

# Evaluation of preCICE in an ESM regridding benchmark

IDP project

Alex Hocks  

School of Computation, Information and Technology, Technical University of Munich

 alex.hocks@tum.de

April 29, 2021

**Abstract** — In the field of Earth System Models, meshes do not align or match usually, necessitating the use of couplers. We evaluate preCICE, a general coupling library, in this field by comparing it to commonly used and specialised couplers of this field. This is done by reproducing the benchmark by Valcke et al. [1]. The process of using preCICE is described and results are evaluated leading to the conclusion that in the current version of preCICE, it yields comparable results. However, when integral properties must be preserved, then, with the methods available, the accuracy is reduced proportional to the disparity of the surface area of the different meshes.

## 1 Introduction

Be it in the field of Meteorology for the next weather forecast or other topics involving a representation of the whole Earth, Earth System Models (ESMs) are used for this purpose. These models often comprise different meshes representing their respective aspect, such as atmosphere (ATM) and ocean (SEA) meshes. Given non-monolithic solvers, simulations carried out on ESMs employ different solvers for its respective models, which operate on not necessarily the same mesh. Thus, to transfer effects from boundary regions of one mesh to another, the solvers require coupling. Towards this extend, there exist different coupling libraries that perform the mapping of values from one mesh to another.

In the Regridding Benchmark by Valcke et al. [1] a selection of such libraries were evaluated on different pairs of ESM meshes. Data was generated by representative test functions, such as one for the Gulfstream. Afterwards, said libraries were used to map the data from one mesh to another. The results were analysed using a selection of metrics. The couplers selected in the paper were specialized for the usage with ESM meshes. The aim of this project is to evaluate preCICE[2] - a generic coupling library - in the benchmark and compare the results

to the original. We want to analyse preCICE in this context due to one of its main benefits being of generic nature. Because of that other solvers can be easily coupled, which extends the space of viable solver softwares. Examples are OpenFOAM, FEniCS, Nutils or CalculiX. The goal is to determine what the main differences are and if specialised couplers are necessary for these problems. This might be relevant for practitioners of this field as they may already have prior knowledge regarding preCICE.

## 2 Background

### 2.1 Mapping Theory

When coupling different simulations with each other, one key aspect is that they usually do not operate on the same models with their respective discretization schemes. Therefore, the underlying meshes are not identical and mesh points/nodes do not align perfectly. In order to transfer node or cell quantities, e.g., velocities or forces, from one mesh to another a translation method (mapping) is required. In the following we do not use physical quantities for our purposes of evaluating mappings, instead we obtain said quantities using a synthetic test function  $t(\mathbf{x})$ .

Assuming we have a source mesh  $S$  and a target mesh  $T$ , given by their points, with:

$$\mathbf{x}_i^S \text{ with } i \in \{1 \dots N\} \text{ and } \mathbf{x}_j^T \text{ with } j \in \{1 \dots M\} \quad (1)$$

Then, there usually is no point  $\mathbf{x}_j^T$  on the target mesh for which we can transfer the test function value  $t(\mathbf{x}_i^S) =: y_i^S$  directly to  $\mathbf{x}_j^T$  as  $\mathbf{x}_i^S \neq \mathbf{x}_j^T$ . Instead, we require a mapping  $m_{\mathbf{x}_1^S, \mathbf{x}_2^S, \dots}(\mathbf{x}_j^T, y_1^S, y_2^S, \dots) = y_j^T$ , that uses the source mesh points  $\mathbf{x}_1^S, \mathbf{x}_2^S, \dots$  and source values  $y_1^S, y_2^S, \dots$  to determine the value at  $\mathbf{x}_j^T$ . We denote  $m$  only as a function of the source values and target point, since the source points are fixed for a given mesh. As the mappings considered can be expressed linearly, we rewrite the data mapping in a more compact fashion.

$$M v_S = v_T \quad (2)$$

$v_S = (t(\mathbf{x}_1^S), \dots, t(\mathbf{x}_N^S))^T$  is the vector representation of node quantities on the source mesh,  $v_T \approx (t(\mathbf{x}_1^T), \dots, t(\mathbf{x}_M^T))^T$  the vector of the approximated quantities on the target mesh and  $M$  the mapping matrix, with  $m_j$  defined implicitly by the equation given by  $m_{\mathbf{x}_j^S, \mathbf{x}_j^T} (\mathbf{x}_j^T, y_1^S, y_2^S, \dots) = y_j^T$ .

