Benchmarking of regridding libraries used in Earth System Modelling
SCRIP, YAC, ESMF and XIOS

Sophie Valcke, Andrea Piacentini, Gabriel Jonville

08 November 2021
CERFACS Technical Report TR-CMGC-21-145

The work reported in this document has been done in the framework of projects that have received funding from the European Union’s Horizon 2020 research and innovation programme:
IS-ENES3 under grant agreement Nº 824084
ESiWACE2 under grant agreement Nº 823988.

Table of contents

1. Introduction

The SCRIP library (Jones 1999) included in the OASIS3-MCT coupler (Craig et al. 2017) has been parallelized with a mix of OpenMP and MPI in the framework of the ESiWACE Centre of Excellence (Piacentini et al. 2018). This parallel version of SCRIP was made available in the OASIS3-MCT coupler with the OASIS3-MCT_4.0 release in June 2018 (Valcke et al. 2018).

A detailed analysis of the quality of the SCRIP library for different types of grids was realised in 2019 (Valcke & Piacentini, 2019; Jonville & Valcke, 2019). The results of the conservative remapping, the impact of the different normalisation options, i.e. DESTAREA and FRACAREA (see Sect. 4.3.4), and the impact of activating the Lambert projection above a certain latitude have been analysed. The grids tested are the ones used in the low-resolution coupled model at IPSL, IPSL-CM6_LR, and at CNRM-CERFACS, CNRM-CM6-1, i.e. NEMO for the ocean and LMDz, DYNAMICO, or the ARPEGE Gaussian Reduced for the atmosphere.

The general conclusion is that the SCRIP 1st Order[1] conservative remapping with the FRACAREA normalisation gives good results for all pairs of grids but, in some cases, only if the Lambert projection is not activated. With DESTAREA, regriddings involving grids other than the Gaussian Reduced grid show reasonable results; for some pairs of grids, the Lambert projection is mandatory, for others Lambert projection should not be activated, and for others the Lambert projection does not change the results. However, with DESTAREA, conservative regriddings involving the Gaussian Reduced grid always show some problems, whether or not the Lambert projection is activated.

This analysis led to the conclusion that other regridding possibilities should be offered in OASIS3-MCT. It was therefore decided, within the IS-ENES3 and ESiWACE2 contexts, to explore other regridding libraries currently available for Earth System Modelling and to make the most appropriate ones available to pre-calculate the regridding weights and addresses with the next version of the coupler, OASIS3-MCT_5.0, due as a deliverable in December 2021.

The current report details the benchmarking work done to evaluate the regridding functionality in ATLAS, MOAB-Tempest Remap, YAC, ESMF and XIOS, which are introduced in Sect. 2. In order to compare these libraries, several aspects had to be covered. In a preliminary analysis (see Sect. 3), we first enquired about the available regridding methods in each software and evaluated the general software development environment, e.g. the coding language, project history and development plans, provision of support to external projects, and manpower committed. Then we benchmarked the quality of regridding for YAC, ESMF and XIOS, and also for SCRIP as a basis for comparison, with the calculation of the metrics proposed by the CANGA project. The benchmark characteristics, the use of SCRIP, YAC, ESMF and XIOS, and the benchmark technical environment we developed are described in detail in Sects. 4, 5 and 6. In Sect. 7, we detail the benchmark results obtained for the four libraries; five regridding algorithms were evaluated for four different functions, some of them having strong gradients, for seven pairs of grids used in real ocean or atmosphere models. The last step was to evaluate the performance and scalability of the libraries, which we did for ESMF, XIOS and for SCRIP as a comparison (Sect. 8). Finally, conclusions and perspectives of this work are presented in Sect. 9.

[1] In this document, we will refer to "1st Order" and "2nd Order" as "1st O" and "2nd O"

2. Regridding libraries considered in the benchmark

The choice of the regridding libraries, or "regridders", to consider for our evaluation was based on our knowledge of regridding libraries used in Earth System Modelling. The libraries considered are ATLAS, MOAB-Tempest Remap, YAC, ESMF, and XIOS.

ATLAS (Deconinck et al, 2017; https://github.com/ecmwf/atlas) is an open source library currently developed at ECMWF providing grids, mesh generation, and parallel data structures targeting Numerical Weather Prediction or Climate Model developments. It is designed as an Object Oriented modular library, with the capability to take advantage of the most recent computer architectures. It is meant to provide, among many other features, a set of parallel interpolation methods for the conversion between different distributed representations of discrete physical fields and is oriented toward the use of an internally consistent set of predefined grids and meshes.

MOAB-Tempest Remap (Mahadevan et al, 2020) is used in the Energy Exascale Earth System Model (E3SM, https://e3sm.org), a state-of-the-art Earth system modelling project funded by the DOE in the US. Through Fortran-compatible interfaces, it enables online regridding strategies in order to simplify the coupled workflow process. This integrated infrastructure is based on the Mesh Oriented datABase (MOAB), which allows for a complete description of the numerical grids and solution data used in each submodel. Through a scalable advancing-front intersection algorithm, the supermesh of the source and target grids are computed, which is then used to assemble the high-order, conservative, and monotonicity-preserving regridding weights between discretization specifications.

YAC, Yet Another Coupler (Hanke et al, 2016; Hanke & Redler, 2019; https://dkrz-sw.gitlab-pages.dkrz.de/yac/), 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 (ICOsahedral Nonhydrostatic) model 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, regridding, 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 assumption about the particular grid structure or grid element types.

ESMF, the Earth System Modelling Framework, (Collins et al, 2005; https://earthsystemmodeling.org) is an open source software for coupling model components to form weather, climate, coastal, and other Earth science related applications. The framework allows for a specific part of a model developed by a particular modelling group to become a module of large scientific applications developed either by the same group or by others. The scientist only codes the scientific part of a model into modular components and adapts them to the standard calling interface and standard data structures of the shared infrastructure software. The ESMF software provides the underlying layers necessary for an efficient parallel execution of the scientific applications on different computer architectures, allowing for the coupling of the module to other components, including transfer and transformation of the coupling data.

XIOS (http://forge.ipsl.jussieu.fr/ioserver) standing for XML-IO-Server is an open source library, dedicated to I/O management in climate codes. XIOS manages output of diagnostics and other data produced by climate components into files. It aims at simplifying the I/O management by supporting a maximum of on-line temporal and spatial processing and regridding of the data. The output definition is defined in an XML file which allows the output configuration to be changed without recompiling. XIOS is fully parallel and targets climate models running on large number of cores. Recently, XIOS has also been used not only as an IO server but also as a coupler, i.e. managing communication of data not only between a component and a file but also between two components.

3. Preliminary analysis

We first realised a preliminary analysis in order to decide if the full benchmarking of each library was appropriate. We verified the available regridding methods and the maturity of the software. Given the SCRIP shortcomings, we were looking in particular for efficient and good-quality support of 1st and 2nd O conservative 2D remapping for masked grids on the sphere. We also evaluated the general software development environment, in particular the provision of support to external projects and the manpower committed.

3.1. ATLAS

The full preliminary analysis for ATLAS is reported in Piacentini (2020). ATLAS is an open source library written in C++ and provides a useful portable toolkit for the best usage of new heterogeneous architectures.

However, some basic capabilities were still missing in the version evaluated (0.21), in particular the handling of missing/masked values and conservative regriddings. Also, the overall design of the APIs was probably too oriented toward the coherent implementation of a full set of treatments based on ATLAS grid and field representations whereas a flexible tool like OASIS3-MCT needs to be able to account for geometries and (limited) representations coming from existing models that cannot be modified. Also, the state of the Fortran APIs was neither complete (w.r.t. C++ APIs) nor particularly flexible.

No roadmap had been announced for the implementation of conservative regridding or extension of the Fortran APIs and it may be that these are going to be progressively abandoned when IFS moves most of its internal procedures toward C++. Also, from a practical point of view, there was, at the time of our evaluation, no real documentation and the only way to get information on how to use the software was to scroll through the different test cases available with the sources. Finally, even though the longevity of ATLAS is probably ensured given ECMWF interest and investment in the software, ECMWF has not committed to any firm community user support. In summary, ATLAS is certainly an appealing library with long-term perspectives but its usage for regridding in OASIS3-MCT on the short term cannot be recommended at this point.

3.2. MOAB-Tempest Remap

The preliminary analysis for MOAB-Tempest Remap, which is also written in C++, leads to a similar conclusion as for ATLAS. Even if it looks like a very promising library benefiting from some important investment in the framework of E3SM, our conclusion is that it was not mature enough at the time of our evaluation to be considered as an efficient complement to the SCRIP library.

MOAB-Tempest Remap supports 1st and 2nd O conservative remapping implemented through an interesting advancing-front intersection algorithm detailed in Mahadevan et al (2020) but no non-conservative algorithm such as nearest-neighbour or bilinear. The most important drawback at the time of our evaluation is that it did not properly support masked cells. To perform calculation for a grid having a mask, the masked cells had to be simply removed from the mesh, which caused problems for 2nd O conservative regridding, particularly for the NEMO rotated and stretched grids.

Therefore, after running few preliminary test cases, we decided not to go on further with the benchmarking of MOAB-Tempest Remap regridding. We will of course follow closely the developments of this library as its full integration with MOAB and its original advancing-front intersection algorithm certainly favours very high computational efficiency.

3.3. YAC

YAC is coded in C, while Fortran interfaces for all API routines are also provided. Multiple regridding methods are available in YAC, e.g. fixed-value, linear, distance-weighted, distance gaussian weighted, 1st and 2nd order conservative, hybrid cubic Bernstein-Bézier patch.

Different best practices in software development were also adopted such the use of an extensive unit test suite, use of Doxygen for automatic production of code documentation from annotated code sources, and source version control with GIT. A tutorial is also available and the repository itself contains a couple of examples that demonstrate usage of YAC.

Even though YAC is the coupler of choice for ICON and is integrated into its configuration and build process, YAC is still a stand-alone library. Currently, one person is devoted to its development and DKRZ pushes YAC to be part of the German national ESM strategy (NatESM), so that its mid- and long-term perspectives can be guaranteed. Since YAC is used by ICON, DKRZ is committed to its development. However, support to non-DKRZ-customers is not guaranteed without third party funding, although in practice, its main developer is very dynamic and responsive.

3.4. ESMF

ESMF development started in the early 2000s as an attempt to create a common framework for Earth System Modelling in the USA. In 2008, a joint project gathering the United States Navy, the National Weather Service, and the United States Air Force started as the National Unified Operational Prediction Capability (NUOPC; see http://earthsystemmodeling.org/nuopc/) in order to increase the level of interoperability between ESMF components. Today ESMF is developed and governed by a set of partners in the USA that includes the National Aeronautics and Space Administration (NASA), the National Oceanic and Atmospheric Administration (NOAA), the Department of Defense (DoD) and the National Science Foundation (NSF).

ESMF offers a full interface to Fortran 90 and partial interface to C/C++ and Python. ESMF offers a “superstructure” of functions and component wrappers with standard interfaces which forms ESMF coupling functionality. ESMF also provides a separate “infrastructure” of utilities that can be used internally by the components to handle technical aspects of the coding, in particular a regridding utility. The source and destination fields can be discretized on global or regional domains using logically rectangular grids or unstructured meshes. ESMF supports regridding on combinations of 2D or 3D spherical or cartesian coordinates with different regridding methods: nearest-neighbour, bilinear, higher-order based on patch recovery, 1st- and 2nd-order conservative.

ESMF is used in major coupled systems at NASA, Navy, NCAR, and NOAA and in other modelling applications from universities and major U.S. research centres. It offers free and active user support and complete reference and user guides are available on-line. All these characteristics give confidence in its future and make it a very strong candidate library for the current benchmarking exercise.

3.5. XIOS

XIOS is an open source library written in C++ with a Fortran interface developed to manage the I/O of the IPSL (Institut Pierre-Simon Laplace) Earth System Models. It offers an impressive ensemble of on-line operations on model data (file rebuild, time series, seasonal means, regridding, vertical interpolation, compression, etc.), based on external XML metadata definition, in order to minimize the post-processing of the data. Its regridding utility offers 1st and 2nd order conservative remapping (but no non-conservative algorithms) on any type of grids used in Earth System Modelling and can produce regridding weight files to be reused in a coupler.

XIOS is developed by a small team of 5 people and is used in all IPSL ESM components. It is the I/O server implemented in NEMO and as such is used by the large community of NEMO users. It is also used at Météo-France and at CERFACS group for the I/O of all our ESM components and has been extensively used in our CMIP6 simulations. Lately, it has been used by other groups in Europe, for example at the UK Met Office and in the European community ESM EC-Earth. Different mailing lists exist and a User Guide and Reference Guide are available online for user support.

As we (CERFACS) already use XIOS for I/O in our ESM, and given the geographical proximity and the common administrative umbrella by CNRS, we benefit from an excellent collaboration with XIOS developers. For all those reasons, we were very interested in including XIOS in our benchmarking exercise and evaluating how XIOS regridding functionality could serve OASIS3-MCT.

3.6. Conclusion of the preliminary analysis

This preliminary analysis leads us not to go on further with the benchmarking of ATLAS and MOAB-Tempest Remap. ESMF and XIOS, on the contrary, are very interesting candidates. The same is true for YAC with a somewhat lower priority given the lack of official support for non-DKRZ-customers.

4. Benchmark characteristics

4.1. Grids

The six grids considered in the benchmark are the followings, given with their acronym and number of grid points (in the rest of the document, we will use the acronyms for simplicity):

- torc : NEMO ORCA2 rotated-stretched logically-rectangular (ocean, 182x149)
- nogt : NEMO ORCA1 rotated-stretched logically-rectangular (ocean, 362x294)
- bggd: LMDz regular latitude-longitude (atmosphere, 144x143)
- sse7 : ARPEGE Gaussian reduced T127 (atmosphere, 24572)[2]
- icos : Dynamico low-resolution icosahedral grid (atmosphere, 15222)
- icoh : Dynamico high-resolution icosahedral grid (atmosphere, 2016012)

These grids are illustrated on Fig.1.

[2] The cells of the Gaussian reduced grid were originally defined with 4 vertices each. Here we use a recent transformation of the definition of the cells with a maximum of 7 vertices each so that a cell on one latitude row include also the original vertices of the upper and lower latitude rows in the case where these vertices are different from those of the cell. This ensure that the whole globe is covered without any holes independently of the projection used. For more details, see (Jonville & Valcke, 2019).

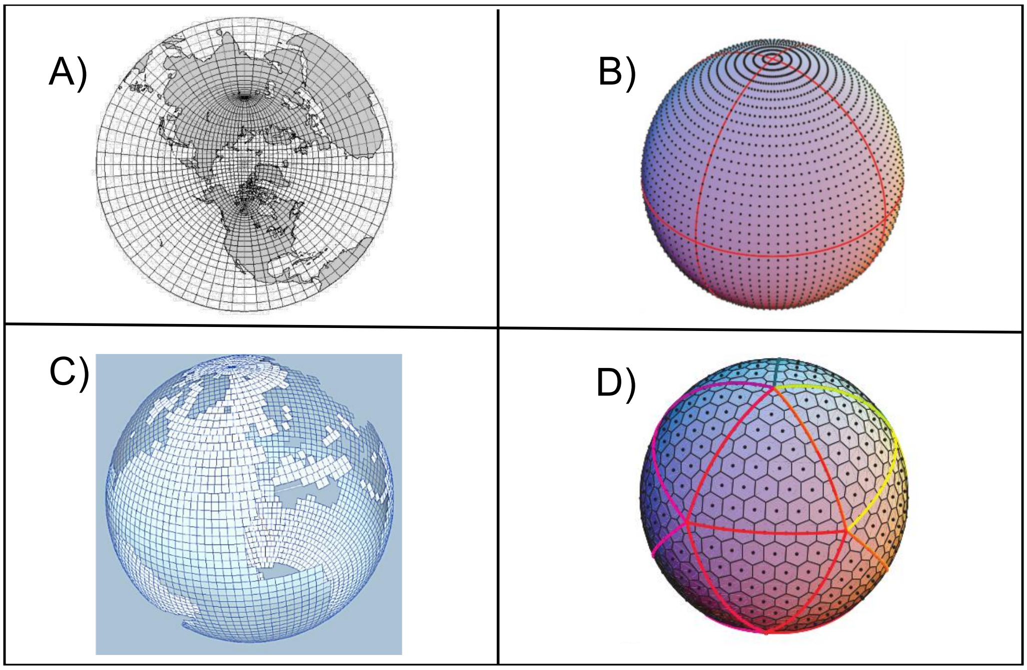


Figure 1 – Types of grids included on the benchmarks: A) rotated-stretched logically-rectangular (NEMO) ; B) regular latitude-longitude (LMDz) ; C) Gaussian-reduced (ARPÈGE)[3] ; D) icosahedral (Dynamico)

The regriddings are evaluated in both directions for the following pairs of grids matching an ocean (torc, nogt) and an atmospheric grid (bggd, icos, sse7): torc-bggd, torc-icos, torc-sse7, nogt-bggd, nogt-icos, nogt-sse7. For the 2nd order conservative regridding, we also analyse the regridding for icos-icoh and nogt-icoh in order to test the impact of having a large resolution difference and the behaviour of the 2nd order in that case (see Sect. 7.6).

The NEMO ORCA12 (i.e. with a resolution of ~1/12 degree) rotated-stretched logically-rectangular grid with 3147x4322 grid points, t12e, was also used for the performance tests, see Sect. 8.

4.2. Analytical functions

The 4 analytical functions used to define the coupling fields to be regridded are illustrated on Fig. 2[4].
a) sinusoid (also called “classic”): a slowly varying standard sinusoid over the globe
b) harmonic: a more rapidly varying function with 16 maximums and 16 minimums in northern and southern bands

[3] Here the Gaussian-reduced grid is plotted with 4 vertices per cell.

[4] Their exact definition is available at https://inle.cerfacs.fr/attachments/10233/function_ana.f90 and https://nitrox.cerfacs.fr/globc/OASIS3-MCT/oasis3-mct_other/blob/master/compare_interpolations/src/function_ana.f90

c) vortex: a slowly varying function with two added vortices, one in the Atlantic and one over Indonesia
d) gulfstream: the slowly varying standard sinusoid with a mimicked Gulf Stream

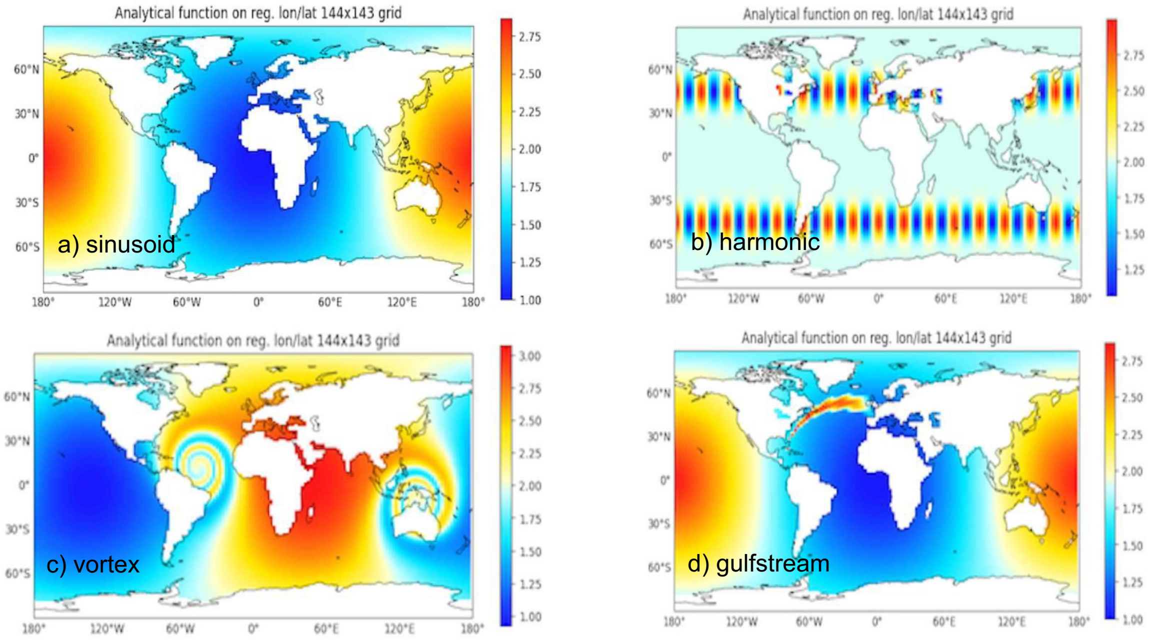


Figure 2 – The 4 functions defining the analytical field to be regridded: a) sinusoid (also called "classic"), b) harmonic, c) vortex, d) gulfstream

4.3. Algorithms

The following algorithms are evaluated for the different regridding libraries, when available. The particularities of the algorithm for each library are detailed, if need be. We note here in particular that XIOS does not implement any non-conservative algorithm (nearest-neighbour, BILINEAR, 2nd O non-conservative).

