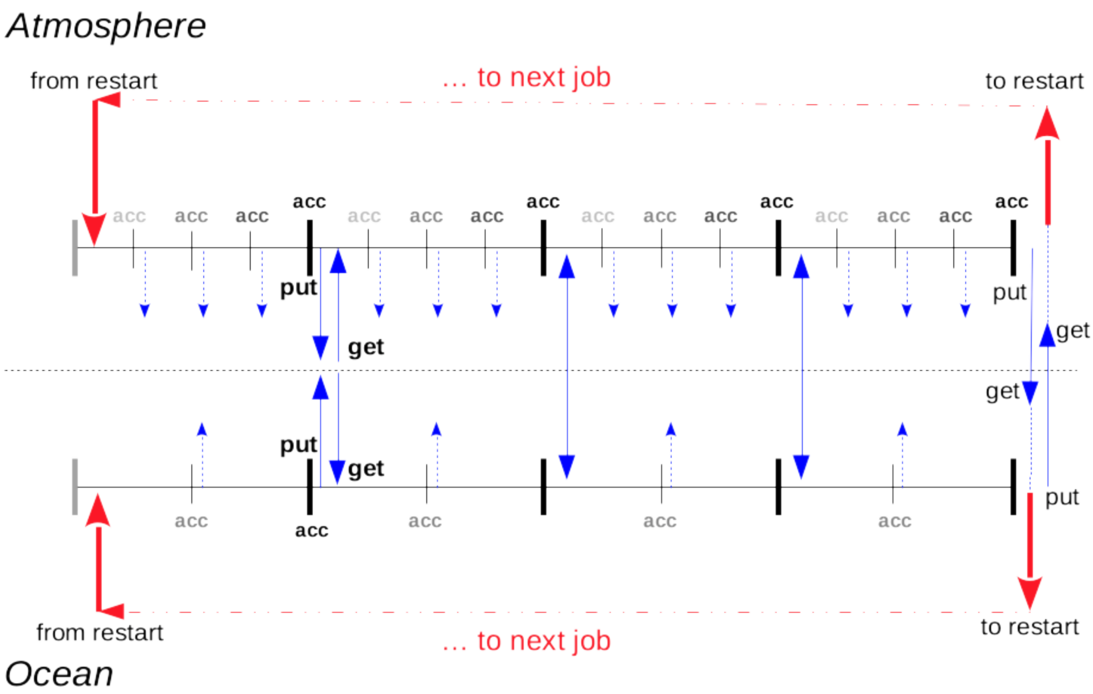

Figure 4: Schematic view of the coupling algorithm in ICON

For the ocean the exchange phase is dominated by the first calls of ocean yac_fget (8560 seconds), while this takes less than 1 second for the atmosphere. The huge differences are solely subject to the load imbalancing of the components and cannot be compensated by the coupling library. Due to memory problems we need to run the ocean component on much more processes than required for an optimal load balancing. The remaining calls to yac_fput and yac_fget accumulate from 1 to 4 seconds (4 seconds for the atmosphere yac_fput, 1 second for the ocean yac_fput, 1 second for the atmosphere yac_fget, and 3 seconds for the ocean yac_fget).

In ICON the coupling data are first send with subsequent calls to yac_fput, only then the subsequent yac_fget calls are invoked to receive the data. Therefore, the first call to yac_fget during a coupling timestep acts as a kind of synchronisation point. In this special configuration the time for the data exchange takes less than 3 seconds of the total run time of 3.5 hours in a 5 day simulation with a coupling interval of 15 minutes. As explained above, the ocean component is spending most of the time in the first call to yac_fget while the subsequent calls return at a similar speed as in the atmosphere.

The timing looks somewhat different in a more complex coupled configuration. The ICON-ESM that is setup targetting the Coupled Model Intercomparison Project (CMIP6) - type experiments uses grids at different horizontal resolution (R2B4 coupled to R2B6). The interpolation stack looks different for different coupling fields, and the two model components are in a better load balance. This makes it difficult to identify MPI wait time during the coupling in both the initialisation and data exchange phase. Here, the initialisation takes less than 2 seconds (0.6 s for the atmosphere and 1.6 seconds for the ocean). With a coupling of 32 times per day (corresponding to an interval of 45 minutes) over a 10 year run and a total elapsed time of approx. 3 hours, the wallclock time for the yac_fput calls adds up 100 seconds for the atmosphere and 30 to 45 seconds for the ocean. Most time-consuming are the first calls to yac_fget with an accumulated time of 28 minutes for the atmosphere and 18 minutes for the ocean. The subsequent calls to yac_fget accumulate to 15 to 20 seconds (atmosphere and ocean are not distinguishable) and are therefore almost negligible. Absolute numbers are hard to compare with this configuration as the number of coupling events per job is two orders of magnitude larger in the CMIP6 configuration 116.800 versus 480). On the other hand, individual numbers per call and process are difficult - if not impossible - to interpret.