Furthermore, mappings can be classified as consistent or conservative. Consistent mappings are a re-evaluation of the quantity function defined by the values  $v_S$  using interpolation. Thus, for a node  $\mathbf{x}_j^T = \mathbf{x}_i^S$  it holds:  $v_{S,i} = v_{T,j}$ . Values of  $v_T$  are at most as large as values in  $v_S$ . This results in a mapping matrix in which the sum over each row equals to one. Conservative mappings ensure that the sums over the quantity vectors are equal on the source and target mesh, i.e.,  $\sum_i v_{S,i} = \sum_j v_{T,j}$ . This leads to a mapping matrix in which the sums over columns equal to one.

## 2.2 preCICE Ecosystem

### Library

In its capacity as a generic partitioned multi-physics coupling library, preCICE is able to interact with a wide-range of solvers, be it a self-made implementation or an existing simulator. It is able to do so, by exposing a globally unified API. Integrations of solvers with preCICE are called adapters. Those call the respective API endpoints and handle for example transfer of data into preCICE buffers. For a selection of solvers, such as CalculiX, FEniCS, Nutils or OpenFOAM, adapters already exist. Consistent with preCICE nomenclature, we call involved solvers in a coupled simulation participants.

preCICE operates on a black-box principle and thus transfers data without information regarding the underlying physics to the required destinations. Towards this, it offers three fundamental aspects:

1. coupling schemes and acceleration
2. communication
3. mappings

Coupling schemes allow the selection between parallel and serial execution of participants as well as the direction of data-flow, i.e., whether participants send to each other or only uni-directional. Additionally, a choice between explicit and implicit coupling has to be made, in which the implicit version always requires a solution to a fix-point equation of the type

$H(x) = x$  up to a defined threshold. As this may require many iterations of all participants for a single global timestep, acceleration schemes in the form of Aitken under-relaxation or Quasi-Newton methods can be used.

Inter-participant communication is also handled by preCICE. First and foremost, participant codes are written independently of each other. By moving communication code to preCICE, it allows for higher flexibility. It is implemented by peer to peer communication, as a single server instance would be a significant bottleneck in high performance computing systems. Furthermore, preCICE determines based on the mesh nodes of each respective participant which data to send to which destination. Thus, minimizing communication.

In terms of mappings preCICE offers three different mapping methods: nearest-neighbour (nn), neareast-projection (np) and a family of radial basis function (rbf) methods. We only describe the rbf-pum-direct version here and refer to it as the rbf methods, as this was the type used for our purposes. Nearest-neighbour is a first-order method and maps nodes on the target mesh to the nearest neighbour on the source mesh, based on which values are assigned. Nearest-projection projects target nodes onto the surface of the source mesh. Then, values are linearly interpolated, allowing it to be up to a second order method. The selected rbf method partitions the domain into subregions, for each of which an interpolation polynomial is constructed on the source mesh. The interpolant is evaluated on the target mesh nodes. Afterwards, solutions for subregions are merged together. Each of the mapping can be consistent, conservative or scaled-consistent, with the first two being as defined in the previous section. For scaled-consistent first a consistent mapping is performed. Afterwards, the results are rescaled in a post-processing step to yield a conservative mapping. It can be defined as follows:

$$v_{tgt} = D M v_{src} \quad (3)$$

with  $v_{src}$  being the point data of each node and  $M$  the consistent mapping matrix.  $D$  is the rescaling matrix and can be obtained with:

$$D = s \cdot Id \quad (4)$$

$s$  is the aforementioned scaling factor and is defined as the balancing factor to equate the surface integrals on the source and target mesh:

$$s = \frac{I_s(v_{\text{src}})}{I_t(v_{\text{cons,tgt}})} \quad (5)$$

$I_k$  is the surface integral over the mesh  $k$  and uses the respective function values  $v$ .  $v_{\text{cons,tgt}}$  denotes the function values on the target mesh after only applying the consistent mapping matrix.

## ASTE

Additionally, preCICE comes with a multitude of tools for different purposes. The only one we used and introduce here is the artificial solver testing environment (ASTE)[3]. It is primarily a lightweight tool providing a reproducible environment and comprises 4 modes: **partition** and **join** perform domain decomposition, while **evaluate** accepts a mesh and a test function to attach the respective function values at all node positions. The mode we used was **run** and it can either replace a preCICE participant in a simulation by utilizing stored data. Alternatively, it calls the preCICE API to map one mesh to another.