More details can be found in:

- SCRIP: SCRIP User Guide at https://github.com/SCRIP-Project/SCRIP/wiki, (Jones, 1999)
- YAC: Hanke & Redler (2019)
- ESMF: ESMF Reference Guide https://earthsystemmodeling.org/docs/release/ESMF_8_1_1/ESMF_refdoc.pdf
- XIOS: Kritsikis et al (2017)

4.3.1. Nearest-neighbour

The values of the nearest neighbour on the source grid gives the value for each target grid point.

4.3.2. Bilinear or equivalent (BILINEAR)

The regridding is based on a bilinear approximation, which uses the value of the coupling field at the four enclosing source grid points. The standard algorithm can be found in many textbooks.

- SCRIP uses a more general scheme based on a local bilinear approximation to interpolate to a point in a quadrilateral grid.
- YAC does not implement a bilinear algorithm per se. We therefore use an inverse distance weighting of the vertices of the source polygon enclosing the target point + 2 nearest neighbours for target points falling outside any source polygon, which should be more or less equivalent.
- ESMF uses a standard bilinear algorithm.

4.3.3. 2nd O non-conservative

Different algorithms are used in the different regridders to evaluate 2nd O non-conservative regriddings:

- For SCRIP, the bicubic regridding follows the bilinear remapping except that four weights for each corner point are required, one for the value at the corner and one for the gradients in each local direction and for the cross gradient.
- YAC uses a method for interpolating scattered data on sphere-like grids based on a local triangular patch which is constructed from a blend of certain spherical Bernstein-Bézier polynomials introduced recently by Alfeld, Neamtu & Schumaker (1996).
- For ESMF, we used the patch algorithm that is a technique commonly used in finite element modelling. Patch interpolation works by constructing multiple polynomial patches for the cells around the vertices of a source cell (e.g. for a square source cell we would compute four patches). For 2D grids, these polynomials are currently 2nd degree 2D polynomials. The interpolated value at the destination point is the weighted average of all the patches for the source cell (e.g. the four patches for a square cell). We will see that this patch averaging prevents too strong over and undershoots.

4.3.4. 1st O conservative with FRACAREA (CONSERV FRACAREA) and DESTAREA (CONSERV DESTAREA) normalisations

In a 1st O conservative remapping, the value for each target cell is computed as a weighted sum of source cell values, with the contribution of a source cell being proportional to the fraction of the target cell intersected by the source cell. This method should be applied when it is important to conserve the area-integrated value of the coupling field, for example to conserve the energy associated with heat flux or water associated precipitation fields.

In case of non-matching sea-land masks between the atmosphere and the ocean grids, different normalisation options exist. DESTAREA uses the whole target cell area for the normalisation, whereas FRACEARA uses the intersected area of the target cell. DESTAREA ensures local conservation but may produce non-physical values while FRACAREA does not ensure local conservation but produces values that are physically consistent. We note also that the FRACAREA normalisation may give some good results for wrong reasons in the sense that the normalisation operation with the intersected target cell area, as calculated by the regridder itself, may lead to hide an error present in the weights before the normalisation; DESTAREA does not involve this error cancellation and therefore often reveals specific algorithmic problems.

4.3.5. 2nd O conservative with FRACAREA normalisation (CONS2ND FRACAREA)

The basis of a 2nd O conservative remapping is the same as the 1st O conservative remapping but additional terms proportional to the gradients of the source field are applied. While remaining conservative, this remapping ensures that field details are reconstructed and that different target cells entirely located under the same source cell receive different values. This difference between the 1st O and 2nd O methods is particularly apparent when going from a coarse source grid to a finer destination grid (see Sect. 7.6). Another difference is that the 2nd O method does not guarantee that after regridding the range of values in the destination field is within the range of values in the source field. For example, if the minimum value in the source field is 0.0, it is possible that after regridding with the 2nd O method, the destination field contains negative values.

SCRIP applies gradients calculated in the longitudinal and latitudinal directions while YAC, ESMF, and XIOS implements the same 2nd O conservative algorithm based on Kritsikis et al (2017).

4.4. Benchmark metrics

The benchmark implements the calculation of the metrics proposed by the CANGA project (https://github.com/CANGA/Remapping-Intercomparison). Following CANGA, aspects to consider when evaluating a regridding library are:

- the sensitivity, i.e. the algorithmic invariance of the scheme to the underlying mesh topology
- the global conservation of integral quantities
- the consistency, i.e. the preservation of discretization order and accuracy
- the monotonicity, i.e. the preservation of global solution bounds
- the scalability and performances of the library
- the dissipation, which has to be minimal

CANGA proposes different metrics that we implemented. With:

- Ψ^s: the analytical function on the source grid
- Ψ^t: the analytical function on the target grid
- $R\Psi^s$: the source analytical function regridded on the target grid
- I_s: the integral on the source grid
- I_t: the integral on the target grid

Those metrics are defined as:

- mean misfit: mean $(|R\Psi^s - \Psi^t| / |\Psi^t|)$
- maximum misfit[5]: max $(|R\Psi^s - \Psi^t| / |\Psi^t|)$

[5] For the calculation of all the metrics for CONSERV DESTAREA, except for global conservation metrics, we explicitly removed coastal cells; in fact, depending on the mismatch of the land-sea mask, DESTAREA may give target values that are not physically coherent but this is purely due to the normalisation and is not an imprecision or problem of the conservative remapping method per se.

- RMS misfit : RMS ($|R\Psi^s - \Psi^t|/|\Psi^t|$)
- L1: $I_t(|R\Psi^s - \Psi^t|) / I_t(|\Psi^t|)$, which is supposed to characterize error in large scale features
- L2: square root [$I_t(|R\Psi^s - \Psi^t|^2) / I_t(|\Psi^t|^2$], which is supposed to characterize error in small scale features
- Lmin: (min Ψ^t – min $R\Psi^s$) / max ($|\Psi^t|$): a positive Lmin detects an undershoot of the function minimum (i.e.it is reinforced) while a negative Lmin detects some smoothing of the function minimum
- Lmax : (max $R\Psi^s$ – max Ψ^t) / max ($|\Psi^t|$) a positive Lmax detects an overshoot of the function maximum (i.e.it is reinforced) while a negative Lmax detects some smoothing of the function maximum
- Source global conservation: $|I_t(R\Psi^s) - I_s(\Psi^s)| / I_s(\Psi^s)$
- Target global conservation: $|I_t(R\Psi^s) - I_t(\Psi^t)| / I_t(\Psi^t)$

We calculated these metrics for all regridders for all pairs of grids for the four functions for all algorithms except when the regridder did not support the algorithm. In our analysis however, we did not consider L1 and L2 as we did not really understand their meaning.

Regarding aspects to consider when evaluating a regridding library, as proposed by CANGA, we can say that our benchmark provides a strong basis to evaluate for the different regridders:

- the sensitivity as we cover 6 pairs of grids in both directions and in addition the icos-icoh and nogt-icoh pairs for the vortex function for 2nd O conservative FRACAREA remappings;
- the global conservation as we provide the source and target global conservation metrics;
- the scalability and performances of the library with the performance analysis presented in Sect. 8.

Also, as we provide and analyse the Lmin and Lmax metrics, our benchmark also allows a first analysis of the regridder monotonicity and dissipation (or smoothing). However, we did not address the consistency, i.e. the preservation of discretization order and accuracy, of the regridding libraries.

5. Using the ESMF, XIOS or YAC to calculate the regridding weights

The steps to realize in order to calculate the benchmark metrics for each regridder is, of course to download the sources, compile them, and develop an environment to generate regridding weights activating the different regridding algorithms for the different pairs of grids for the different functions. We went through these steps for ESMF, XIOS or YAC. With all regridders, specific issues arose while doing so and generated lots of interactions with the regridder developers to improve the first results obtained. These interactions were really interesting but we will not detail them here for sake of conciseness. For completeness, we also describe the environment used to generate the weights with the SCRIP library, as the benchmark metrics were also calculated for the SCRIP for comparison.

5.1. SCRIP

5.1.1. SCRIP sources and regridding weight generation

The OASIS3-MCT sources used for the regridding benchmark correspond to the trunk of the OASIS git developer repository (https://nitrox.cerfacs.fr/globc/OASIS3-MCT/oasis3-mct) dated 2021/05/05. The benchmark tests were run on LENOVO cluster nemo at CERFACS (288 bi-socket nodes with 12 Intel cores E5-2680-v3 2.5 Ghz with 64 GB of memory).

An environment to calculate regridding weights with the SCRIP library in OASIS3-MCT is available on CERFACS git developer repository at https://nitrox.cerfacs.fr/globc/OASIS3-MCT/oasis3-mct_other/tree/master/test_hybrid . Its version dated 2021/07/22 tagged "d5b8bc99be24270aaaf2f99eb1a7ad80a4365b19" is also available in the tar file test_hybrid.tar on Zenodo at https://doi.org/10.5281/zenodo.5342548. This environment is an evolved version of the test_interpolation environment available with OASIS3-MCT sources. Instructions on how to install OASIS3-MCT, and therefore the SCRIP library, are provided in section 6 of the User Guide (https://oasis.cerfacs.fr/en/documentation/). Calculating the regridding weights for the benchmark was therefore quite straight forward for the SCRIP library.

The environment test_hybrid implements one coupling exchange between two toy models and includes the calculation of the regridding weights. The programs model1.F90 and model2.F90 have to be compiled with the Makefile therein and the coupling can be launched with the script `run_hybrid.sh` with:

```
> ./ run_hybrid.sh src_dst nnodes_nprocs_nthreads method
```

where `src` and `dst` are the source and target grids, `nnodes` is the total number of nodes for the run, `nprocs` is the number of MPI tasks per node, `nthreads` is the number of OpenMP threads per MPI task, and `method` is the chosen regridding algorithm. After the run, the weights can be found in the execution directory in the file named following the OASIS3-MCT convention `rmp_src_to_dst_method.nc.`

5.1.2. SCRIP regridding particularities and options

No specific options, besides the normalisation options DESTAREA or FRACAREA, were activated for the different regriddings tested.

By default, the non-conservative algorithms NEAREST-NEIGHBOUR, BILINEAR and BICUBIC use the nearest non-masked source neighbour value for non-masked target points that do not receive a value with the original algorithm.

For the conservative remappings, the SCRIP library assumes by default that the edges of the meshes are linear in longitude and in latitude. It is however possible, for the edge intersection calculation, to switch to a Lambert equivalent azimuthal projection above a certain latitude threshold if specified. We performed the benchmark tests either without any projection, or with a projection above 1.45 radians in latitude North. In the later case, the results are denoted “SCRIP-L” and in the former case, they are denoted “SCRIP”.

Another limitation of the SCRIP conservative remapping algorithm is that it supposes, for line integral used in the area calculation, that `sin(latitude)` is linear in longitude on the cell borders which again is in general not valid close to the pole.

As mentioned in the introduction, a detailed analysis of the quality of the SCRIP library was realized in 2019 and results are detailed in (Valcke & Piacentini, 2019) and (Jonville & Valcke, 2019). The general conclusion is that the 1st O conservative remapping gives reasonable results only if the Lambert projection is activated for some pairs of grids and only if it is not activated for some other pairs of grids. Also, conservative regriddings with the DESTAREA normalisation involving the Gaussian Reduced grid always show some problems, whether or not the Lambert projection is activated. This is why we decided to consider other regridding libraries. In the current benchmarking exercise, we performed metric calculations for SCRIP and SCRIP-L, as a basis for comparison. However, if specific problems are revealed by the benchmark, we do not further investigate them as the current objective is to evaluate other regridding libraries.

5.2. YAC

5.2.1. YAC sources and regridding weight generation

YAC sources used for the regridding benchmark corresponds to a pre-release state of YAC v2.0.0 that was provided by the developers. All developments used in this version are now included in the official release YAC v2.3.0 available at https://www.dkrz.de/en/services/software-development.

The environment to calculate regridding weights with YAC is available on CERFACS git developer repository at https://nitrox.cerfacs.fr/globc/OASIS3-MCT/oasis3-mct_other/tree/master/generate_weights/YAC. Details on YAC installation and how to use it to calculate the weights are provided in Appendix 1 section 1. The programs used in this environment toy_scrip.c and generate_OASIS_mask_file.c are now part of YAC sources and will follow YAC evolution.

All regridding weight calculations were done on a PC Dell Precision M7720 with 6 cores Intel Xeon E-2186M, 64 Gb RAM.

5.2.2. YAC regridding particularities and options

For the conservative remapping, the edges of the grid cells can be either defined with longitude and latitude circles or with great circles depending on the interface used. We used the interface defining the edges of the grid cells with great circles. We have to note here that this is not totally appropriate for the cell edges following a latitude circle in the regular latitude-longitude grid bggd and in the Gaussian-Reduced grid, sse7. Also for simplicity, we described all grids as unstructured but this should have no impact on the benchmark metric results.

Also, for the 2nd order conservative in cases were the gradient computation fails (for example due to a lack of neighbours, which can occur at land--sea mask borders), the algorithm automatically assumes a zero gradient, which basically is a fall-back to 1st O conservative remapping (as in XIOS and ESMF).

5.3. ESMF

5.3.1. ESMF sources and regridding weight generation

We downloaded and installed ESMF sources following on-line instructions, see ESMF User Guide at https://earthsystemmodeling.org/docs/release/ESMF_8_1_1/ESMF_usrdoc/ . The sources used for the results presented in Sect. 7 correspond to the branch ESMF_8_2_0_beta_snapshot_08, which can be obtained with the git command

```
git clone https://github.com/esmf-org/esmf.git --branch
ESMF_8_2_0_beta_snapshot_08 --depth 1
```

Then the following environment variables have to be defined according to the platform used (here for kraken, the platform used for the performance tests described in section 8):

```
export PYTHON=`which python`
export ESMF_DIR=/scratch/globc/valcke/SOFTS/esmf
export ESMF_COMPILER=intel
export ESMF_INSTALL_PREFIX=/scratch/globc/valcke/opt/INTEL/esmf
export ESMF_INSTALL_HEADERDIR=include
export ESMF_INSTALL_MODDIR=mod/modg/Linux.intel.64.intelmpi
export ESMF_INSTALL_LIBDIR=lib/libg/Linux.intel.64.intelmpi
export ESMF_INSTALL_BINDIR=bin/bing/Linux.intel.64.intelmpi
export ESMF_INSTALL_DOCDIR=doc
export PATH=$ESMF_INSTALL_PREFIX/$ESMF_INSTALL_BINDIR:$PATH
export ESMF_NETCDF=split
export ESMF_NETCDF_LIBPATH="/softs/local_intel/netcdf/4.4.4_phdf5_1.10.4/lib -
L/softs/local_intel/netcdf/4.4.4_phdf5_1.10.4/lib"
export ESMF_NETCDF_INCLUDE="/softs/local_intel/netcdf/4.4.4_phdf5_1.10.4/include -
I/softs/local_intel/netcdf/4.4.4_phdf5_1.10.4/include"
export ESMF_COMM=intelmpi
export PATH=$ESMF_INSTALL_PREFIX/$ESMF_INSTALL_BINDIR:$PATH
```

Finally, ESMF can be compiled with “`gmake lib`” and “`gmake install`” in ESMF directory.

Once installed the major steps in using ESMF are to describe the grids of the coupling fields in the appropriate format and to identify options to activate for the different regriddings.

An environment developed to generate regridding weights with ESMF is available on CERFACS git developer repository at https://nitrox.cerfacs.fr/globc/OASIS3-MCT/oasis3-mct_other/tree/master/generate_weights/ESMF. Its version dated 2021/07/22 tagged “d5b8bc99be24270aaaf2f99eb1a7ad80a4365b19” is also available in the tar file generate_weights_ESMF.tar on Zenodo at https://doi.org/10.5281/zenodo.5343048 .

The main scripts in this environment are `ESMFGenRmp.sh`, `ESMFWeightsToOasis.sh` and `OasisGridsToESMF.py`. The scripts `batterie_test.sh` and `nres_kraken.sh` were developed for the performance tests described in Sect. 8 (to chain different tests and to extract the time needed to calculate the weights). To launch the calculation of regridding weights, the online command is:

```
> /ESMFGenRmp.sh nnodes_nprocs_nthreads src dst method
```

where nnodes is the total number of nodes for the run, nprocs is the number of MPI tasks per node, nthreads is number of OpenMP threads per MPI task, src and dst are the source and target grids and method is the chosen regridding algorithm.

The script then launches a batch job that calls OasisGridsToESMF.py, which first transforms the grids files from their original OASIS3-MCT format to the ESMF SCRIP format and then, if specified for the grid, from the ESMF SCRIP format to the ESMF unstructured format (see Sect. 5.3.3). Then it calls the ESMF function ESMF_RegridWeightGen , that performs the calculation of the regridding, in parallel with an mpirun command and different arguments. The regridding weights and associated source and target addresses are then calculated and available in the working directory in the file ESMFweights.nc. The script ESMFWeightsToOasis.sh can then be used to transform the weights file to the OASIS3-MCT weight file format.

5.3.2. ESMF regridding particularities and options

The arguments used for the weight generation are the following:

- NEAREST-NEIGHBOUR (neareststod in ESMF language): --src_loc center --dst_loc center --ignore_degenerate
- BILINEAR: --extrap_method neareststod --src_loc center --dst_loc center --ignore_degenerate
- 2nd O non-conservative (PATCH): --extrap_method neareststod --src_loc center --dst_loc center --ignore_degenerate
- 1st O conservative CONSERV DESTAREA(conserve_destarea): --ignore_unmapped --ignore_degenerate
- 1st O conservative CONSERV FRACAREA(conserve_destarea): --ignore_unmapped --norm_type fracarea --ignore_degenerate
- 2nd O conservative CONS2ND FRACAREA(conserve2nd_destarea): --ignore_unmapped --norm_type fracarea --ignore_degenerate

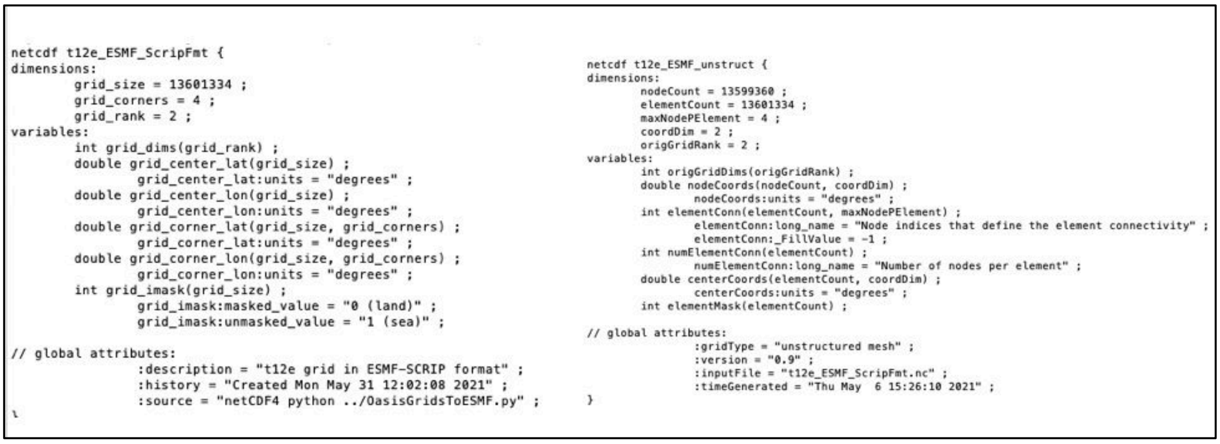
These options have the following effect:

- --src_loc center --dst_loc center : allow non-conservative regriddings on the cell centre locations of an unstructured grid defined with the ESMF unstructured file format;
- --extrap_method neareststod : each target point that does not receive a value with the original algorithm uses the closest unmasked source point to define its value; we activated this option for non-conservative algorithms so as to reproduce the default behaviour of the SCRIP library;
- --ignore_unmapped : do not do anything special for target point that does not receive a value with the original algorithm; we activated this option for conservative algorithms so as to reproduce the default behaviour of the SCRIP library;
- --ignore_degenerate to ignore degenerate cells in either the source or the destination grid; this can be useful for the NEMO ORCA grids torc and nogt which may have masked cells (i.e. not used in the regridding) collapsing into a point or line.