5. Future Plans

The Radial Basis Function interpolation is the latest addition to YAC. It currently only supports a Gaussian kernel as the radial basis function. Depending on user feedback we are planing to also support other kernels and to further improve this interpolation method in respect to functionality and performance.

For some time, we thought about a new feature which would allow the user to change masks at run-time. With the current design of YAC this is especially difficult for source masks. Currently, there is no significant interest in this feature. Therefore, we postpone any developments in this direction. Except for the usual bug fixing, maintenance and user support there are currently no plans for further developments.

6. Further Information

We provide further information about the supported interpolation methods in the manual of the graphical user interface. ICON users find the pdf file in their ICON main directory under `external/yac/gui/manual` after having run `make pdf`.

Online documentaion about YAC is available on the YAC Doxygen web-page
`https://doc.redmine.dkrz.de/YAC/html`

ICON users may also have a look at
`https://code.mpimet.mpg.de/projects/icon-esm/wiki/ICON_Coupled_Model_Development`.

A tutorial about the YAC user interface is available from the authors on demand.

7. Acknowledgement

Thomas Jahns (DKRZ) and Sergey Kosukhin (MPI-M) improved the configure mechanism. Uwe Schulzweida provided several use cases for interpolations from the world outside ICON, which helped to improve the internal algorithms, in particular the clipping, and made them more robust.

Stephan Lorenz is the first end user who exploited the full functionality of YAC in the context of an Earth system model, the ICON-ESM. His feedback and collaboration with him helped us to improve the usability of YAC. In particular, he carefully checked the mapping results of the source to target method for correctness - one of the prerequsites to get a closed water cycle in the coupled model.

References

Alfeld, P., M. Neamtu, and L. L. Schumaker, 1996: Bernstein-Bézier polynomials on spheres and sphere-like surfaces. *Comput. Aided Geom. Des.*, **13**(4), 333–349.

Antonov, J., D. Seidov, T. Boyer, R. A. Locarnini, A. Mishonov, H. Garcia, O. Baranova, M. Zweng, and D. Johnson, 2010: *World Ocean Atlas 2009, Volume 2: Salinity. S. Levitus, Ed. NOAA Atlas NESDIS 69.* U.S. Government Printing Office, Washington, D.C. 184 pages.

Craig, A., S. Valcke, and L. Coquart, 2017: Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0. *Geosci. Model Dev.*, **10**(9), 3297–3308.

Gu, H., Z. Zong, and K. C. Hung, 2003: A modified superconvergent patch recovery method and its application to large deformation problems. *Finite Elem. Anal. Des.*, **40**(5-6), 665–687.

Hanke, M., R. Redler, T. Holfeld, and M. Yastremsky, 2016: YAC 1.2.0: new aspects for coupling software in Earth system modelling. *Geosci. Model Dev.*, **9**(1), 2755–2769.

Khoei, A. R. and S. R. Gharehbaghi, 2007: The superconvergent patch recovery technique and data transfer operators in 3d plasticity problems. *Finite Elem. Anal. Des.*, **43**(8), 630–648.

Kritsikis, E., M. Aechtner, Y. Meurdesoif, and T. Dubos, 2017: Conservative interpolation between general spherical meshes. *Geosci. Model Dev.*, **10**, 425–431.

Liu, X. and L. L. Schumaker, 1996: Hybrid Bézier patches on sphere-like surfaces. *J. Comput. Appl. Math.*, **73**(1.2), 157–172.

Reinheimer, J., 2018: Vector Field Interpolation using Radial Basis Functions. pp 68.

Zienkiewicz, O. C. and J. Z. Zhu, 1992: The Superconvergent Patch Recovery and a Posteriori Error Estimates. Part 1: The Recovery Technique. *Int. J. Numer. Methods Eng.*, **33**, 13311364.