## 2.3 Benchmark Setup

### 2.3.1 Meshes

As Earth is not a perfect sphere, we use the WGS84 model, resulting in an ellipsoid. ATM models are likely to be a shell around said ellipsoid, whilst SEA models may only contain a subsection of it. The resulting meshes are different discretization approaches of these models. As the purpose of the regridding benchmark is the evaluation of the mapping from one mesh to another, the considered meshes only contain the boundary area, i.e. the surface of the WGS84 ellipsoid.

ATM meshes span across the entire ellipsoid, since there exist no gaps in the atmosphere. SEA meshes, on the contrary, should only be valid in the regions of oceans. Therefore, certain areas are masked out. All meshes are discretized into cells of not necessarily uniform geometry and number of points. Meshes are provided in two forms: a) Each cell is defined via the location of its corner points b) Each cell is defined via its center point.

The SEA meshes are:

- **torc**: low resolution quadrilateral
- **nogt**: higher resolution quadrilateral

The ATM meshes are:

- **boggd**: only quadrilaterals
- **sse7**: mostly quadrilaterals, but has certain regions in which cells contains up to 7 points
- **icoh**: hexa- and pentagons with very high resolution
- **icos**: hexa- and pentagons with low resolution

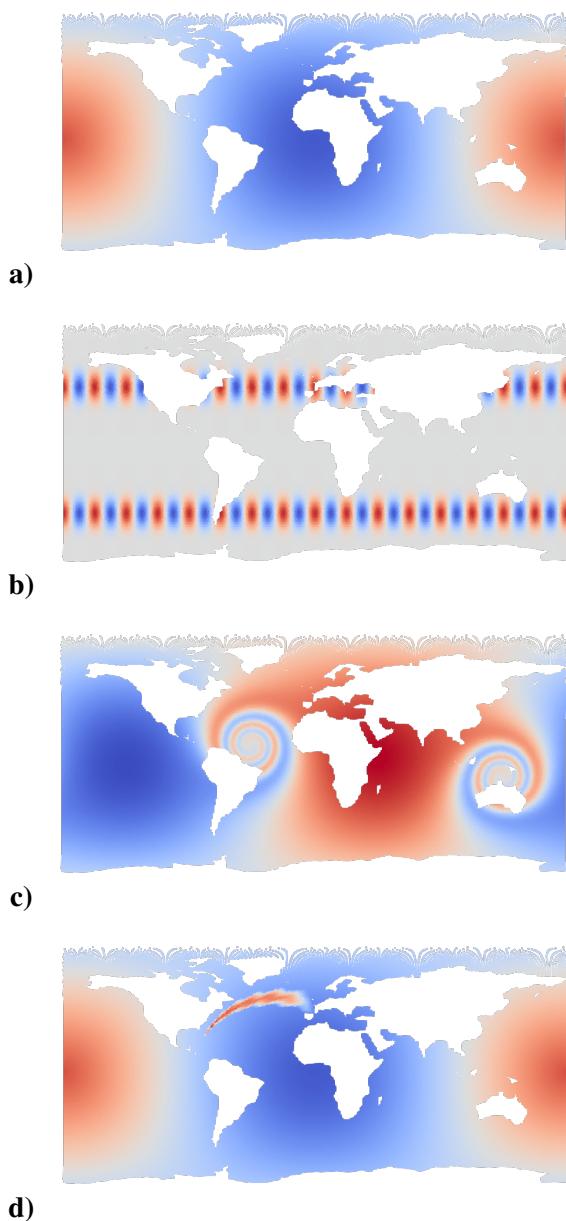
Meshes are expressed in two forms. a) 2D geodesic using lat/lon coordinates b) 3D cartesian using x/y/z coordinates. The original meshes are provided in NetCDF (NC) files, which is a self contained binary data format including all needed meta-information.[4] Those only contain a 2D geodesic representation. For our purposes we convert those into 3D cartesian representation.

### 2.3.2 Test Functions

To provide test data for the mapping 4 different test functions are evaluated on the meshes. These are:

- **Sinusoid**: Slowly varying standard sinusoid over the globe
- **Harmonic**: More rapidly varying function with 16 maximums and 16 minimums in northern and southern bands
- **Vortex**: Slowly varying function with two added vortices, one in the Atlantic and one over Indonesia
- **Gulfstream**: Slowly varying standard sinusoid with a mimicked Gulf Stream

The functions are evaluated using geodesic coordinates. See Figure 1 for an example on the nogt mesh. The formal definition is provided in the reference paper [1].