For conservative regridding ESMF supports great circle edges, and for non-conservative it supports both cartesian and great circle edges. The “cartesian” option specifies that the edge follows a straight path through the 3D Cartesian space in which the sphere is embedded. Under this option cells are approximated by planes in 3D space. The “greatcircle” option specifies that the edge follows a great circle path along the sphere surface. We used the default options, i.e. “cartesian” for non-conservative methods and “greatcircle” for conservative methods.

5.3.3. ESMF grid formats: SCRIP and unstructured

With ESMF, grids can be described with the so-called “SCRIP” format or with an unstructured format. As an illustration, the headers of the NetCDF files (“ncdump -h” command) describing the NEMO ORCA12 grid, t12e, in the two formats are provided at Fig.3. The SCRIP format describes the grid with the latitude and the longitude of the centre (`grid_center_lat, grid_center_lon`) and corners (`grid_corner_lat, grid_corner_lon`) of each cell (1360334 cells for t12e). The unstructured format describes the grid as an ensemble of elements (1360334 elements for t12e) and provides the element connectivity associating for each element a certain number of nodes (`elementConn`) in the list of nodes for which the latitude and longitude are provided (`nodeCoords`).



```
netcdf t12e_ESMF_ScripFmt {
dimensions:
        grid_size = 13601334 ;
        grid_corners = 4 ;
        grid_rank = 2 ;
variables:
        int grid_dims(grid_rank) ;
        double grid_center_lat(grid_size) ;
                grid_center_lat:units = "degrees" ;
        double grid_center_lon(grid_size) ;
                grid_center_lon:units = "degrees" ;
        double grid_corner_lat(grid_size, grid_corners) ;
                grid_corner_lat:units = "degrees" ;
        double grid_corner_lon(grid_size, grid_corners) ;
                grid_corner_lon:units = "degrees" ;
        int grid_imask(grid_size) ;
                grid_imask:masked_value = "0 (land)" ;
                grid_imask:unmasked_value = "1 (sea)" ;

// global attributes:
                :description = "t12e grid in ESMF-SCRIP format" ;
                :history = "Created Mon May 31 12:02:08 2021" ;
                :source = "netCDF4 python ../OasisGridsToESMF.py" ;
}
```

```
netcdf t12e_ESMF_unstruct {
dimensions:
        nodeCount = 13599360 ;
        elementCount = 13601334 ;
        maxNodePElement = 4 ;
        coordDim = 2 ;
        origGridRank = 2 ;
variables:
        int origGridDims(origGridRank) ;
        double nodeCoords(nodeCount, coordDim) ;
                nodeCoords:units = "degrees" ;
        int elementConn(elementCount, maxNodePElement) ;
                elementConn:long_name = "Node indices that define the element connectivity" ;
                elementConn:_FillValue = -1 ;
        int numElementConn(elementCount) ;
                numElementConn:long_name = "Number of nodes per element" ;
        double centerCoords(elementCount, coordDim) ;
                centerCoords:units = "degrees" ;
        int elementMask(elementCount) ;

// global attributes:
                :gridType = "unstructured mesh" ;
                :version = "0.9" ;
                :inputFile = "t12e_ESMF_ScripFmt.nc" ;
                :timeGenerated = "Thu May  6 15:26:10 2021" ;
}
```

Figure 3 – Headers of NetCDF files describing the NEMO ORCA12 grid (t12e) with the SCRIP (left) or unstructured (right) formats supported in ESMF

5.4. XIOS

5.4.1. XIOS sources and regridding weight generation

Details on XIOS installation are available at https://forge.ipsl.jussieu.fr/ioserver/wiki/documentation.

The sources used for the results presented in Sect. 7 correspond to SVN revision 2134 dated 2021-04-29 that can be extracted with the SVN command:

```
svn co -r 2134 http://forge.ipsl.jussieu.fr/ioserver/svn/XIOS/trunk XIOS
```

The compiling environment has to be defined in the XIOS/arch directory in files arch-xxxx.env, arch-xxxx.fcm, arch-xxxx.path where xxxx refers to the specific platform (see examples in there). XIOS compilation can then be launched with “`./make_xios --arch xxxx`” .

The environment developed to generate regridding weights with XIOS is available on CERFACS git developer repository at https://nitrox.cerfacs.fr/globc/OASIS3-MCT/oasis3-mct_other/tree/master/generate_weights/XIOS. Its version dated 2021/07/22 and tagged “d5b8bc99be24270aaaf2f99eb1a7ad80a4365b19” is also available in the tar file `generate_weights_XIOS.tar` on Zenodo at https://doi.org/10.5281/zenodo.5342491.

As for YAC, all regridding weight calculations were done on a PC Dell Precision M7720 with 6 cores Intel Xeon E-2186M, 64 Gb RAM.

The program used to generate the weights is `oasis_testcase/oasis_testcase.f90.` This program generates a field corresponding to the analytical function chosen and outputs it in a file `toy_output_tgt.nc` after having interpolated it to a target grid given by `tgt_domain`, as defined in the configuration files `oasis_testcase/iodef.xml` and `oasis_testcase/context_toy.xml.` The corresponding weights are saved in a file `remap_weights.nc` .

For compilation, `oasis_testcase.f90` has to be copied to the XIOS sub-directory XIOS/src/test and a line "bld::target oasis_testcase.exe" has to be added in XIOS/bld.cfg. At XIOS recompilation, the executable oasis_testcase.exe will then be automatically generated in directory XIOS/bin .

The program `oasis_testcase.f90` can be executed in parallel using an mpirun command (see `oasis_testcase/ job.sh_t12e_icoh` for an example).

The python scripts `XiosGenRmp.py` and `XiosWeightsToOasis.py` can be used to chain tests for different pairs of grids and the script `XiosWeightsToOasis.py` to transform the weight file produced to the OASIS3-MCT format.

5.4.2. XIOS particularities and options

No specific options besides the normalisation options were activated for the different regridding tested i.e. CONSERV DESTAREA, CONSERV FRACAREA, CONS2ND FRACAREA.

We note here that by default for the 2nd O conservative remapping, the algorithm in XIOS, as for YAC and ESMF, will automatically fall back to a 1st O conservative remapping in cases where the gradient computation fails, for example due to a lack of neighbours, which can occur near the sea-land mask borders.

With XIOS, the mesh edges can be described with great circle or latitude circles and this is automatically defined by the grid type. For unstructured and curvilinear (i.e. torc, nogt, icos, icoh in our case), great circles are used. For longitude-latitude (i.e. bggd in our case), and gaussian reduced (i.e. sse7), latitude circles are used for edges located on a latitude circle and great circles are used otherwise.

6. The environment for calculating benchmark metrics compare_interpolations

A scripting environment was developed to calculate the benchmark metrics for the four regridders considered. The regridding weights have to be pre-calculated with the other environments described for each regridder in Sect. 5 above. This environment is available on CERFACS git developer repository at https://nitrox.cerfacs.fr/globc/OASIS3-MCT/oasis3-mct_other/tree/master/compare_interpolations . Its version dated 2021/07/22 tagged “d5b8bc99be24270aaaf2f99eb1a7ad80a4365b19” is also available in the tar file compare_interpolation.tar on Zenodo at https://doi.org/10.5281/zenodo.5342778 .

6.1. General description

The environment `compare_interpolations` allows:

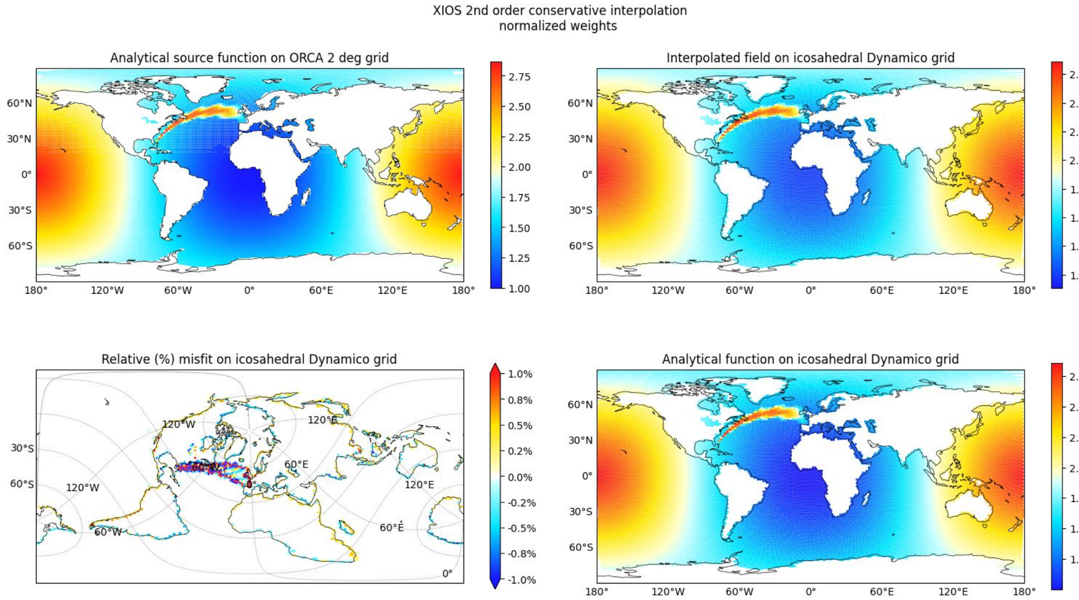
- to remap an analytical function with OASIS3-MCT using weights previously computed by a (possibly different) regridder with a specific remapping method from a specific source grid to a specific target grid ;

- to compute the related benchmarking metrics: mean misfit, maximum misfit, RMS misfit, L1, L2, Lmin, Lmax, source global conservation or target global conservation (see Sect. 4.4)
- to generate output text files (csv and txt) of these metrics and/or to generate 2D plots of the analytical function, its remapping and the relative misfit (in png and/or vector pdf formats).

Here is an example of the text output file generated for the 2nd O conservative remapping of the gulfstream function using weights calculated by XIOS from the torc to the icos grid, `Metrics_torc_to_icos_xios_CONS2ND_FRACAREA_gulfstream.txt` :

```
REMAPPING QUALITY METRICS FOR COUPLER XIOS

Source grid: ORCA 2 deg (torc)
Destination grid: icosahedral Dynamico (icos)
Test function: gulfstream
Method: conservative interpolation of order 2
Weights are normalized by the target covered cell fraction

Global conservation metrics:
Src ana: 16.519125051217895 | Dst ana: 16.519039100337075 | Dst rmp: 16.520469909169666
|It(Dst rmp) - Is(Src ana)|/|Is(Src ana)| = 0.00814121781632899 %
|It(Dst rmp) - It(Dst ana)|/|It(Dst ana)| = 0.0086615742229339 %

Relative misfit Metric:
Max(|Dst rmp - Dst ana|/|Dst ana|) = 47.8844765698243 %
Mean(|Dst rmp - Dst ana|/|Dst ana|) = 0.09670968677098436 %
Rms(|Dst rmp - Dst ana|/|Dst ana|) = 1.003348706138882 %

L1 Metric:
It(|Dst rmp - Dst ana|)/It(|Dst ana|) = 0.0006326293549358778
L2 Metric:
sqrt(It(|Dst rmp - Dst ana|^2))/sqrt(It(|Dst ana|^2) = 0.007478044239826028
Linf Metric:
Max|Dst rmp - Dst ana|/Max|Dst ana| = 0.2604558642666471
Lmin Metric:
(min(Dst ana) - min(Dst rmp))/Max|Dst ana| = 0.0400516793677859
Lmax Metric:
(Max(Dst rmp) - Max(Dst ana))/Max|Dst ana| = 0.00017238015584725229
```

and an example of the associated 2D plot:


XIOS 2nd order conservative interpolation
normalized weights
Analytical source function on ORCA 2 deg grid
Interpolated field on icosahedral Dynamico grid
Relative (%) misfit on icosahedral Dynamico grid
Analytical function on icosahedral Dynamico grid
60°N
30°N
0°
30°S
60°S
180°
120°W
60°W
0°
60°E
120°E
180°
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
1.0%
0.8%
0.5%
0.2%
0.0%
-0.2%
-0.5%
-0.8%
-1.0%

The compare_interpolations/ directory contains the following sub-directories and files:

```
bin
build
clean_workspace.sh
grids_compare
inputRun_test.json
inputSuite_ESMF.py
inputSuite_MTR.py
inputSuite_SCRIP.py
inputSuite_XIOS.py
inputSuite_YAC.py
inputSuite_test.py
python
src
```

- `bin/` directory contains the executables of the toy models
- `build/` directory contains the Makefile used to compile the sources of the toy models; it also contains the compilation objects.
- `grids_compare/` directory contains:
    - the file `grids.nc` describing the different grids following the OASIS3-MCT format
    - the file `areas.nc` describing the different grid cell areas following the OASIS3-MCT format

- one directory per regridder `XXX_masks/`, where `XXX` is the regridder name containing the binary and fractional masks of the different grids adapted to each ocean grid, i.e. one for nogt and one for torc[6];
  - the file `masks_no_atm.nc` which contains the masks of all grids with the atmosphere ones being totally unmasked (i.e. all points are valid)

- `python/` directory contains the following python scripts:
  - `ConfigsOasisToy.py` defines a class TestConfig containing the configuration of the regridding
  - `GridsOasisToy.py` defines the characteristics of all the grids used in the comparison
  - `MetricsOasisToy.py` computes the remapping quality metrics
  - `PlotsOasisToy.py` generates the 2D-plots of the remapping
  - `RunOasisToy.py` can be used to execute one regridding for one regridder, one method, one couple of grids and one analytical function
  - `SuiteOasisToy.py` can be used to execute a regridding suite for some regridders (generally only one), some methods, some pairs of grids and some analytical functions
- `src/` contains the sources of the toy models coupled by OASIS3-MCT which performs the regridding

6.2. Using `compare_interpolations` to calculate the benchmark metrics

The script `RunOasisToy.py` can be used to perform one regridding after filling in the labels of the `inputRun_test.json` file, i.e. the names of the regridder, the algorithm (method), the weight directory, the grid directory, the source and target grid names, the metric output directory, and a working directory (otherwise the outputs are directly written in the execution directory `compare_interpolations`).

Here is an example of a `inputRun_test.json` file:

```
{
    "remapper": {
        "software": "XIOS",
        "method": "conservative",
        "order": "1",
        "normalization": true,
        "truearea": false,
        "input_dir":"$OASIS3_COMP_DIR/RMP_WEIGHTS"
    },
    "geometry_files": {
        "input_dir": "$OASIS3_COMP/grids_compare"
    },
```

[6] To set up a consistent atmosphere-ocean system and have a well-posed coupled problem, we adopted the following best practice. The original sea-land mask of the ocean model is taken as is. For the atmosphere model, the fraction of water in each cell is defined by the conservative remapping of the ocean mask on the atmospheric grid performed with the specific regridder used in the current test. Then, the atmospheric coupling mask is adapted associating a valid/active index to cells containing at least a surface fraction (1/1000) of water; under 1/1000 of water, the atmospheric cell is considered completely masked. This method ensures that the total sea and land surfaces are the same in the ocean and atmosphere models, allowing global conservation of sea or land integrated quantities.

```
        "source": {
            "grid": "nogt"
        },
        "destination": {
            "grid": "icoh"
        },
        "testfunction": "vortex",
        "metrics": {
            "output_dir": "."
        },
        "plotting": {
            "show": true,
            "png": false,
            "multipagepdf": false,
            "output_dir": ".",
            "misfit_1%": false,
            "draw_edges": false
        },
        "toy_outputs": {
            "save": false
        }
    }
```

The execution command is:

```
> RunOasisToy.py inputRun_test.json
```

RunOasisToy.py compiles the toy models, executes coupling with OASIS3-MCT including the specified regridding, computes and writes the metrics in csv and txt files, and if requested draw the 2D-plots in png and/or vector pdf files.

The script `clean_workspace.sh` can be used to clean the execution directory.

The script `SuiteOasisToy.py` can be used to perform a suite of regriddings using one regridder filling the labels of the `inputSuite_XXX.py` file where `XXX` is the regridder name or `inputSuite_test.py`. SuiteOasisToy executes a loop on RunOasisToy.

The execution command is, for example for ESMF:

```
> SuiteOasisToy.py inputSuite_ESMF.py
```

7. Results

All benchmark metrics were calculated for:

- all pairs of grids: torc-bggd, torc-icos, torc-sse7, nogt-bggd, nogt-icos, nogt-sse7 in both directions, icos->icoh (for the vortex function for 1st and 2nd O conservative FRACAREA remapping), and nogt->icoh (for the vortex function for 2nd O conservative FRACAREA remapping) (see Sect.4.1)
- the four functions: sinusoid, harmonic, vortex, gulfstream (see Sect. 4.2)
- for all regridders: SCRIP (+SCRIP-L, i.e. with Lambert projection for 1st & 2nd O conservative remapping), YAC, ESMF and XIOS (see Sect. 2)
- for all algorithms: NEAREST-NEIGHBOUR, BILINEAR, 2nd O non-conservative, 1st O conservative, 2nd O conservative, except when the regridder does not support the algorithm, such as e.g. BILINEAR for XIOS (see Sect. 4.3).

Results are gathered in a tar file, Regridding_Benchmark_metrics.tar, available on Zenodo at https://doi.org/10.5281/zenodo.5343166. The list of the individual files contained in the tar file is provided at Appendix 2. Plots of all results are also gathered in a tar file, Regridding_Benchmark_metrics_plots.tar, available on Zenodo at https://doi.org/10.5281/zenodo.5347696; the list of individual plots is provided at Appendix 3.

We cannot discuss all metrics obtained but we illustrate here the main conclusions of our analysis with specific examples.

7.1. nearest-neighbour

The nearest-neighbour algorithm is available in SCRIP, ESMF and YAC.

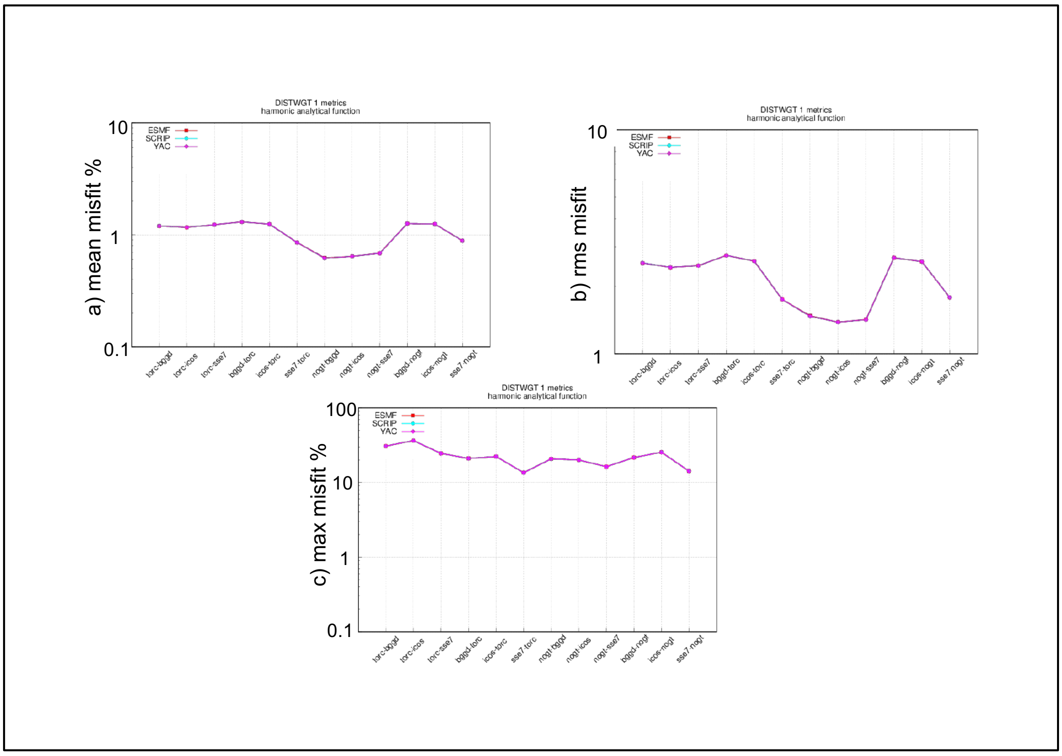


Figure 4 – a) mean, b) rms and c) maximum misfit for the different pairs of grids for the harmonic function for NEAREST-NEIGHBOUR for ESMF, SCRIP and YAC

Fig. 4 shows the mean, rms and maximum misfit for the different pairs of grids for the harmonic function. The three regridders produce almost exactly the same (and very reasonable) results: the curves are superimposed and not distinguishable; this is also true for the other functions (not shown).

We note here that the function used has a strong impact on the maximum of the misfit, as illustrated on Fig.5, which shows the maximum misfit for the different pairs of grids for the 4 functions. The maximum misfit seems to be directly linked to the gradient of the function, being much higher for the gulfstream and the harmonic functions than, for example, the slowly varying sinusoid function.

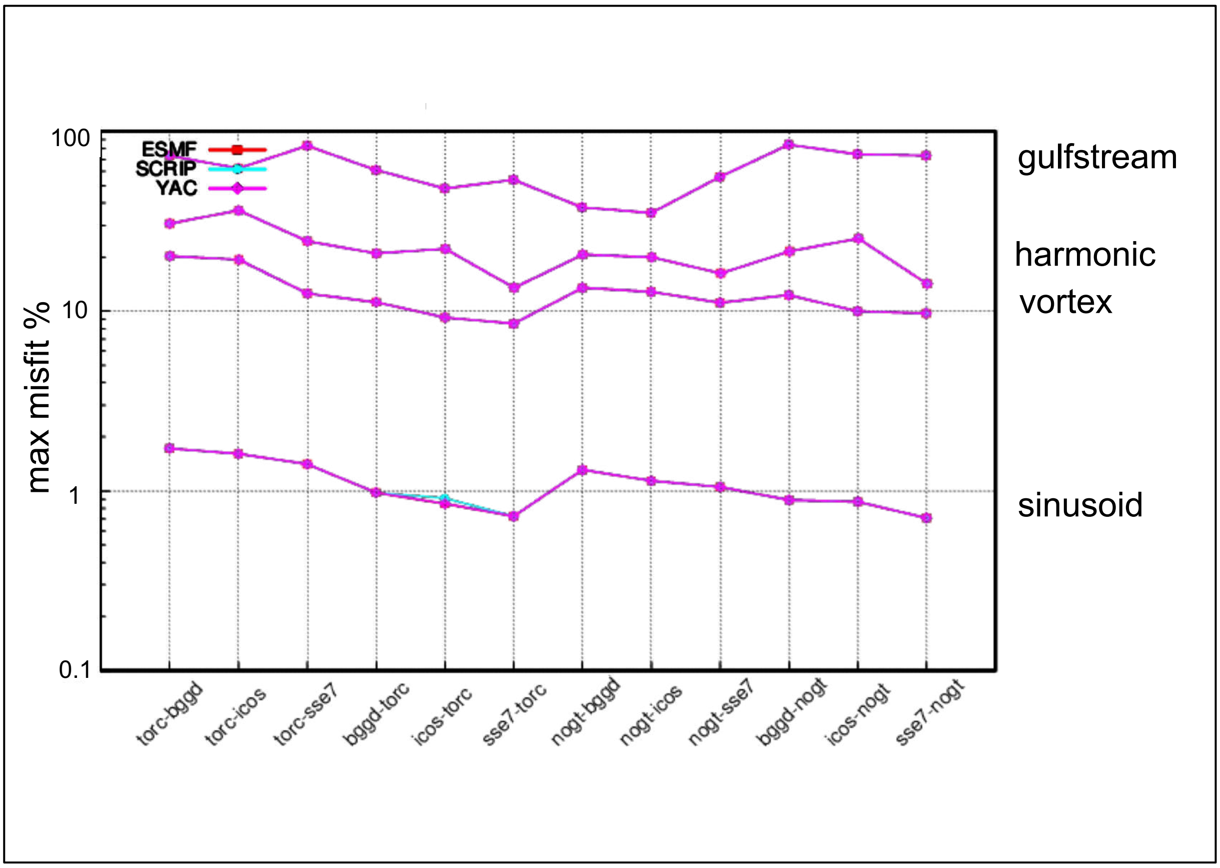


Figure 5 – Maximum misfit for the different pairs of grids for the four functions for NEAREST-NEIGHBOUR for ESMF, SCRIP and YAC

7.2. BILINEAR (or equivalent)

BILINEAR algorithms (or equivalent) are available in SCRIP, ESMF and YAC. Fig. 6 and Fig. 7 show the mean, rms and maximum misfit and Lmax for the different pairs of grids for these three regridders, respectively for the vortex function and for the gulfstream function. The algorithm in YAC, which, as noted in Sect. 4.3.2, is not a pure bilinear algorithm but implements an inverse distance weighting of the vertices of the source cell enclosing the target point, is less accurate on average (the mean misfit is higher on average) and leads to more smoothing (Lmax is more negative). For the gulfstream function, we note some high maximum misfit for ESMF for torc->sse7 and torc->bggd; these high misfits also appear for ESMF patch algorithm and are analysed in Sect. 7.3.

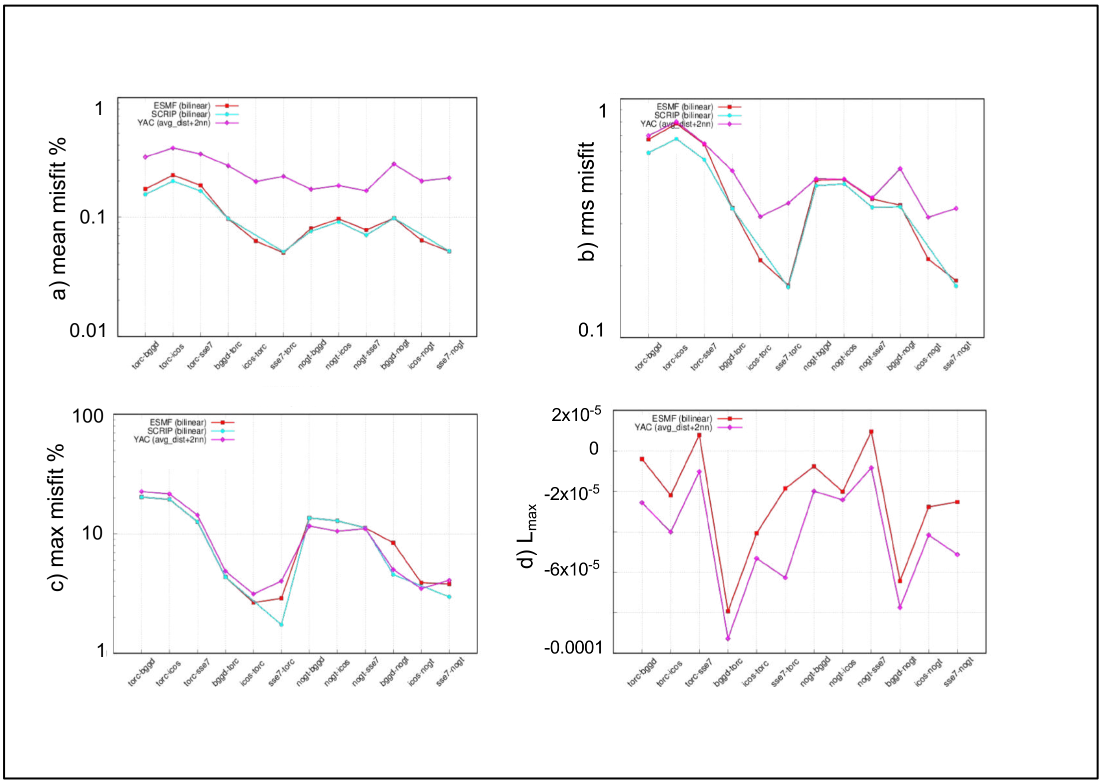


Figure 6 – a) mean, b) rms and c) maximum misfit and d) Lmax for the different pairs of grids for the vortex function for BILINEAR or equivalent for ESMF, SCRIP and YAC (only ESMF and YAC for Lmax).

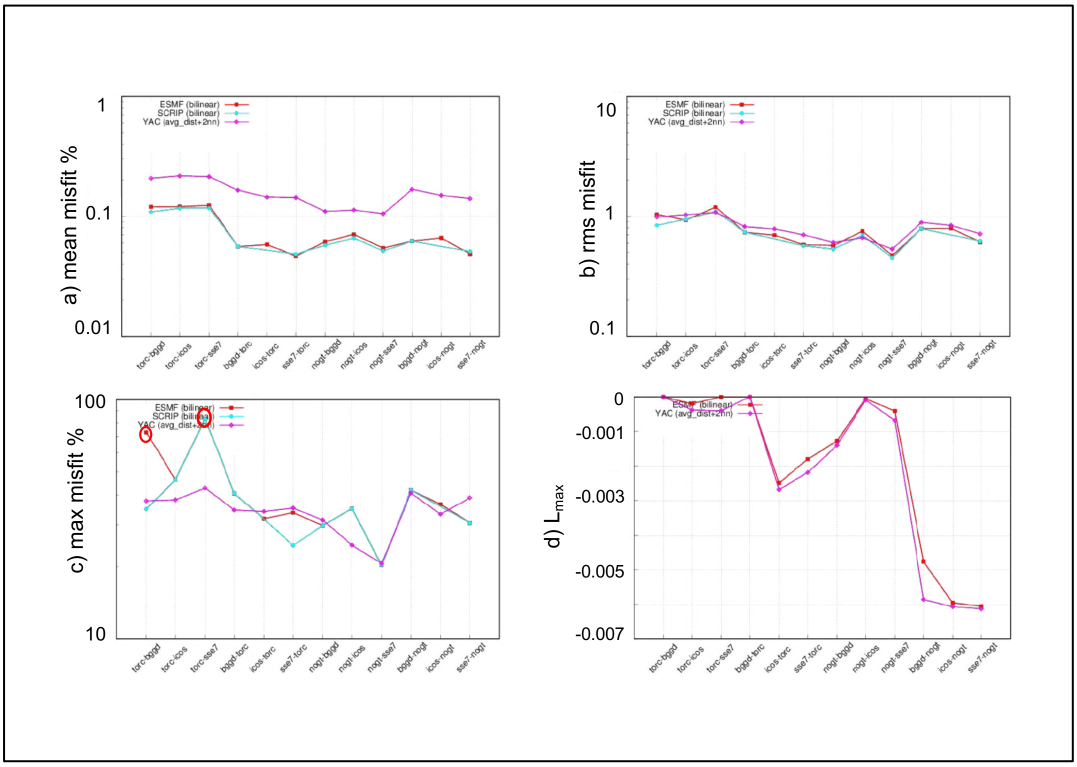


Figure 7– a) mean, b) rms and c) maximum misfit and d) Lmax for the different pairs of grids for the gulfstream function for BILINEAR or equivalent for ESMF, SCRIP and YAC (only ESMF and YAC for Lmax).

7.3. 2nd O non-conservative

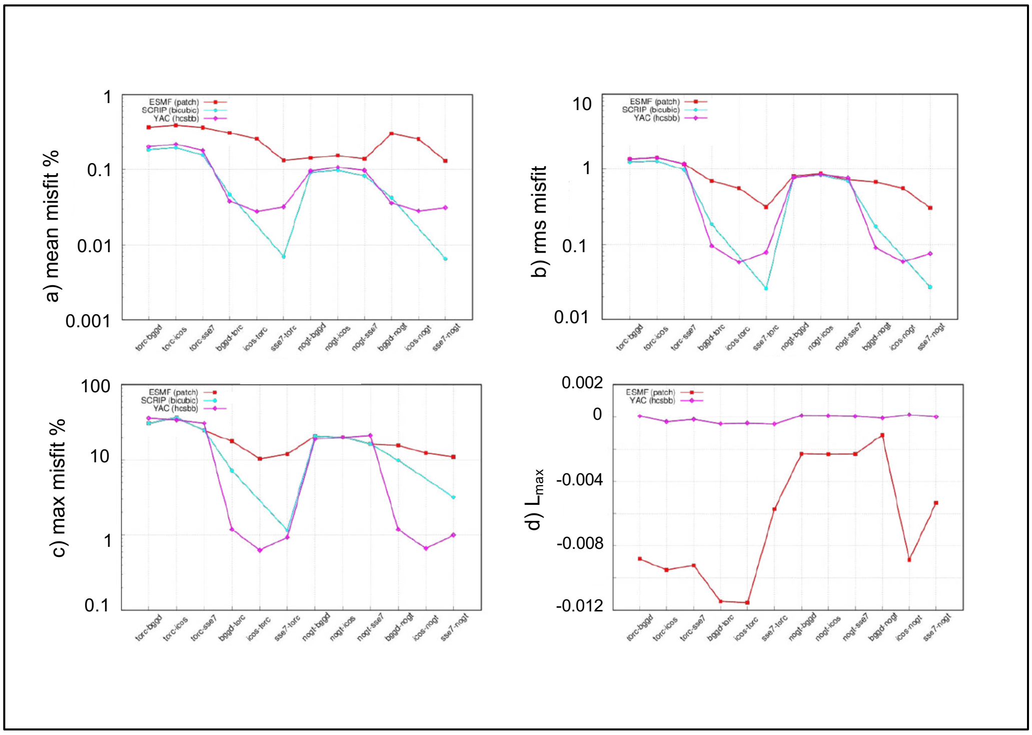


Figure 8 – a) mean, b) rms and c) maximum misfit and d) Lmax for the different pairs of grids for the harmonic function for 2nd order non-conservative for SCRIP, ESMF and YAC (only ESMF and YAC for Lmax)

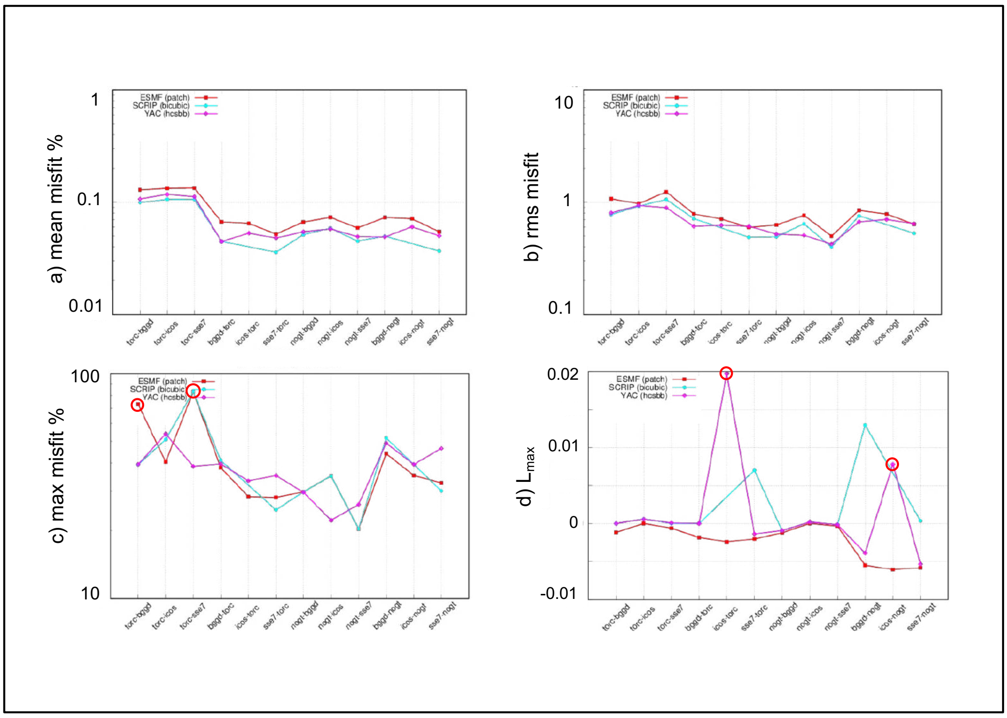


Figure 9 – a) mean, b) rms and c) maximum misfit and d) Lmax for the gulfstream function for 2nd O non-conservative algorithm for SCRIP, ESMF and YAC

2^{nd} O non-conservative algorithms are available in SCRIP, ESMF and YAC (see details in Sect.4.3.3). Fig.8 and Fig 9 show the mean, rms and maximum misfit and Lmax for the different pairs of grids for these three regridders for respectively the harmonic function and the gulfstream function.

On average, SCRIP bicubic algorithm gives slightly better results for certain couple of grids and ESMF patch algorithm gives slightly less accurate results. The patch averaging in ESMF (not present in SCRIP nor in YAC) smooths the resulting function and prevents over- or undershoots, as can be seen by the more negative values for Lmax.

On Fig. 9 c), we note some high maximum misfit for ESMF for torc->sse7 and torc->bggd for the gulfstream function (not for the other functions, see for example results for the harmonic function on Fig. 8 c)). These anomalous points also appear for the BILINEAR regridding, as noted in Sect. 7.2. The 2D plots of the misfit in the gulfstream function for the torc->bggd regridding shown on Fig. 10 indeed show one anomalous value near the coast (in yellow) for ESMF patch algorithm. This particular case is currently under investigation with ESMF developers.

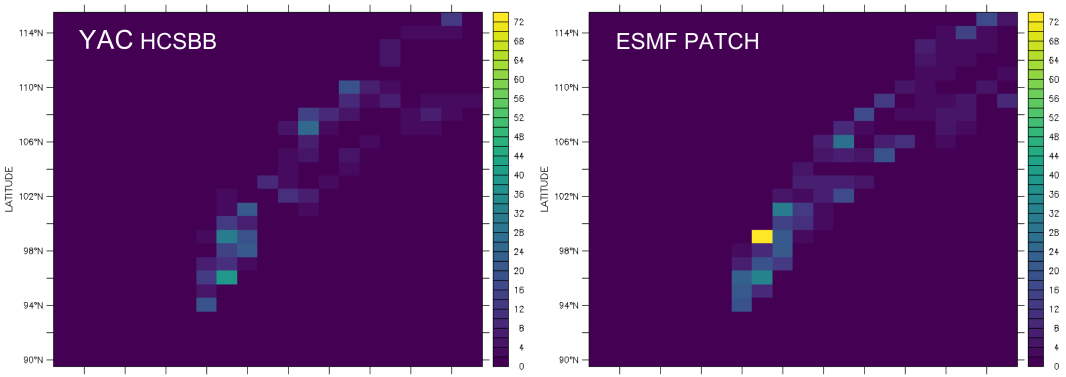


Figure 10 – Misfit for the remapped gulfstream function in the gulfstream region for the HCSBB algorithm for YAC (left) and the patch algorithm for ESMF (right)

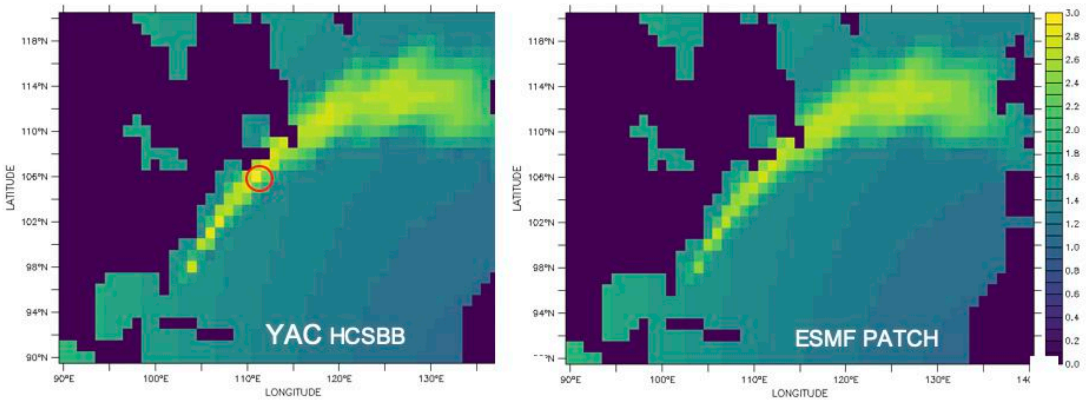


Figure 11 – 2D plots of the regridded field for the gulfstream function with HCSBB algorithm for YAC (left) and the patch algorithm for ESMF (right)

On Fig. 9 d), we also see that YAC shows some overshoots for icos-torc and icos-nogt for gulfstream (that do not appear for the other functions). Fig. 11 shows 2D plots of the regridded field in the gulfstream region, which shows that indeed, compared to ESMF, YAC gives higher,but a priori non-anomalous, values in the centre of the gulfstream.

The general conclusion is therefore that ESMF and YAC present reasonable results for the 2nd O non-conservative regridding. Given the averaging operation included in ESMF patch algorithm, ESMF gives slightly less accurate results but prevents over- and under-shoots. The only particular problems are for ESMF for the gulfstream function for torc->sse7 and torc->bggd, for which the remapped function shows one anomalous grid point near the coast; this is also the case for the BILINEAR regridding.