**Figure 1** Mesh noga evaluated on test functions. a) sinusoid  
b) harmonic c) vortex d) gulfstream

### 2.3.3 Accuracy Metrics

The following notation is from the reference paper [1] and reused for consistency purposes. We define  $\Psi^s$ ,  $\Psi^t$  as the test function evaluated on the source and target mesh.  $R\Psi^s$  is the test function on the source mesh mapped to the target mesh.  $I_s$ ,  $I_t$  are integrals approximations on the source and target meshes, which are computed by the sum of cell areas multiplied by the relative function value.  $M := |R\Psi^s - \Psi^t|/|\Psi^t|$  is the misfit.

As accuracy measures we consider the mean, max and rms of the misfit. Furthermore, we use

$L_{\min} := (\min \Psi^t - \min R\Psi^s)/\max |\Psi^t|$  and  $L_{\max} := (\max R\Psi^s - \max R\Psi^t)/\max |\Psi^t|$ . Source and target global conservation are  $|I_t(R\Psi^s) - I_s(\Psi^s)|/I_s(\Psi^s)$  and  $|I_t(R\Psi^s) - I_t(\Psi^t)|/I_t(\Psi^t)$ .

### 2.3.4 Mappings in Paper

The mapping libraries used in the paper were SCRIP [5], YAC [6], ESMF [7] and XIOS [8]. When possible, Valcke et al. performed the mappings in [1] for all libraries using nearest-neighbour as well as first and second order methods. Each had a conservative and non-conservative version. In Table 1 is an overview of the naming schemes used by the reference paper.

	conservative	non-conservative
nearest-neighbour	-	distwgt-1
1-st order	conserv-fracarea	bilinear
2-nd order	cons2nd-fracarea	bicubic

**Table 1** Naming scheme of mappings in reference paper

## 3 Process with preCICE

Recreating the benchmark using preCICE required several preprocessing steps due to the format of the provided data. The entire process can split be into 6 steps. These are explained in the following subsections. All subsequent transformations from 2D geodesic to 3D cartesian and vice versa are performed using the Proj library [9] bindings for Python. The used code, scripts and data can be found in our repository<sup>1</sup>.

Since we heavily rely on the VTK data format [10], a brief introduction is provided here. Data contained in VTK files may be unorganised in the form of data objects or have a certain underlying structure in the form of datasets. Our dataset of choice is the unstructured grid (vtkUnstructuredGrid), which represents a collection of 3-dimensional points and can have arbitrary topology. The later may be constructed as a combination of 0D (points), 1D (lines), 2D (triangles, quads,...) or 3D (tetrahedron,...) objects by specifying the respective indices of points. Each topological object can be referred to as a cell, to which one can also associate cell-data. Furthermore, one can attach point-data to points. Both cell- and point-data can be scalars, vectors or higher

<sup>1</sup><https://github.com/Snapex2409/ESM-NC-VTK-Tool/tree/main>

dimensional objects. As cells and points are indexed sequentially, associated data comes in form of data arrays, i.e., vtkPointData or vtkCellData respectively. Additionally, since multiple data arrays may be attached, a name should be specified. [10]