7.4. 1st O conservative with DESTAREA normalisation (CONSERV DESTAREA)

To detect specific problems, we analysed the 1st O conservative remapping with DESTAREA normalisation, as this normalisation does not result in a cancellation of errors (see Sect. 4.3.4).

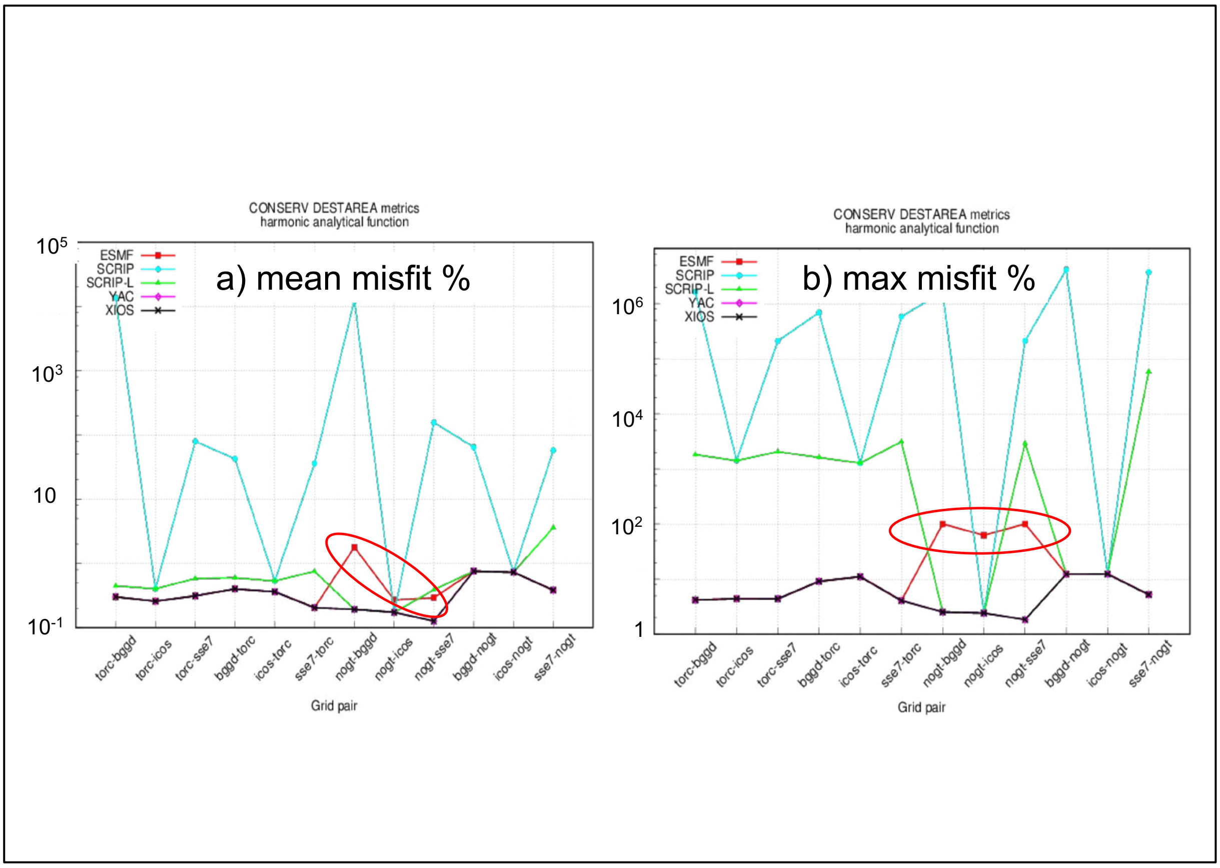
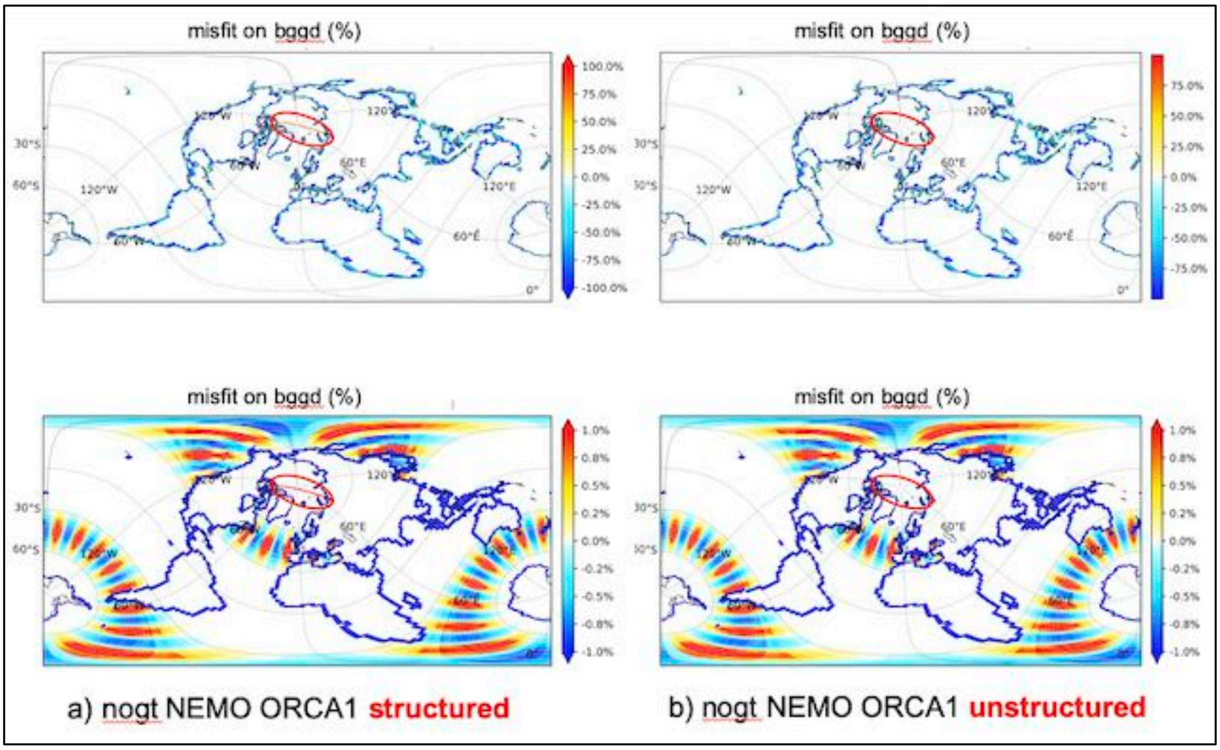


Figure 12 –a) mean and b) maximum misfit for ESMF, SCRIP, SCRIP-L, YAC and XIOS for the 1st O conservative remapping with DESTAREA normalisation for the different pairs of grids for the harmonic function.

Fig. 12 shows the mean and the maximum misfit for the harmonic function. For this function (and for the three others - not shown), we distinctly see that, besides SCRIP/SCRIP-L that presents many issues that we will not further investigate here, ESMF also presents anomalous mean and maximum misfits when nogt is the source grid. These results were produced with nogt described with the SCRIP format (see Sect. 5.3.3)

We interacted with ESMF developers to solve this problem. Although the problem per se is still under investigation, we found that one way to bypass it is to describe the nogt grid as an unstructured grid (see Sect. 5.3.3).



a) nogt NEMO ORCA1 structured

b) nogt NEMO ORCA1 unstructured

Figure 13 –Misfit for ESMF for the 1st O conservative remapping with DESTAREA normalisation for nogt->bggd for the harmonic function for a) nogt described with the SCRIP format, and b) for nogt described with the unstructured format.

Fig. 13 shows the 2D plot of the misfit (between -100% and +100% at the top and between -1% and +1% at the bottom) of the remapped function for the nogt->bggd remapping for nogt described a) with the SCRIP format and b) with the unstructured format. The problem, clearly linked to the North fold of the NEMO, disappears when nogt is described as an unstructured grid.

All metrics for ESMF for conservative algorithms (CONSERV DESTAREA, CONSERV FRACAREA, CONS2ND FRACAREA) have therefore been calculated with describing nogt as an unstructured grid[7].

Fig. 14 shows the global conservation metric for the four functions for all regridders for the different pairs of grids. It is again very clear that the SCRIP/SCRIP-L library presents some important problems with the conservative remapping but ESMF, YAC and XIOS show similar and very reasonable results.

[7] We note here however that describing torc as unstructured leads the remapping calculation to fail unless masked cells are explicitly removed from the grid, even if the -ignore_degenerate option is activated. This is linked to the presence of collapsed cells in torc. After interacting with ESMF developers, this type of cells now receives a special treatment when torc is described as structured; this treatment should also be applied in ESMF when torc is described as unstructured.

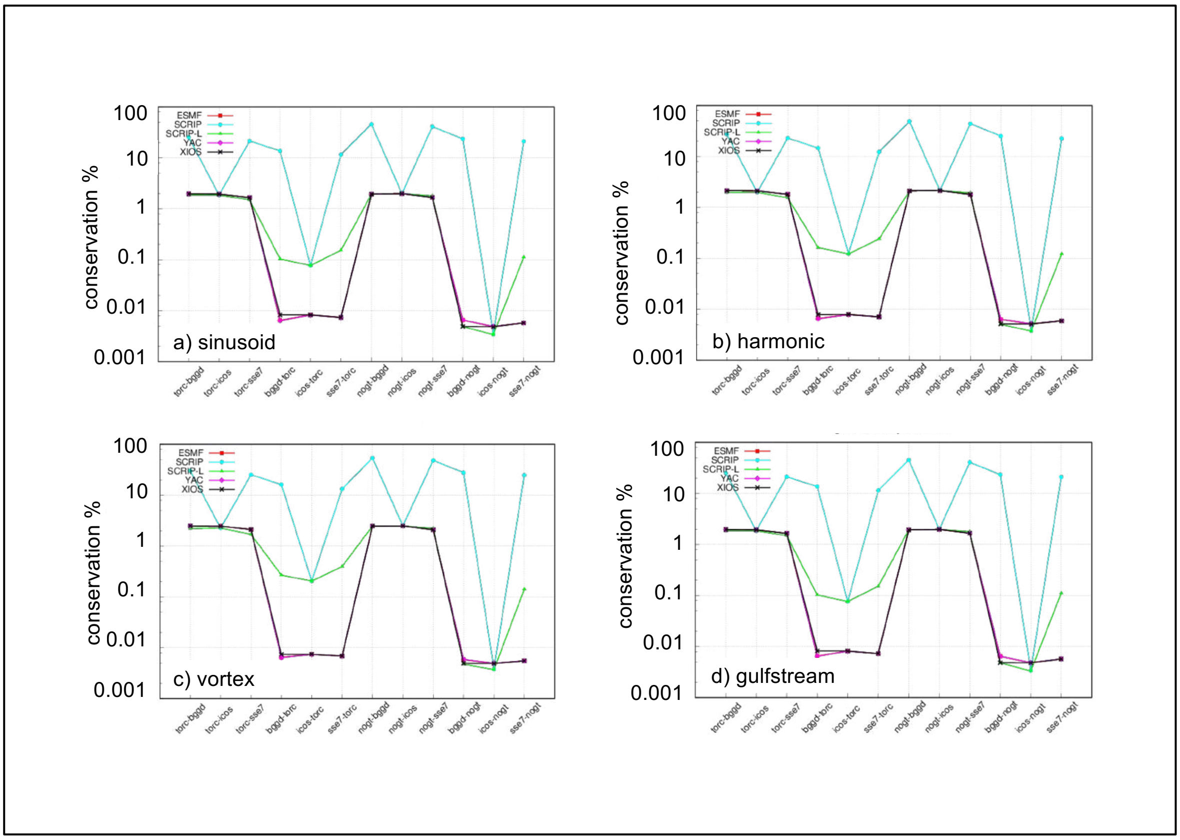


Figure 14 – Conservation (%) for ESMF, SCRIP, SCRIP-L, YAC and XIOS for the 1st O conservative remapping with DESTAREA normalisation for the different pairs of grids for the 4 functions: a) sinusoid, b) harmonic, c) vortex, d) gulfstream function

The general conclusion is therefore that ESMF (with the NEMO nogt grid described as unstructured), YAC and XIOS present very similar and good results for the 1st O conservative remapping with DESTAREA normalisation.

7.5. 1st O conservative with FRACAREA normalisation (CONSERV FRACAREA)

Fig. 15 shows the max misfit for the 1st O conservative regridding with FRACAREA normalisation for the four functions. All regridders have the same max misfit, except SCRIP and SCRIP-L, which we will not further discuss here. For the gulfstream function (Fig. 15 d), the max misfit for torc-sse7 is particularly high; as it appears for all regridders and not for the other functions, it is probably linked to the sharp gradients of the gulfstream function.

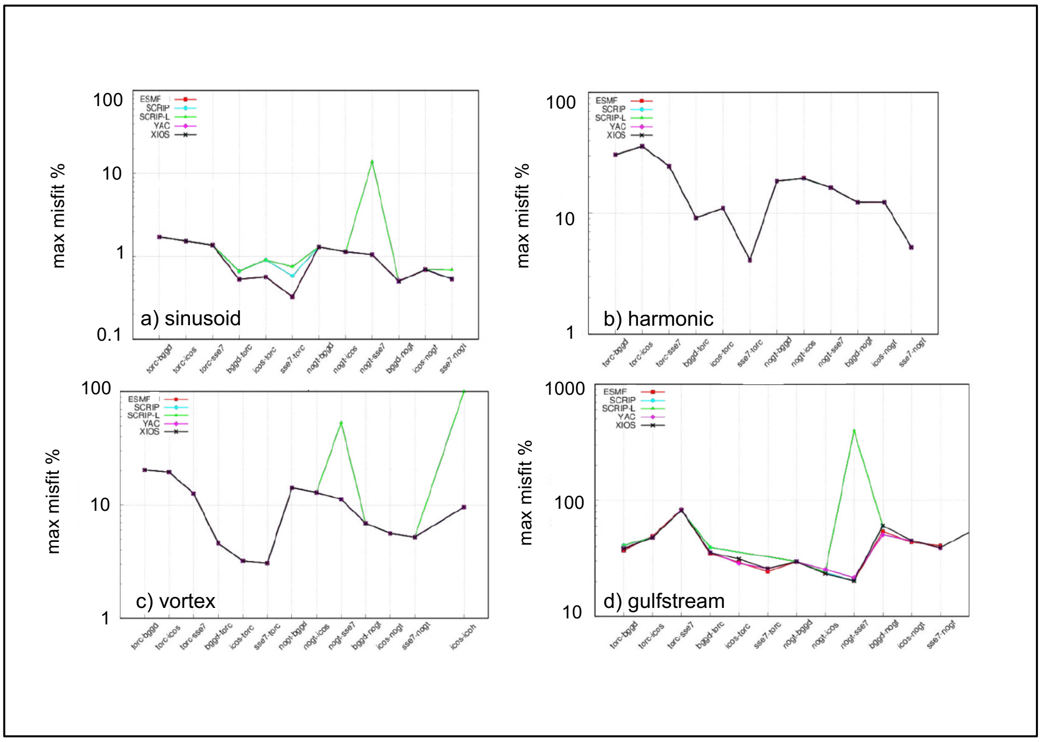


Figure 15 – Maximum misfit for ESMF, SCRIP, SCRIP-L, YAC and XIOS for the 1st O conservative remapping with FRACAREA normalisation for the different pairs of grids for the four functions: a) sinusoid, b) harmonic, c) vortex, d) gulfstream function

Fig. 16 shows the global conservation metric for the 1st O conservative regridding with FRACAREA normalisation for the four functions.

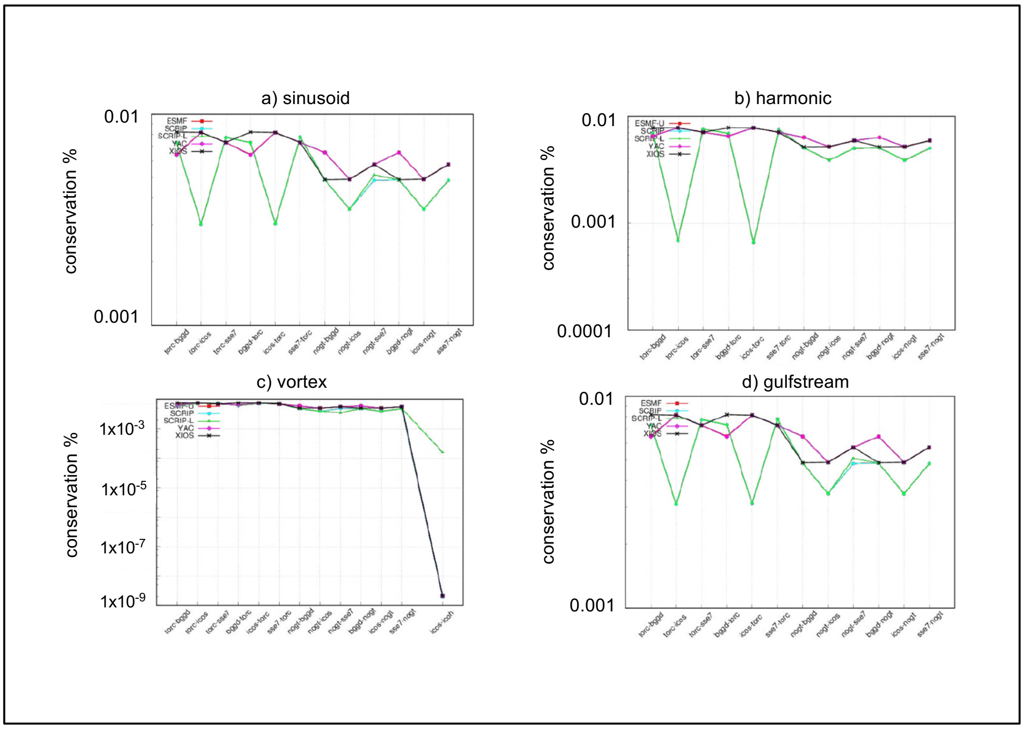


Figure 16 – Conservation (%) for ESMF, SCRIP, SCRIP-L, YAC and XIOS for the 1st O conservative remapping with FRACAREA normalisation for the different pairs of grids for the four functions: a) sinusoid, b) harmonic, c) vortex, d) gulfstream

For vortex, the conservation of the icos-icoh is also shown. ESMF and YAC curves are indistinguishable from one another. All regridders for all functions show pretty reasonable conservation, the values being less than 0.01% in all cases.

The overall conclusion is therefore that ESMF, YAC and XIOS present very similar and good results for the 1st O conservative remapping with FRACAREA normalisation.

7.6. 2nd O conservative with FRACAREA normalisation (CONS2ND FRACAREA)

Fig. 17 and 18 shows the mean, max and rms misfit and conservation for the 2nd O conservative remapping for the different pairs of grids for all regridders for the harmonic function and the gulfstream function respectively. Besides SCRIP and SCRIP-L, which we will not further analyse here, we see that all regridders show more or less the same behaviour with good global conservation. This is not surprising as they all implement the same algorithm based on Kritsikis et al (2017). The only particularity seems to be for ESMF when the source grid is the icosahedral one (icos) which shows a relatively high mean misfit.

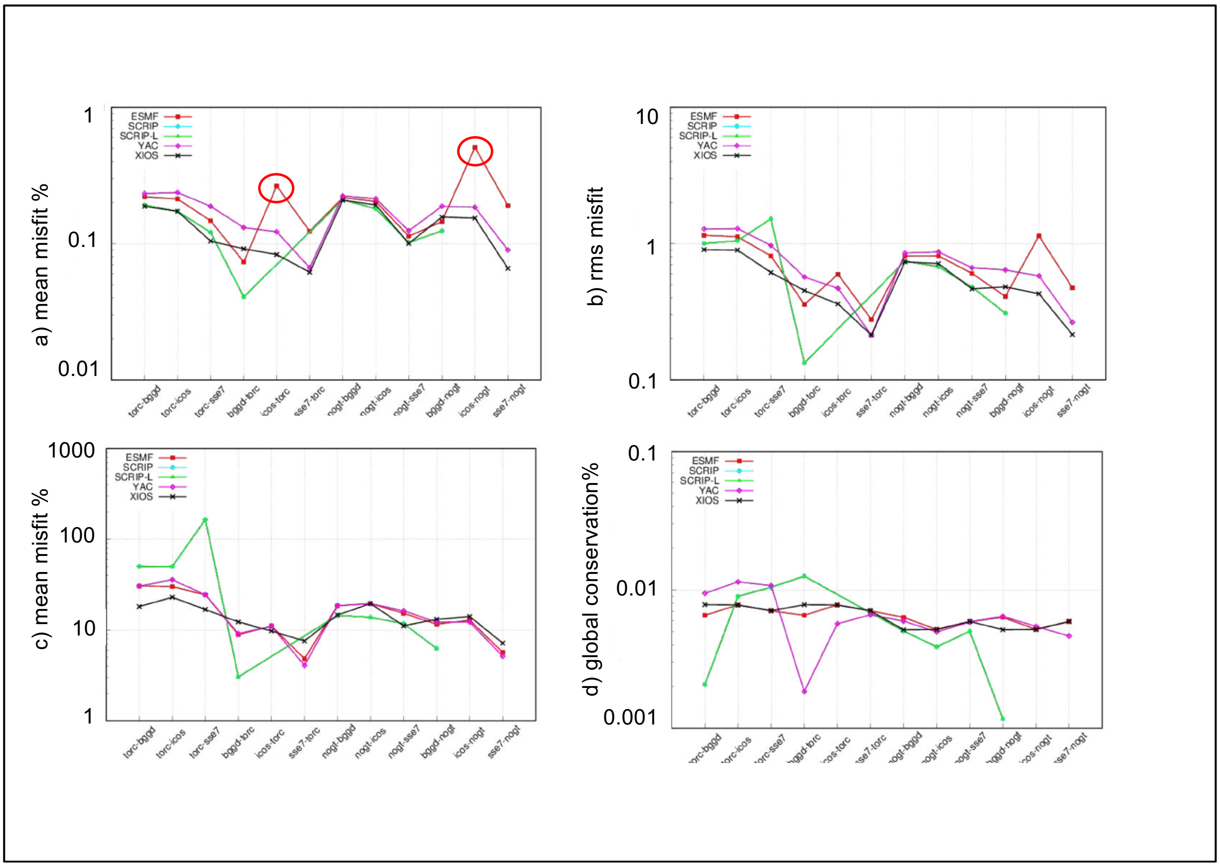


Figure 17 – a) mean, b) rms and c) maximum misfit and d) global conservation for the different pairs of grids for the harmonic function for 2nd O conservative remapping

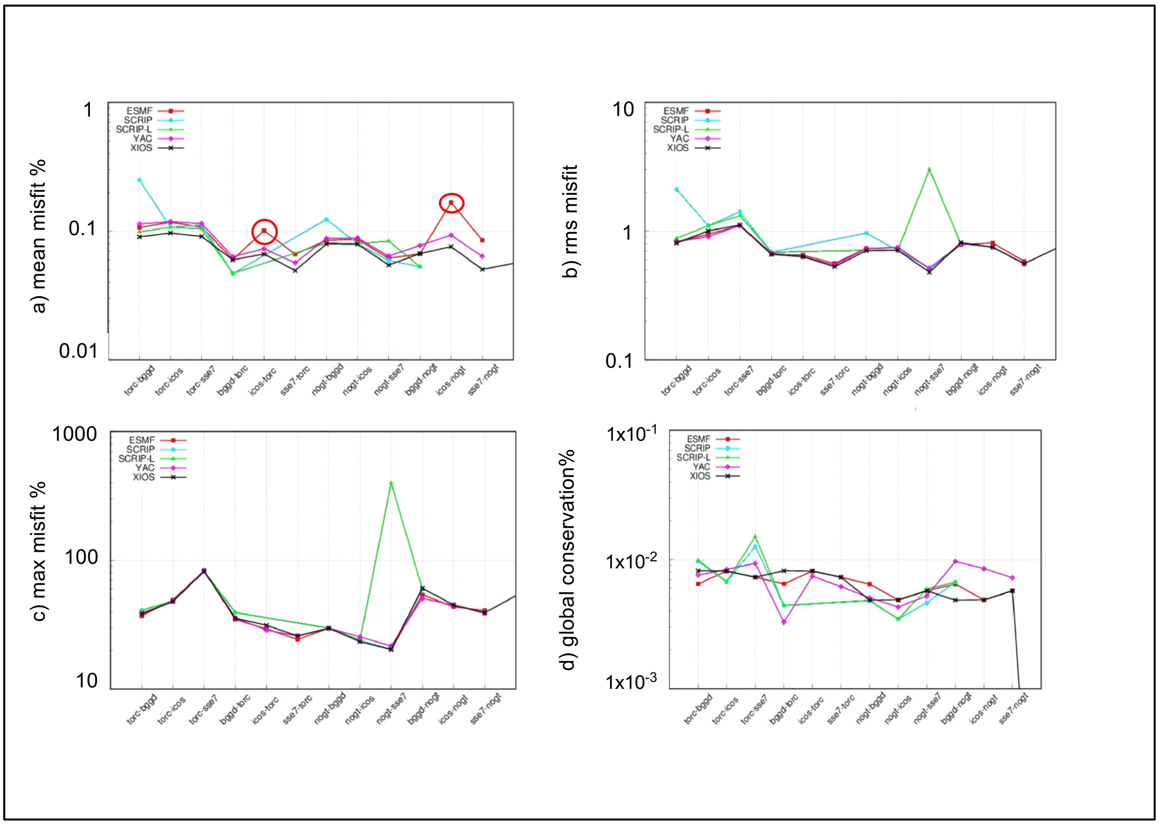


Figure 18 – a) mean, b) rms and c) maximum misfit and d) global conservation for the different pairs of grids for the gulfstream function for 2nd O conservative remapping

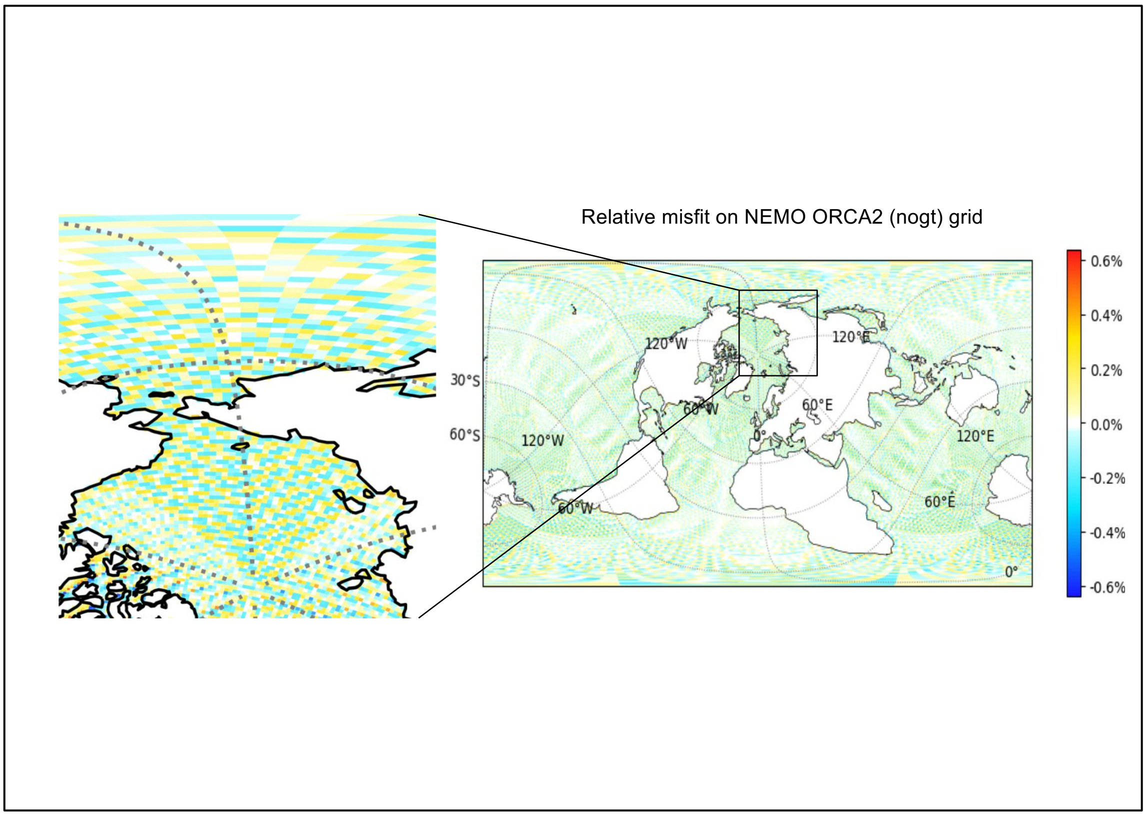


Figure 19 - Misfit on the target grid nogt for the icos->nogt ESMF 2nd O conservative remapping, with a zoom on the left

Fig. 19 shows a 2D plot with a zoom of the misfit for the icos->nogt case. The misfit shows an alternating positive and negative pattern which causes the relatively high misfit for ESMF. Work in underway with ESMF developers to solve this issue.

Figure 20 shows Lmin and Lmax for the 2nd O conservative remapping for the gulfstream function, which present some outstanding values (the other functions show no real outstanding values). XIOS shows a strong undershoot for torc-icos, as shown by Lmin, and a strong overshoot for bggd-nogt as shown by Lmax.

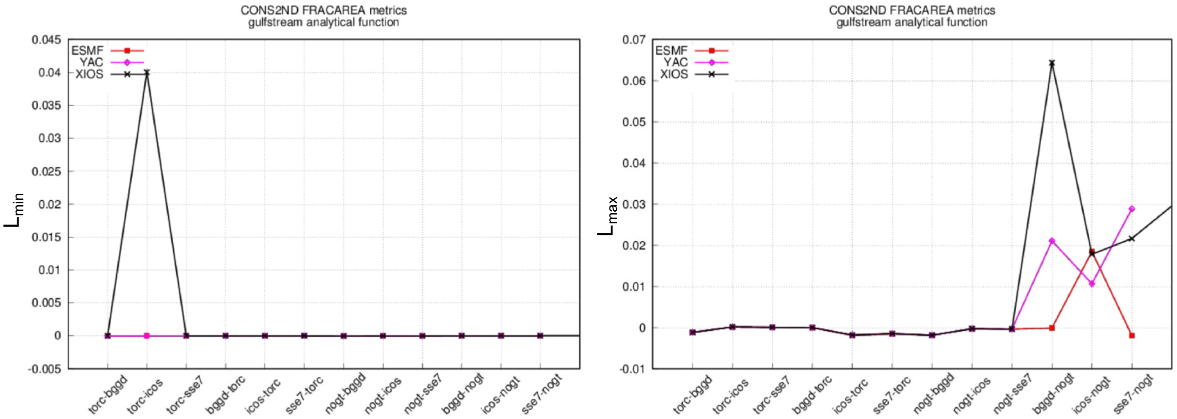


Figure 20 – Lmin and Lmax for the different pairs of grids for the gulfstream function for 2nd O conservative remapping

To understand XIOS undershoot for torc->icos, we looked at the 2D misfit in the gulfstream region for XIOS, ESM and YAC as shown on Figure 21.

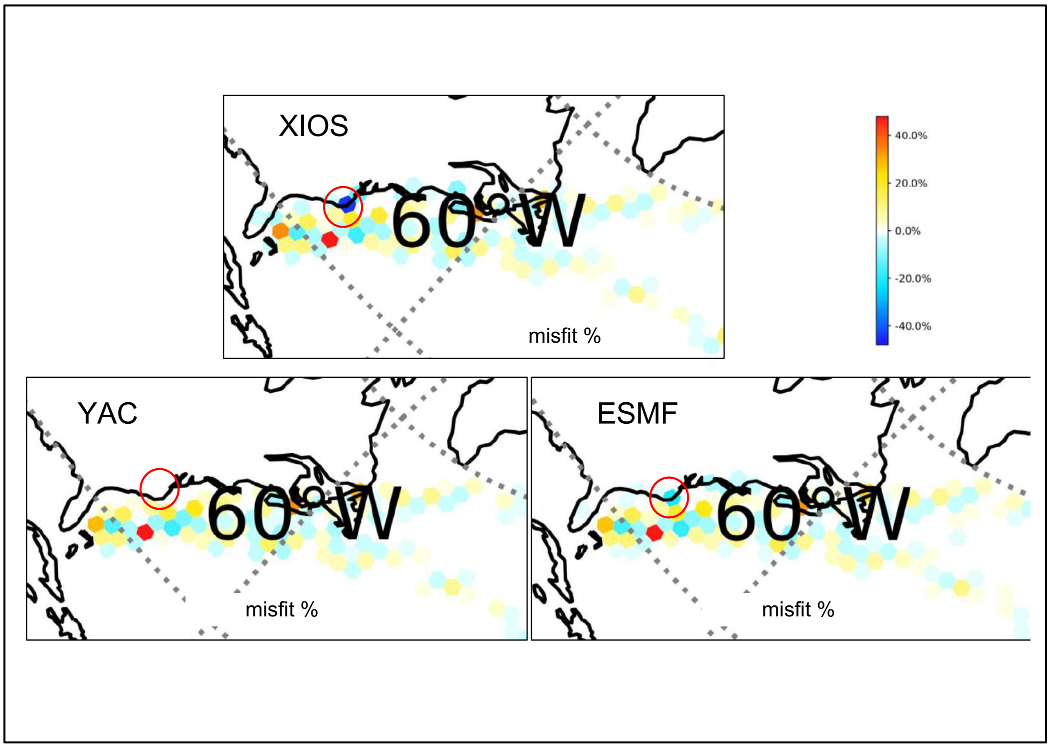


Figure 21- misfit in the gulstream region for the 2nd O conservative remapping of the gulfstream function for torc->icos for XIOS, YAC and ESMF

There is indeed one clearly outstanding point near the coast for XIOS, which causes this undershoot. ESMF also shows some negative misfit at this point but much smaller than XIOS. YAC does not show any important misfit at this point. This difference between the three regridders has to be investigated in more detail. As the three regridders are based on the same algorithm, it must be linked to some implementation differences in the way the regridders calculate the gradients and eventually switch to a 1st O conservative remapping when the gradient cannot be calculated (e.g. near the coast) or the way they define the edges of the cell meshes (see details in Sects 5.2.2, 5.3.2 and 5.4.2)

To understand XIOS overshoot for bggd->nogt, we looked at the 2D regridded gulfstream function in the gulfstream region for XIOS, ESMF and YAC as shown on Fig. 22. We see that XIOS indeed shows higher values near the centre of the gulfstream, which explains the overshoot, but this does not seem to be linked to an anomalous behaviour. By looking at 2D regridded fields for the other regridders and other pairs of grids (not shown), we checked that the other overshoots shown on Fig. 20 for Lmax are not a priori linked to any anomalous behaviours either.

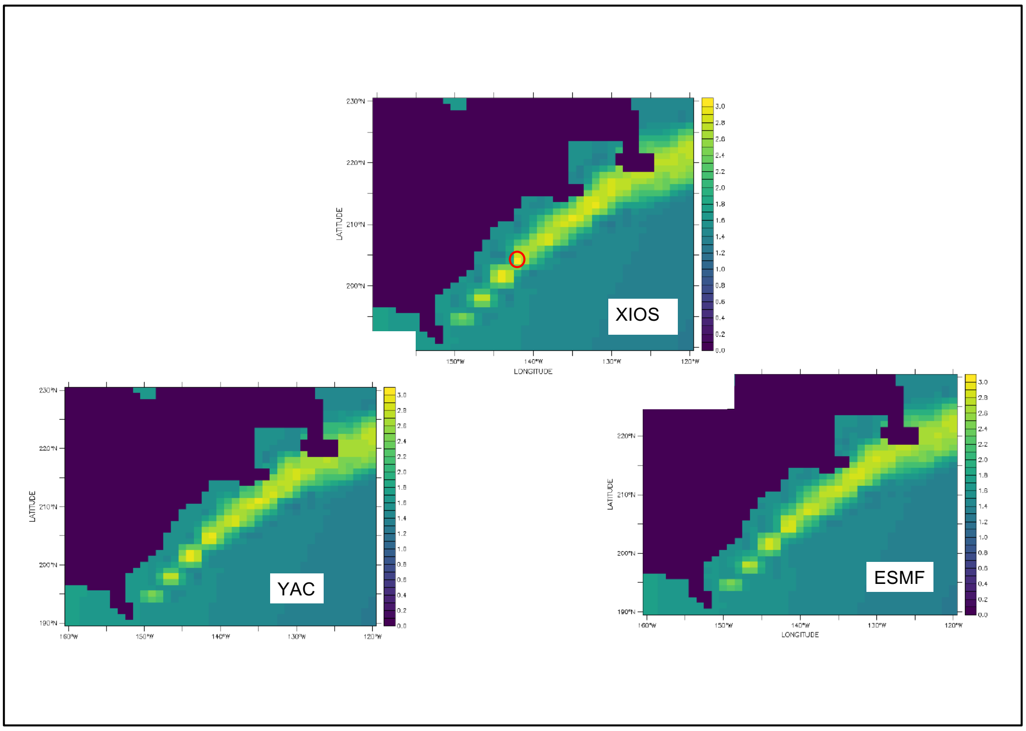
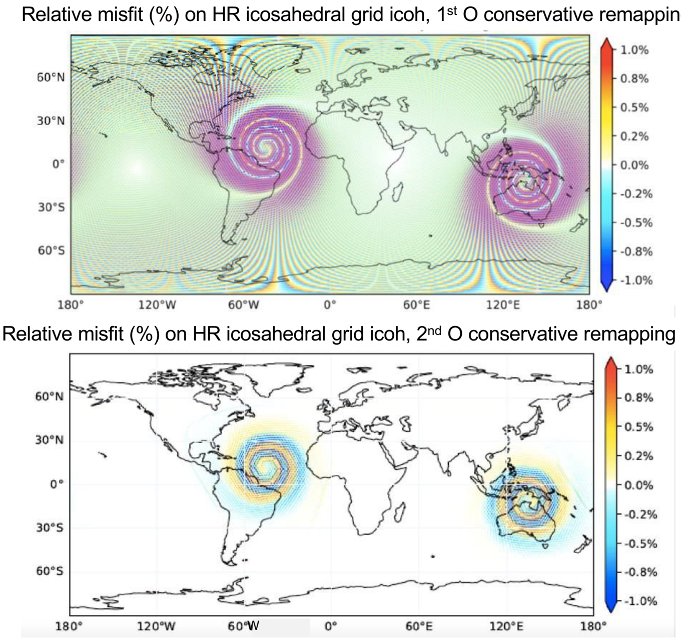


Figure 22 - 2D regridded gulfstream function in the gulfstream region for XIOS, ESM and YAC for 2nd O conservative remapping for bggd->nogt

Fig. 23 shows 2D plots of the relative misfit for the remapping of the vortex function from the low-resolution icosahedral grid icos to the high-resolution icosahedral grid icoh with YAC for the 1st O CONSERV FRACAREA at the top and for the 2nd O CONSERV FRACAREA at the bottom. We see the clear benefit of the 2nd O as compared to the 1st O when this remapping involves two grids with very different resolutions, the 2nd O allowing the reconstitution of gradients on the higher-resolution target grid. XIOS shows very similar results but not ESMF, probably because of the problem identified above for the 2nd O conservative remapping for icos->nogt, which also exists for icos->icoh (alternating positive and negative pattern in the misfit, see Fig. 19).



Relative misfit (%) on HR icosahedral grid icoh, 1st O conservative remapping

Relative misfit (%) on HR icosahedral grid icoh, 2nd O conservative remapping

Figure 23 - Relative misfit (%) on HR icosahedral grid icoh, 1st O conservative remapping

Finally, Fig. 24 shows 2D plots of the gulfstream region for the 2nd O CONSERV FRACAREA remapping of the gulfstream function for bggd-nogt with a) the regridded function on nogt, b) the analytical gulfstream function on nogt, c) the misfit of the regridded function between -60% and + 60%, and d) the misfit of the regridded function between -1% and + 1%. This figure is for XIOS but ESMF and YAC show very similar results, i.e. a regridded field with strong oscillations (i.e. positive and negative misfit values) near strong gradients, which is an expected characteristic of 2nd O conservative remapping.