### 3.1 SEA Masks

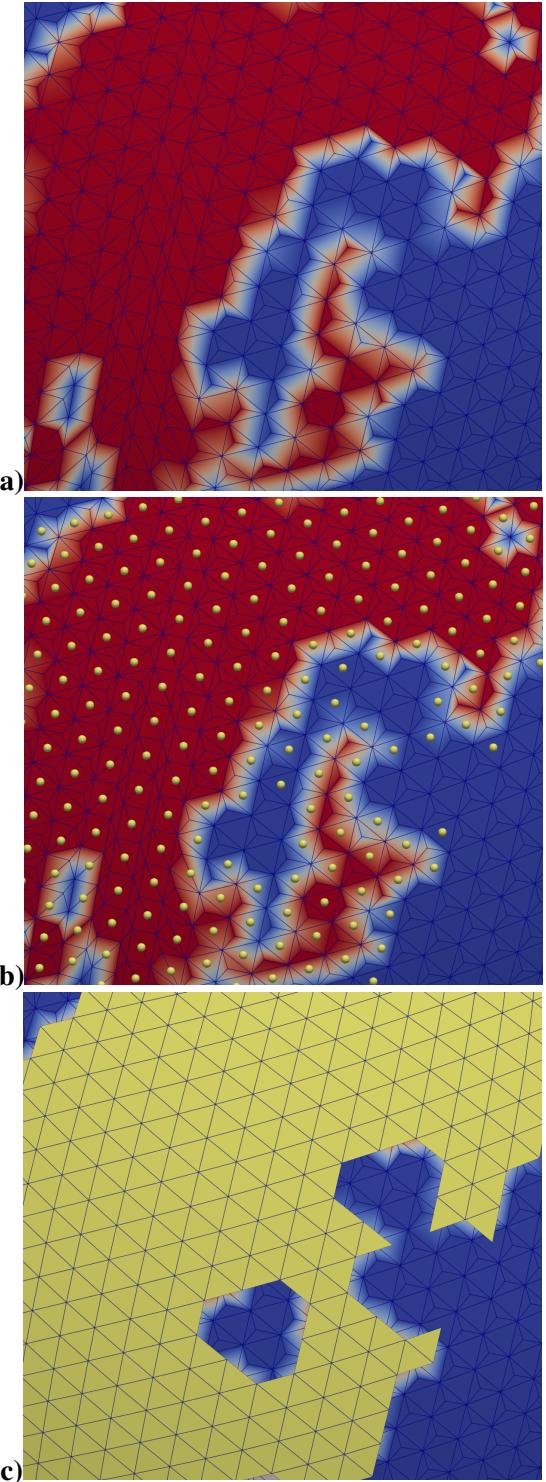
As only the SEA meshes have masks we need to transfer those to the ATM meshes in order to have a well posed mapping problem[1]. Towards this extend, the representation for the masks is based on the meshes defined by the cells using their corner points. Therefore, we extract as the first step all the meshes from the NC files into VTK format. Each VTK file contains one mesh using a vtkUnstructuredGrids, in which each cell is given via its corner points in 3D Cartesian coordinate system. For the SEA meshes, we attach one vtkPointData array containing ones for all points, that are contained in sea-cells, i.e., 1 represents water and 0 represents land. This information is extracted from the masks NC file. It should be noted, that at this stage all duplicated points as well as points that differ only spuriously are merged into a single entity. Furthermore, connectivity information is added as it is required by the next step. This is done by performing a triangulation of the surface defined by the points of each cell. A triangulation was necessary since there exist meshes that contain cells with more than 4 points, whereas preCICE only supports geometries up to quads. The triangulation was performed in the following manner:

For the points  $p_1, \dots, p_N$  of a given cell, one can assume due to proximity of the points that they are approximately laying on a common plane. We first compute the normalised basis vectors  $v_1$  and  $v_2$  using the first three points, then compute the normal  $n$  to the basis vectors. Next, the points are sorted counter-clockwise with respect to the normal vector, in order to create triangles with normals facing the same direction. Afterwards, we project the points onto the plane defined by  $v_1$  and  $v_2$  to obtain 2-dimensional coordinates for each point. As a final step a Delaunay triangulation is performed on the 2-dimensional representations.

### 3.2 ATM Masks

To obtain valid masks for the ATM meshes, each SEA mesh mask is mapped to every ATM mesh using a conservative nearest-projection mapping with preCICE.

The previously mentioned connectivity information is required due to the nearest-projection mapping. This results in two masks being created for every ATM mesh, one for each SEA mesh.



**Figure 2** Mesh conversion process on icos.

a) Corner based mesh containing mask information in each cell corner point b) Overlayed center points for each valid cell using yellow points c) Added connectivity information for center points

### 3.3 Mesh Extraction

The meshes used for the benchmarking purposes only use the center points of each cell. By using the original and mapped masks the fraction of sea content is evaluated for each cell through the average of the mask-values of its points. If the average is less than a specifiable threshold value (we used 1/1000), then the cell is considered to be land and is removed[1].

As all meshed now only contain center points, no connectivity information between points exists explicitly. In order to reconstruct it, we consider which cells contained common points. Based on this, potential edges are found. Afterwards only specific edges are selected, such that in the resulting mesh surface, there exist no triangles that overlap with each other. Furthermore, a vtkPointData array is attached containing the area of each cell. As a result, we obtain two center based meshes for each ATM mesh, as there are two masks available. For the SEA meshes, there is only a single output. An overview of the first three steps is highlighted in Figure 2.