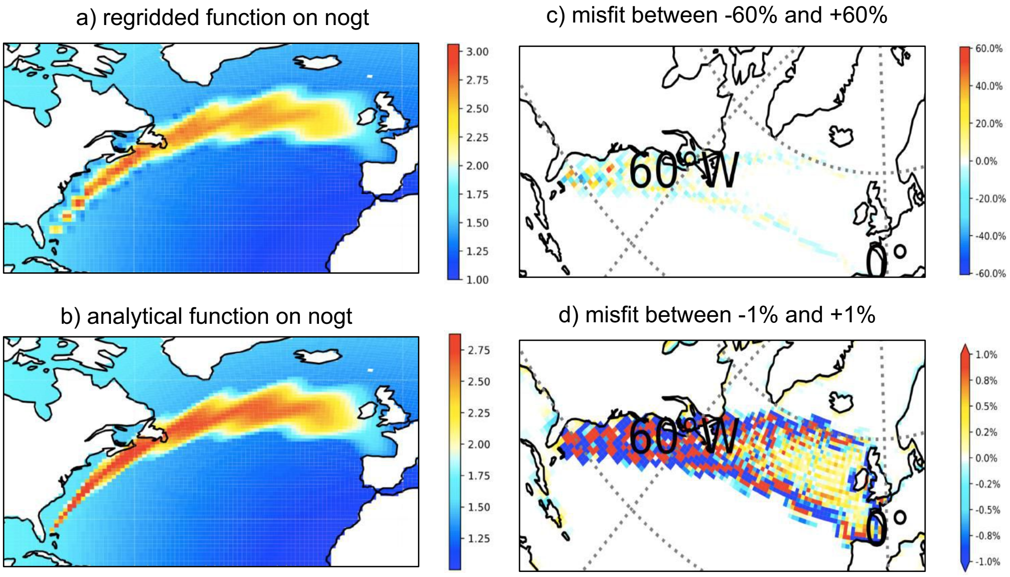

a) regridded function on nogt
c) misfit between -60% and +60%
b) analytical function on nogt
d) misfit between -1% and +1%
60°W
0°
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
60.0%
40.0%
20.0%
0.0%
-20.0%
-40.0%
-60.0%
1.0%
0.8%
0.5%
0.2%
0.0%
-0.2%
-0.5%
-0.8%
-1.0%

Figure 24 - 2D plots of the gulfstream region for XIOS for 2nd O conservative remapping for bggd->nogt with a) the regridded function on nogt, b) the analytical gulfstream function on nogt, c) the misfit of the regridded function between -60% and + 60%, and d) the misfit of the regridded function between -1% and + 1%; this figure is for XIOS but ESMF and YAC show very similar results.

In summary, YAC, ESMF, and XIOS show very similar and good results for all 2nd O conservative remappings tested with good global conservation. This is not surprising as they all implement the same algorithm based on Kritsikis et al (2017). There are however some differences, which are unexplained for now and need more analysis. One of them is the alternating positive and negative pattern in the misfit produced by ESMF for all functions when the source grid is the icosahedral grid, as shown on Figure 19. The other one is the strong undershoot produced by XIOS for one point in the gulf stream region for the gulfstream function, as shown on Figure 21.

8. Performance and scalability

The benchmark results shown in the previous section lead us to conclude that YAC, ESMF and XIOS can all be considered as high-quality regridding libraries, although a few details still need to be fixed. An additional criterion for evaluating the viability of a library is of course its performance. In this paragraph, we present a very first analysis of ESMF and XIOS performances and compare them with SCRIP. We did not yet complete this analysis for YAC, mainly because of lack of time, and as YAC has been given a somewhat lower priority after our preliminary analysis (see Sect. 3.6).

The environment used to measure the time to generate regridding weights with SCRIP, ESMF, and XIOS are the ones described in Sect. 5.1, 5.2 and 5.4. The regridding is the 1st O conservative remapping from the NEMO ORCA12 t12e source grid with 3,147x4,322 grid points to the high-resolution icosahedral grid icoh with 2,016,012 grid points[8]. The tests were realized on Cerfacs cluster kraken that has 185 Intel Skylake nodes, each with 2x18 cores @ 2.3 Ghz and 96 GB of memory with compiler Intel 18.0.1.163 and MPI Intel library 2018.1.163. The number of cores used for weights calculation varied from 8 to 1620 for SCRIP and from 1 to 900 for ESMF and XIOS.

Fig. 25 shows the time to generate regridding weights with SCRIP, ESMF and XIOS as a function of the number of cores for the t12e->icoh 1st O conservative remapping. Three tests were performed for each regridder for each number of cores to detect any instability. **We have to note here that the time for ESMF includes the time to write the weights to a disk file while it does not for SCRIP and XIOS.** For ESMF, we did not find out how to measure only the time for calculating the weights for this very first performance analysis.

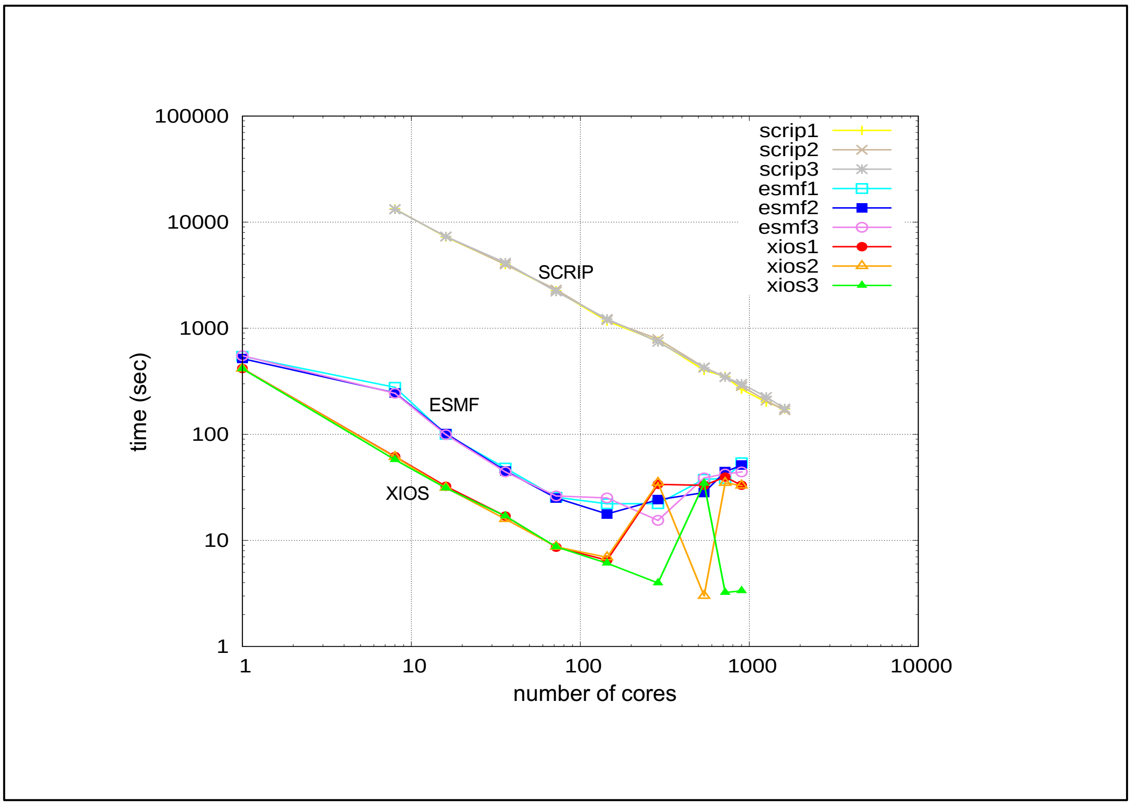


Figure 25 - Time to generate regridding weights with SCRIP and XIOS and time to generate regridding weights and write them to a file for ESMF as a function of the number of cores for the 1st O conservative remapping from the NEMO ORCA12 grid t12e (3,147x4,322) to the icosahedral icoh grid (2,016,012); three tests were performed for each regridder for each number of cores

It shows that both ESMF and XIOS have much better (O(100)) performance than the SCRIP. It also shows **XIOS seems more performant than ESMF at low numbers of cores but this is probably linked to the fact that the XIOS measure does not include the time to write**

[8] Note that for those performance measurements, we did not follow the best practice that would be to recalculate the mask of the icoh grid interpolating the mask of the t12e grid on the icoh grid. Instead, we used the icoh mask calculated for the nogt grid but this has no impact on the performances.

the weights to a file while it does for ESMF. XIOS also shows unstable behaviour for more than ~200 cores, which still has to be investigated.

Many factors can in fact explain the difference of performance between the regridders, such as the parallel decomposition, the search algorithm used to identify the source cells used for each target cell, the implementation of the edge intersection calculation given different definition of the mesh edges, etc. This first analysis does not address a deeper investigation of all these factors, but at least it is quite reassuring regarding the much-improved performance of ESMF and XIOS with respect to SCRIP.

9. Conclusions and perspectives

The objective of the work reported here is to offer other regridding possibilities than the SCRIP library in the OASIS3-MCT coupler. Therefore, we have investigated the regridding behaviour of different software used in Earth System Modelling, i.e. ATLAS, MOAB-Tempest Remap, YAC, ESMF, and XIOS.

A first preliminary analysis (Sect. 3) led us to conclude that ATLAS and MOAB-Tempest Remap are certainly appealing libraries with good long term perspectives regarding their development and support but their usage for regridding in OASIS3-MCT cannot be recommended at this point in time. ESMF and XIOS, on the contrary, are very interesting candidates in their current state. The same is true for YAC with a somewhat lower priority given the lack of official community support, even if its main developer is very dynamic and responsive.

To go further, we have developed a benchmark environment (Sections 4, 5 and 6) that was used to calculate the metrics proposed by the CANGA project for SCRIP, YAC, ESMF, and XIOS. Five regridding algorithms were evaluated for four different functions for seven pairs of grids used in real ocean or atmosphere models.

Results of the benchmark for YAC, ESMF, and XIOS are detailed in Sect. 7 and can be summarized as follows:

- nearest-neighbour

Nearest-neighbour is available in ESMF and YAC and produces almost exactly the same and very reasonable results, comparable to the SCRIP results (see Fig. 4). For all regridders, the analytical function used to define the field has a strong impact on the maximum of the misfit, which is directly linked to the field gradients (see Fig. 5).

- BILINEAR (or equivalent)

ESMF implements a standard bilinear scheme. This is not the case for YAC and so we used for YAC an inverse distance weighting of the vertices of the source polygon enclosing the target point + 2 nearest neighbours for target points falling outside any source polygon, which should be approximately equivalent. XIOS does not support any non-conservative algorithm.

The YAC algorithm is less accurate on average and leads to more smoothing (see Fig. 6). All results are reasonable, but for ESMF for the regridding of the gulfstream function for torc->sse7 and torc->bggd (see Fig. 7)., which shows one anomalous value near the coast (as for ESMF patch algorithm).

- 2nd O non-conservative

2nd O non-conservative regridding is implemented in ESMF with a patch algorithm and in YAC with spherical Bernstein-Bézier polynomials. ESMF gives slightly less accurate results and the patch averaging step smooths the resulting function and prevents over- or undershoots (see Fig. 8).

As for BILINEAR, all results are reasonable, but for ESMF for the regridding of the gulfstream function for torc->sse7 and torc->bggd, which shows one anomalous value near the coast (see Fig. 9 and Fig. 10). This issue still has to be investigated with ESMF developers.

YAC also shows some overshoots for icos-torc and icos-nogt for gulfstream functions, linked to higher, but a priori non anomalous values, in the centre of the gulfstream (see Fig. 11).

- CONSERV DESTAREA

1st O conservative remapping with DESTAREA normalisation is implemented in YAC, ESMF, and XIOS. The three regridders show very similar and good results (see Fig. 14) but for ESMF only if the nogt grid is described as unstructured for ESMF (see Fig. 13). This issue still has to be investigated with ESMF developers.

- CONSERV FRACAREA

1st O conservative remapping with FRACAREA normalisation is implemented in YAC, ESMF, and XIOS. The three regridders show very similar and good results for all functions (see Fig. 15 and Fig. 16). This regridding raises no specific issues for any regridder.

- CONS2ND FRACAREA

2nd O conservative remapping with FRACAREA normalisation is implemented in YAC, ESMF, and XIOS, all three using the same algorithm by Kritsikis et al (2017). All regridders show approximately the same behaviour with good global conservation (see Fig. 17 and 18).

One issue however is for ESMF when the source grid is the icosahedral one, icos, which shows for all functions a relatively high mean misfit, with an alternating positive and negative pattern in the misfit (see Fig. 19). This issue is under investigation with ESMF developers.

Another issue is for XIOS which shows a strong undershoot for the gulfstream function for torc-icos, with one clearly outstanding point near the coast for XIOS; ESMF also shows some negative misfit at this point but much smaller than XIOS; YAC does not show any important misfit at this point (see Fig. 21). This difference between the three regridders has to be investigated in more details.

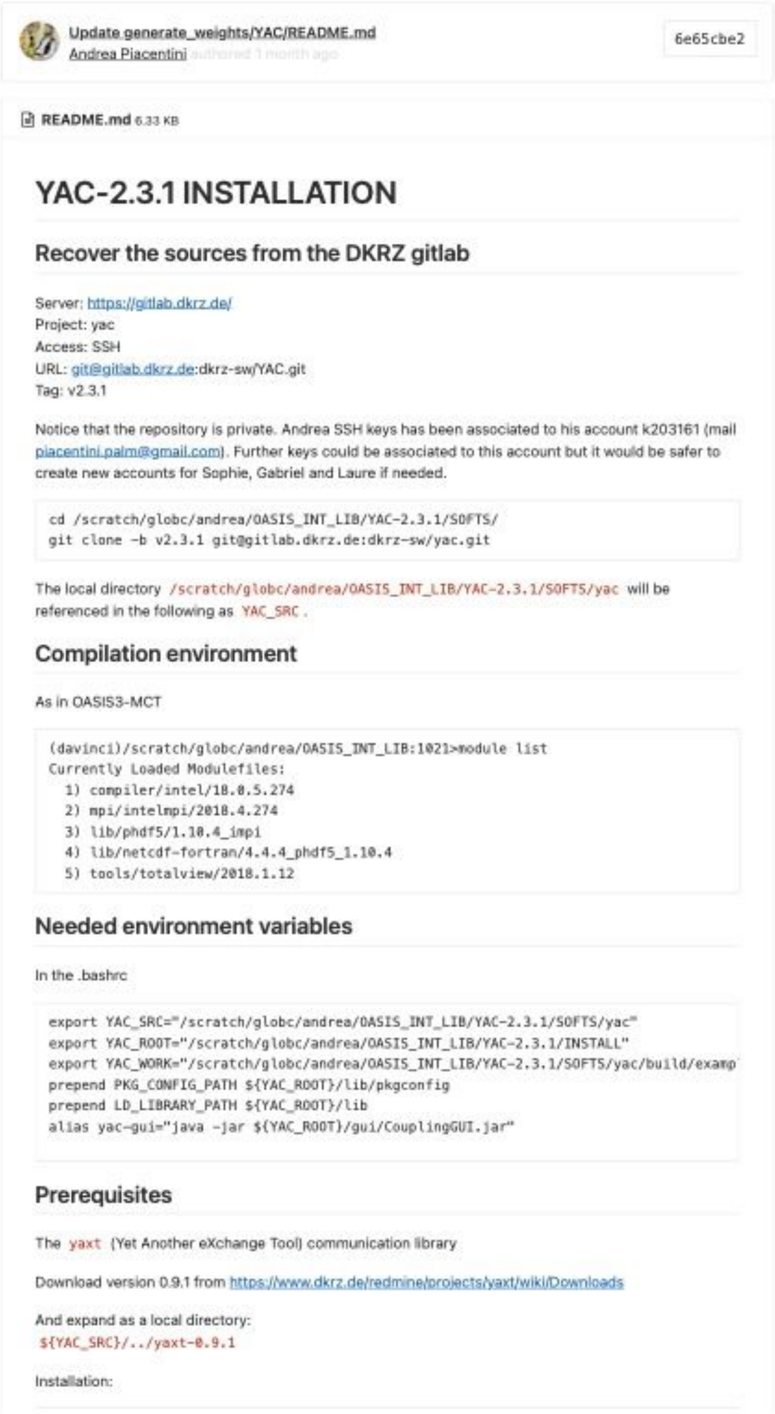
In conclusion, the benchmark results lead us to conclude that YAC, ESMF and XIOS can all be considered as high-quality regridding libraries, while some details still need to be fixed.

We then proceeded to a very first performance analysis of SCRIP, ESMF and XIOS which shows that both ESMF and XIOS show much better (O(100)) performance than the SCRIP. At low number of cores, XIOS is more performant than ESMF but shows unstable behaviour for more than ~200 cores. This first analysis is quite reassuring regarding the much-improved performance of ESMF and XIOS with respect to SCRIP. Of course, on the longer term, it will be crucial for the further evolution of the libraries to adapt to new architectures like GPUs and accelerators and we know that both ESMF and XIOS are starting to consider this important issue.

At this point, instead of choosing one new library among the ones that we evaluated and replacing the SCRIP with it (as was originally described in the IS-ENES3 DoW), we decided to keep the horizon more open, in agreement with our users. We think that it will be more beneficial to help our users to test and use those different libraries so that they get more experience and so that we get their feedback before investing more effort into interfacing one particular library into OASIS3-MCT. Given the benchmark results and our preliminary analysis, our objective now is therefore to provide, with OASIS3-MCT sources, a unified scripting environment for SCRIP, ESMF and XIOS so that users can test those three libraries on their own grid for their own regridding. Given the effort we spent getting familiar and designing an environment to use the libraries ourselves, we consider we are well placed to achieve this task.

Appendix 1 – YAC installation

Details about YAC installation and how to use it for calculating the benchmark regridding weights are provided in the README.md from the nitrox repository, which is reproduced here.



Update generate_weights/YAC/README.md
Andrea Piacentini authored 1 month ago

6e65cbe2

README.md 6.33 KB

YAC-2.3.1 INSTALLATION

Recover the sources from the DKRZ gitlab

Server: https://gitlab.dkrz.de/
Project: yac
Access: SSH
URL: git@gitlab.dkrz.de:dkrz-sw/YAC.git
Tag: v2.3.1

Notice that the repository is private. Andrea SSH keys has been associated to his account k203161 (mail piacentini.palm@gmail.com). Further keys could be associated to this account but it would be safer to create new accounts for Sophie, Gabriel and Laure if needed.

```
cd /scratch/globc/andrea/OASIS_INT_LIB/YAC-2.3.1/SOFTS/
git clone -b v2.3.1 git@gitlab.dkrz.de:dkrz-sw/yac.git
```

The local directory `/scratch/globc/andrea/OASIS_INT_LIB/YAC-2.3.1/SOFTS/yac` will be referenced in the following as `YAC_SRC`.

Compilation environment

As in OASIS3-MCT

```
(davinci)/scratch/globc/andrea/OASIS_INT_LIB:1021>module list
Currently Loaded Modulefiles:
  1) compiler/intel/18.0.5.274
  2) mpi/intelmpi/2018.4.274
  3) lib/phdf5/1.10.4_impi
  4) lib/netcdf-fortran/4.4.4_phdf5_1.10.4
  5) tools/totalview/2018.1.12
```

Needed environment variables

In the .bashrc

```
export YAC_SRC="/scratch/globc/andrea/OASIS_INT_LIB/YAC-2.3.1/SOFTS/yac"
export YAC_ROOT="/scratch/globc/andrea/OASIS_INT_LIB/YAC-2.3.1/INSTALL"
export YAC_WORK="/scratch/globc/andrea/OASIS_INT_LIB/YAC-2.3.1/SOFTS/yac/build/examp
prepend PKG_CONFIG_PATH ${YAC_ROOT}/lib/pkgconfig
prepend LD_LIBRARY_PATH ${YAC_ROOT}/lib
alias yac-gui="java -jar ${YAC_ROOT}/gui/CouplingGUI.jar"
```

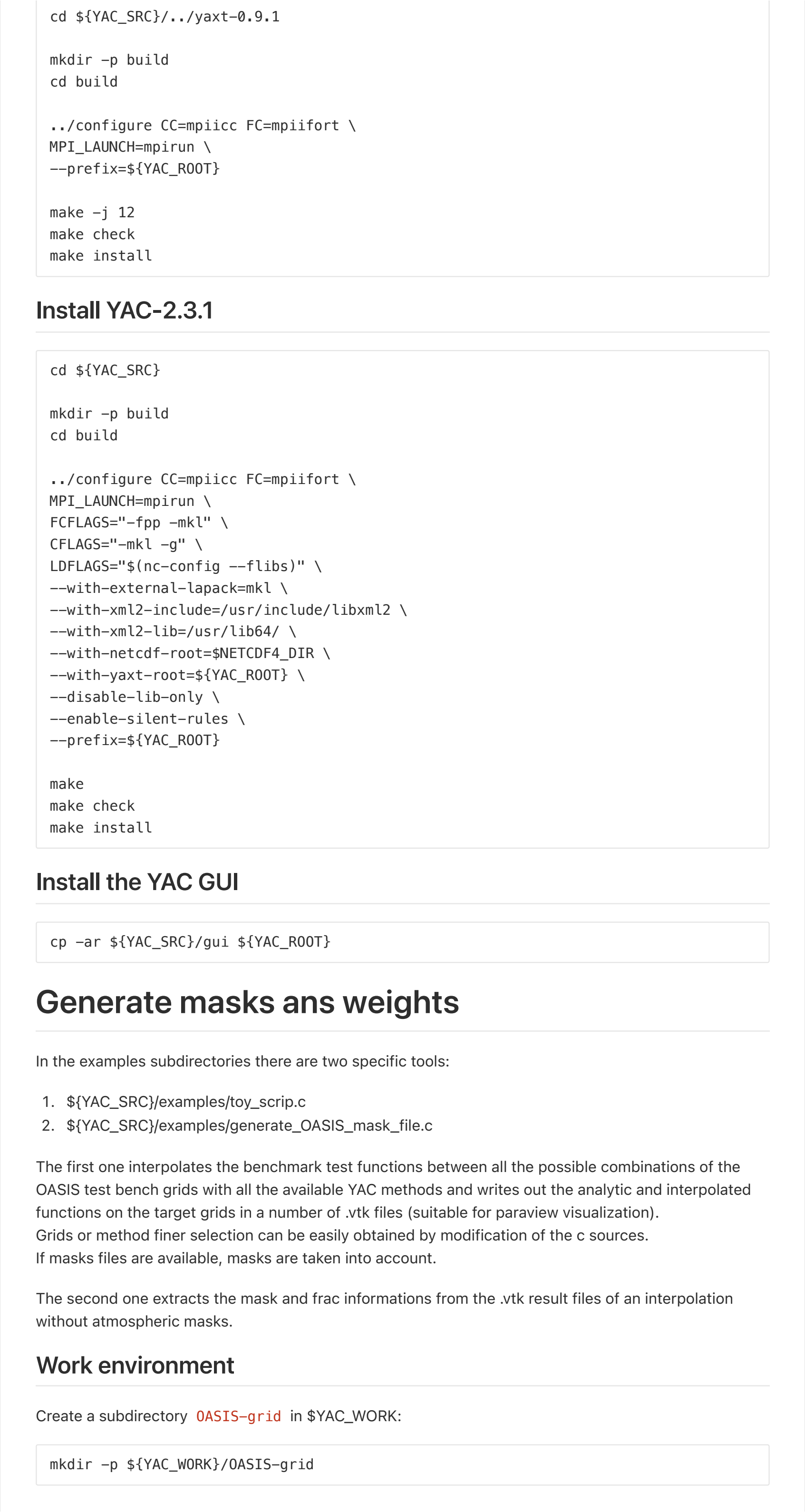
Prerequisites

The `yaxt` (Yet Another eXchange Tool) communication library

Download version 0.9.1 from https://www.dkrz.de/redmine/projects/yaxt/wiki/Downloads

And expand as a local directory:
`${YAC_SRC}/../yaxt-0.9.1`

Installation:



```
cd ${YAC_SRC}/../yaxt-0.9.1

mkdir -p build
cd build

../configure CC=mpiicc FC=mpiifort \
MPI_LAUNCH=mpirun \
--prefix=${YAC_ROOT}

make -j 12
make check
make install
```

Install YAC-2.3.1

```
cd ${YAC_SRC}

mkdir -p build
cd build

../configure CC=mpiicc FC=mpiifort \
MPI_LAUNCH=mpirun \
FCFLAGS="-fpp -mkl" \
CFLAGS="-mkl -g" \
LDFLAGS="$(nc-config --flibs)" \
--with-external-lapack=mkl \
--with-xml2-include=/usr/include/libxml2 \
--with-xml2-lib=/usr/lib64/ \
--with-netcdf-root=$NETCDF4_DIR \
--with-yaxt-root=${YAC_ROOT} \
--disable-lib-only \
--enable-silent-rules \
--prefix=${YAC_ROOT}

make
make check
make install
```

Install the YAC GUI

```
cp -ar ${YAC_SRC}/gui ${YAC_ROOT}
```

Generate masks ans weights

In the examples subdirectories there are two specific tools:

1. ${YAC_SRC}/examples/toy_scrip.c
2. ${YAC_SRC}/examples/generate_OASIS_mask_file.c

The first one interpolates the benchmark test functions between all the possible combinations of the OASIS test bench grids with all the available YAC methods and writes out the analytic and interpolated functions on the target grids in a number of .vtk files (suitable for paraview visualization).
Grids or method finer selection can be easily obtained by modification of the c sources.
If masks files are available, masks are taken into account.

The second one extracts the mask and frac informations from the .vtk result files of an interpolation without atmospheric masks.

Work environment

Create a subdirectory `OASIS-grid` in $YAC_WORK:

```
mkdir -p ${YAC_WORK}/OASIS-grid
```

edit toy_scrip.nc
Set (line 85): `char *experiment = "noicoh";`
then

```
cd $YAC_WORK
make

mpirun -np 5 ./toy_scrip.x
```

Generate the weights for the nogt icoh pair:

```
cd  ${YAC_SRC}/examples
```

edit toy_scrip.nc
Set (line 85): `char *experiment = "nogt_icoh";`
then

```
cd $YAC_WORK
make

mpirun -np 2 ../bin/toy_scrip.x
```

Generate the weights for the icos icoh pair:

cd ${YAC_SRC}/examples

edit toy_scrip.nc
Set (line 85): `char *experiment = "icos_icoh";` then

```
cd $YAC_WORK
make

mpirun -np 2 ./toy_scrip.x
```

Adapt the files to scrip format

The generated weight files have a different naming convention and lack the source dest and grid dimension sizes.

The dimension to be added are:

```
dimensions:
        src_grid_size = XXX ;
        dst_grid_size = YYY ;
```

The grid sizes are

```
griddim = {
"bggd" : 20592,
"icoh" : 2016012,
"icos" : 15212,
"nogt" : 106428,
"sse7" : 24572,
"torc" : 27118
}
```

Just run

```
cd $YAC_WORK
$(...)/oasis3-mct_other/generate-weights/YAC/YacWeightsToOasis.py
```

Appendix 2 – List of CSV files containing metrics values

Here is the list of the files included in the tar file Regridding_Benchmark_metrics.tar available on Zenodo at https://doi.org/10.5281/zenodo.5343166 . This tar file contains the regridding benchmark metrics calculated for all pairs of grids (see Sect.4.1) and all functions (see Sect. 4.2) for all regridders (see Sect.2) and for all algorithms (see Sect. 4.3), except when the regridder does not support the algorithm, such as e.g. BICUBC or BILINEAR for XIOS.

The name of the file is given as R_A_f.csv, where R is the regridder, A is the algorithm (with use of “DISTWGT_1” for nearest-neighbour), f is the function (here “classic” is equivalent to “sinusoid”).

For XIOS, there are therefore no files for DISTWGT_1, BILINEAR and BICUBIC as those algorithms are not supported in XIOS. For the 1st and 2nd O conservative remapping for the SCRIP, there are two files: one with (SCRIP-L) and one without (SCRIP) the Lambert Azimuthal projection. For ESMF, all results are provided for the version tagged ESMF_8_2_0_beta_snapshot_08 (ESMF-820bs08). For ESMF, for the 1st O CONSERV_DESTAREA, 1st O CONSERV_FRACAREA, 2nd O CONS2ND_FRACAREA algorithm, nogt grid was described as unstructured, as it solved issues with the North fold of the NEMO grid (see Sect. 7.4); in that case R=” ESMF-820bs08-U”. For ESMF, results for 1st O CONSERV_DESTAREA are also provided describing the nogt grid as with the SCRIP format for comparison; in that case R=” ESMF-820bs08”.

ESMF-820bs08-U_CONS2ND_FRACAREA_classic.csv
ESMF-820bs08-U_CONS2ND_FRACAREA_gulfstream.csv
ESMF-820bs08-U_CONS2ND_FRACAREA_harmonic.csv
ESMF-820bs08-U_CONS2ND_FRACAREA_vortex.csv
ESMF-820bs08-U_CONSERV_DESTAREA_classic.csv
ESMF-820bs08-U_CONSERV_DESTAREA_gulfstream.csv
ESMF-820bs08-U_CONSERV_DESTAREA_harmonic.csv
ESMF-820bs08-U_CONSERV_DESTAREA_vortex.csv
ESMF-820bs08-U_CONSERV_FRACAREA_classic.csv
ESMF-820bs08-U_CONSERV_FRACAREA_gulfstream.csv
ESMF-820bs08-U_CONSERV_FRACAREA_harmonic.csv
ESMF-820bs08-U_CONSERV_FRACAREA_vortex.csv
ESMF-820bs08_BICUBIC_classic.csv
ESMF-820bs08_BICUBIC_gulfstream.csv
ESMF-820bs08_BICUBIC_harmonic.csv
ESMF-820bs08_BICUBIC_vortex.csv
ESMF-820bs08_BILINEAR_classic.csv
ESMF-820bs08_BILINEAR_gulfstream.csv
ESMF-820bs08_BILINEAR_harmonic.csv
ESMF-820bs08_BILINEAR_vortex.csv
ESMF-820bs08_CONSERV_DESTAREA_classic.csv
ESMF-820bs08_CONSERV_DESTAREA_gulfstream.csv
ESMF-820bs08_CONSERV_DESTAREA_harmonic.csv
ESMF-820bs08_CONSERV_DESTAREA_vortex.csv
ESMF-820bs08_DISTWGT_1_classic.csv
ESMF-820bs08_DISTWGT_1_gulfstream.csv
ESMF-820bs08_DISTWGT_1_harmonic.csv
ESMF-820bs08_DISTWGT_1_vortex.csv
SCRIP-L_CONS2ND_FRACAREA_classic.csv
SCRIP-L_CONS2ND_FRACAREA_gulfstream.csv
SCRIP-L_CONS2ND_FRACAREA_harmonic.csv
SCRIP_CONS2ND_FRACAREA_harmonic.csv
SCRIP_CONS2ND_FRACAREA_vortex.csv
SCRIP_CONSERV_DESTAREA_classic.csv
SCRIP_CONSERV_DESTAREA_gulfstream.csv
SCRIP_CONSERV_DESTAREA_harmonic.csv
SCRIP_CONSERV_DESTAREA_vortex.csv
SCRIP_CONSERV_FRACAREA_classic.csv
SCRIP_CONSERV_FRACAREA_gulfstream.csv
SCRIP_CONSERV_FRACAREA_harmonic.csv
SCRIP_CONSERV_FRACAREA_vortex.csv
SCRIP_DISTWGT_1_classic.csv
SCRIP_DISTWGT_1_gulfstream.csv
SCRIP_DISTWGT_1_harmonic.csv
SCRIP_DISTWGT_1_vortex.csv
XIOS_CONS2ND_FRACAREA_classic.csv
XIOS_CONS2ND_FRACAREA_gulfstream.csv
XIOS_CONS2ND_FRACAREA_harmonic.csv
XIOS_CONS2ND_FRACAREA_vortex.csv
XIOS_CONSERV_DESTAREA_classic.csv
XIOS_CONSERV_DESTAREA_gulfstream.csv
XIOS_CONSERV_DESTAREA_harmonic.csv
XIOS_CONSERV_DESTAREA_vortex.csv
XIOS_CONSERV_FRACAREA_classic.csv
XIOS_CONSERV_FRACAREA_gulfstream.csv
XIOS_CONSERV_FRACAREA_harmonic.csv
XIOS_CONSERV_FRACAREA_vortex.csv
YAC_BICUBIC_classic.csv
YAC_BICUBIC_gulfstream.csv
YAC_BICUBIC_harmonic.csv
YAC_BICUBIC_vortex.csv
YAC_BILINEAR_classic.csv

SCRIP-L_CONS2ND_FRACAREA_vortex.csv
SCRIP-L_CONSERV_DESTAREA_classic.csv
SCRIP-L_CONSERV_DESTAREA_gulfstream.csv
SCRIP-L_CONSERV_DESTAREA_harmonic.csv
SCRIP-L_CONSERV_DESTAREA_vortex.csv
SCRIP-L_CONSERV_FRACAREA_classic.csv
SCRIP-L_CONSERV_FRACAREA_gulfstream.csv
SCRIP-L_CONSERV_FRACAREA_harmonic.csv
SCRIP-L_CONSERV_FRACAREA_vortex.csv
SCRIP_BICUBIC_classic.csv
SCRIP_BICUBIC_gulfstream.csv
SCRIP_BICUBIC_harmonic.csv
SCRIP_BICUBIC_vortex.csv
SCRIP_BILINEAR_classic.csv
SCRIP_BILINEAR_gulfstream.csv
SCRIP_BILINEAR_harmonic.csv
SCRIP_BILINEAR_vortex.csv
SCRIP_CONS2ND_FRACAREA_classic.csv
SCRIP_CONS2ND_FRACAREA_gulfstream.csv
YAC_BILINEAR_gulfstream.csv
YAC_BILINEAR_harmonic.csv
YAC_BILINEAR_vortex.csv
YAC_CONS2ND_FRACAREA_classic.csv
YAC_CONS2ND_FRACAREA_gulfstream.csv
YAC_CONS2ND_FRACAREA_harmonic.csv
YAC_CONS2ND_FRACAREA_vortex.csv
YAC_CONSERV_DESTAREA_classic.csv
YAC_CONSERV_DESTAREA_gulfstream.csv
YAC_CONSERV_DESTAREA_harmonic.csv
YAC_CONSERV_DESTAREA_vortex.csv
YAC_CONSERV_FRACAREA_classic.csv
YAC_CONSERV_FRACAREA_gulfstream.csv
YAC_CONSERV_FRACAREA_harmonic.csv
YAC_CONSERV_FRACAREA_vortex.csv
YAC_DISTWGT_1_classic.csv
YAC_DISTWGT_1_gulfstream.csv
YAC_DISTWGT_1_harmonic.csv
YAC_DISTWGT_1_vortex.csv

Appendix 3 – List of metrics plots

Here is the list of the files included in the tar file Regridding_Benchmark_metrics_plots.tar available on Zenodo at https://doi.org/10.5281/zenodo.5347696 .

This tar file contains the regridding benchmark metric plots (besides the target global conservation as only the source global conservation is illustrated) calculated for all pairs of grids (see Sect.4.1) and all functions (see Sect. 4.2) for all regridders (see Sect.2) and for all algorithms (see Sect. 4.3), except when the regridder does not support the algorithm, such as e.g. BICUBIC or BILINEAR for XIOS.

There is one plot for each algorithm for each function for all metrics for all pairs of grids and for all regridders. The name of the file is given as plot_remap_metrics_A_f.pdf, where A is the algorithm (with use of "DISTWGT_1" for nearest-neighbour) and f is the function (here "classic" is equivalent to "sinusoid")

For XIOS, there is no curve for nearest-neighbour, BILINEAR and BICUBIC as those algorithms are not supported in XIOS.

For ESMF, all plots are provided for the version tagged ESMF_8_2_0_beta_snapshot_08 with nogt declared as unstructured, as it solved issues with the North fold of the NEMO grid (see Sect. 7.4).

plot_remap_metrics_BICUBIC_classic.pdf
plot_remap_metrics_BICUBIC_gulfstream.pdf
plot_remap_metrics_BICUBIC_harmonic.pdf
plot_remap_metrics_BICUBIC_vortex.pdf
plot_remap_metrics_BILINEAR_classic.pdf
plot_remap_metrics_BILINEAR_gulfstream.pdf
plot_remap_metrics_BILINEAR_harmonic.pdf
plot_remap_metrics_BILINEAR_vortex.pdf
plot_remap_metrics_CONS2ND_FRACAREA_classic.pdf
plot_remap_metrics_CONS2ND_FRACAREA_gulfstream.pdf
plot_remap_metrics_CONS2ND_FRACAREA_harmonic.pdf
plot_remap_metrics_CONS2ND_FRACAREA_vortex.pdf
plot_remap_metrics_CONSERV_DESTAREA_classic.pdf
plot_remap_metrics_CONSERV_DESTAREA_gulfstream.pdf
plot_remap_metrics_CONSERV_DESTAREA_harmonic.pdf
plot_remap_metrics_CONSERV_DESTAREA_vortex.pdf
plot_remap_metrics_CONSERV_FRACAREA_classic.pdf
plot_remap_metrics_CONSERV_FRACAREA_classic_sansconservationMTR.pdf
plot_remap_metrics_CONSERV_FRACAREA_gulfstream.pdf
plot_remap_metrics_CONSERV_FRACAREA_harmonic.pdf
plot_remap_metrics_CONSERV_FRACAREA_vortex.pdf
plot_remap_metrics_DISTWGT_1_classic.pdf
plot_remap_metrics_DISTWGT_1_gulfstream.pdf
plot_remap_metrics_DISTWGT_1_harmonic.pdf
plot_remap_metrics_DISTWGT_1_vortex.pdf

REFERENCES

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

Collins, N., Theurich, G., DeLuca, C., Suarez, M., Trayanov, A., Balaji, V., Li, P., Yang, W., Hill, C. and da Silva, A. 2005: Design and Implementation of Components in the Earth System Modeling Framework, Int. J. High Perfor. Comput. Apps, 19 (3), 341–350.

Craig A., Valcke S., Coquart L. 2017: Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0, Geosci. Model Dev., 10, 3297-3308, https://doi.org/10.5194/gmd-10-3297-2017.

Deconinck, W., Bauer, P., Diamantakis, M., Hamrud, M., Kühnlein, C., Maciel, P., Mengaldo, G., Quintino, T., Raoult, B., Smolarkiewicz, P.K. and Wedi, N.P. 2017: Atlas: A library for numerical weather prediction and climate modelling, Computer Physics Communications, 220, 188–204. DOI: 10.1016/j.cpc.2017.07.006

Hanke, M., Redler, R., Holfeld, T. and Yastremsky, M. 2016: YAC 1.2.0: new aspects for coupling software in Earth system modelling, Geosci. Model Dev., 9, 2755–2769. https://doi.org/10.5194/gmd-9-2755-2016.

Hanke, M, and Redler, R. 2019: New features with YAC 1.5.0, DWD report 2019, https://doi.org/10.5676/DWD_pub/nwv/icon_003

Jones, P. 1999: Conservative remapping: First-and second-order conservative remapping, Mon. Weather Rev., 127, 2204–2210.

Jonville, G. and Valcke, S. 2019: Analysis of SCRIP conservative remapping in OASIS3-MCT – Part B, Technical Report TR/CMGC/19-155, Cerfacs, France.

Kritsikis, E. Aechtner, M. Meurdesoif, Y. and Dubos, T. 2017: Conservative interpolation between general spherical meshes, Geosci. Model Dev., 10, 425–431, https://doi.org/10.5194/gmd-10-425-2017

Mahadevan, V. S., Grindeanu, I., Jacob, R. and Sarich, J. 2020: Improving climate model coupling through a complete mesh representation: a case study with E3SM (v1) and MOAB (v5.x), Geosci. Model Dev., 13, 2355–2377. https://doi.org/10.5194/gmd-13-2355-2020

Piacentini, A., Maisonnave, E., Jonville, G., Coquart, L. and Valcke, S. 2018: A parallel SCRIP interpolation library for OASIS, Technical Report TR/CMGC/18-34, Cerfacs, France. https://cerfacs.fr/wp-content/uploads/2018/03/GLOBC-TR-PIANCENTINI-cmgc_18_34.pdf

Piacentini, A. 2020: Evaluation of Atlas 0.21 interpolation capability, Technical Report TR/CMGC/20-105, Cerfacs, France. https://cerfacs.fr/wp-content/uploads/2020/09/TR-atlas-0.21-analysis-CGCM-20-105.pdf

Valcke, S. and Piacentini, A. 2019: Analysis of SCRIP conservative remapping in OASIS3-MCT – Part A, Technical Report TR/CMGC/19-129, Cerfacs, France. https://cerfacs.fr/wp-content/uploads/2019/09/GLOBC-TR-Valcke-SCRIP_CONSERV_TRNORM_partA.pdf

Valcke, S., Craig, A. and Coquart, L. 2018: OASIS3-MCT User Guide, OASIS3-MCT4.0, Technical Report TR/CMGC/18-77, Cerfacs, France. https://cerfacs.fr/wp-content/uploads/2018/07/GLOBC-TR-oasis3mct_UserGuide4.0_30062018.pdf