### 3.4 Function Evaluation

As the next step all test functions are evaluated on all meshes based on the center points. The evaluation results are attached as another vtkPointData array.

### 3.5 Mapping

We perform all possible mappings. When we map from a SEA mesh  $a$  to an ATM mesh  $b$ , then we only use the ATM mesh that was created using the mask generated by  $a$ . The same applies for the opposite mapping direction as well. Mapping between two SEA meshes requires no further precaution. When mapping between ATM meshes, we use the meshes created using the nogt mask since nogt has the highest resolution.

We perform all mappings using three different mapping methods supported by preCICE: nearest neighbour, nearest projection and radial basis function. As the radial basis function mapping requires further parameters, we fine tuned those for stable results. All mappings were performed using the 3D cartesian representation of the meshes.

### 3.6 Metrics Evaluation

As the final step we evaluate all mapping results based on the prior mentioned metrics. For this, all mapping results need to be converted to 2D geodesic representation, because we need to compute analytical function evaluations on the target meshes. As output we create a JSON file with the mapping metrics as well as a further VTK file. The VTK file is based on the mapping result. However, we attach a further vtkPointData array containing the error metric, misfit, for each point.

## 4 Results of preCICE

### 4.1 Overview

We compare our results to the results from the original paper. For the papers non-conservative mappings we compare distwgt-1 to nearest neighbour, bilinear to nearest projection and bicubic to radial basis function. For conservative mappings nearest neighbour is omitted and we compare conserv-fracarea to nearest projection and cons2nd-fracarea to radial basis function. For a description of these names, refer to Table 1.

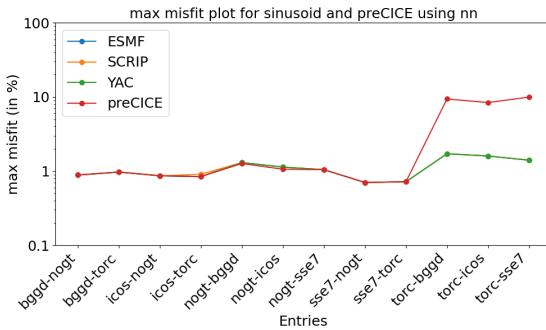
The preCICE documentation states, that nearest-neighbour has order 1 and nearest-projection order 2. Based on the definition of rbf, we infer that is able to express higher order behaviour. On the other hand, the reference paper assigns no order to distwgt-1, order 1 to bilinear, order 2 to bicubic, order 1 to conserv-fracarea and order 2 to cons2nd-fracarea. Based on this, our choice of compared methods may seem counter-intuitive. However, as the pairs of selected methods express similar behaviour in terms of order, we assume the disparity stems from a shift in index in the order calculation. Thus, we proceed with the stated comparison pairs.

In general, we can determine, that preCICE is able to achieve comparable results in terms of misfit, i.e., mapping to the target mesh and yielding low errors with respect to the analytical function of the target mesh. While the nearest-neighbour and nearest-projections mappings of preCICE match the reference results, rbf has up to approximately 2 orders of magnitude smaller misfit errors. On the other hand, preCICE is not able to achieve similar results regarding conservation of integral properties. Those errors lie consistently at about 4% for consistent mappings or 2% for scaled-consistent mappings with

preCICE. A more in detail discussion of the results will be held in the following sections.

## 4.2 Meshing Issues

One noteworthy difference overall is for the sinusoid test function with torc being the source mesh. This is highlighted in figure 3. The higher maximum error can



**Figure 3** Maximum misfit of nearest neighbour mapping with malformed torc mesh for preCICE

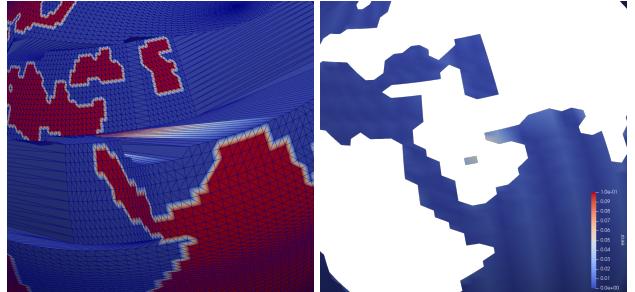
be located at a few cells around the south pole of the respective meshes. When analyzing those cells, one can identify that these cells should not exists, as they are part of Antarctica, thus, not part of the ocean. Upon checking the respective masks, it is appearant that the relevant cells were not masked out. Therefore, the issue lies within the mapping from the torc mesh mask to the target mesh. Since no mesh information exists in the torc mesh for a circular region  $A$  within the land region of the south pole, there are two small regions  $r_{1,2}$  in the boundary of  $A$  in which the mask values are interpreted as ocean cells. When performing the mapping from the torc mesh mask to the target mesh, this results in cells being not mask out in the regions  $r_{1,2}$  to the south pole, yielding the visible triangular areas, as displayed in figure 4. A similar issue arises in region



**Figure 4** Malformed torc mesh in antarctic region. Left: torc mesh mask Right: Error on bggd

of the mediterranean ocean. Here, malformed cells exist in the source (torc) mesh, connecting regions in

north Africa to Saudi-Arabia. This results once more in necessary cell not being masked out, see figure 5. Therefore, the maximum misfit is higher. After removing these cells, the created masks are correct and the mapping results with preCICE display no significant deviation from the reference mappings. These



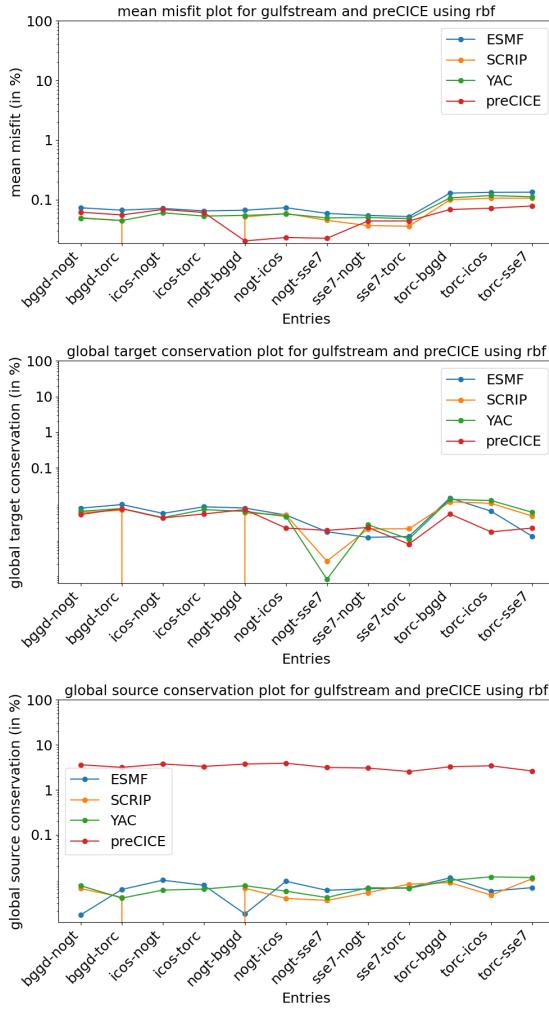
**Figure 5** Malformed torc mesh in northern Africa. Left: torc mesh mask Right: Error on bggd

issues are most likely not visible in the reference results, because error handling measures native to the respective refrence mappers were activated, such as `-ignore_degenerate`.

## 4.3 Non-Conservative Mappings

Comparing the non-conservative mapping, we observe that preCICE exhibits a similar behaviour in terms of misfit. To be more accurate, when using nn, there is no noticeable difference. The np mapping is in most cases a lower bound for the reference. rbf often has about 2 orders of magnitude smaller misfits. This means that the error to the analytical solution on the target mesh is overall lower with preCICE. A similar pattern can be observed for the global target conservation metrics, as its computation shares similarities with the misfit. On the other hand, when considering the source mesh conservation, preCICE is worse by 2 to 4 orders of magnitude depending the the scenario. See plots in figure 6. Furthermore, the source conservation for preCICE appears to be independent of mapping method and test function. This is also valid for the reference mappers, however to a smaller extent.

Considering max-misfits, then the results of preCICE follow the same trajectory as the reference's. For nearest-neighbour and nearest-projection mappings the max-misfits are almost identical. For radial basis function it is significantly less than the reference, with exception of the gulfstream cases. There rbf performs the same as the reference with respect to max-misfit. The behaviour of nearest-neighbour and



**Figure 6** Overall representative behaviour of consistent preCICE mapping compared to non-conservative reference.

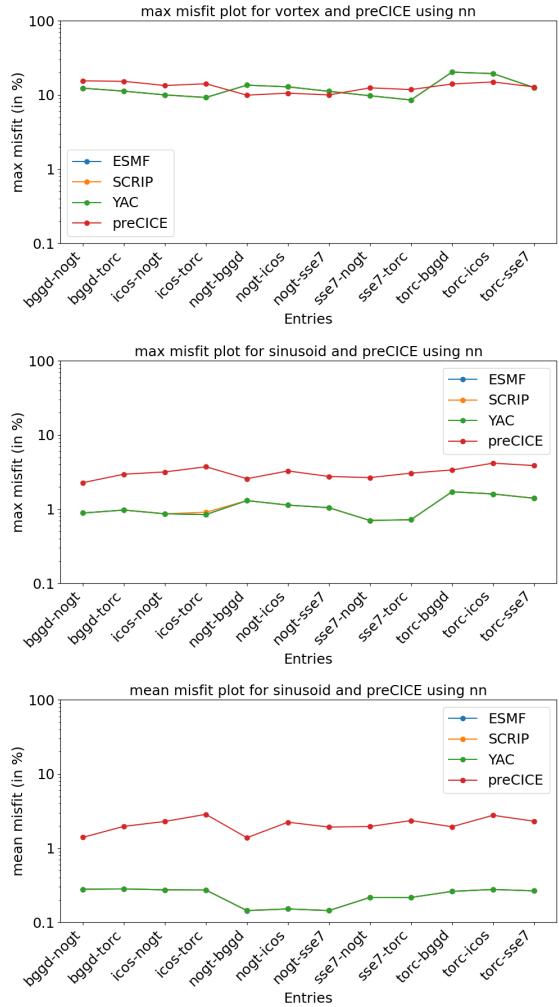
nearest-projection being similar to the reference and rbf being better carry over to mean- and rms-misfit. Regarding l-min and l-max, preCICE yields similar results as the reference without much deviation. For select cases, l-min and l-max are closer to 0 with preCICE. This means that the mapping results over- or underestimate the target analytical function less, i.e., the target function is represented with higher accuracy.

#### 4.4 Conservative Mappings

When performing a scaled-consistent mapping with preCICE, which is an approximation of the conservative mapping in the paper, the following observations are made. The mapped results have a higher error to the analytical function on the target mesh. Thus, the target mesh conservation as well as misfits are across the board worse than in the consistent mapping. The source mesh conservation improved proportionally,

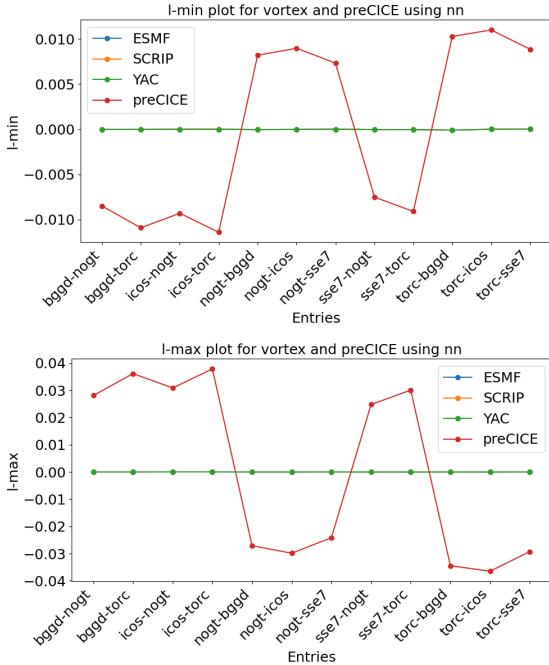
but still displays deviations from the results of the conservative and non-conservative mappings of the paper.

To determine the impact of the scaled-consistent mapping with preCICE we first compare the "conservative" results of preCICE with the consistent result of the reference. Checking the maximum misfits one can say, that our results are roughly in accordance with the reference. However, exceptions apply. In the nearest-neighbour and nearest-projection mappings for the sinusoid cases the max-misfit is higher by approximately factor 3. This behaviour extends to mean- and rms-misfits in which all mappings have higher misfit values for all cases, see figure 7. The l-min metric displays absolute values up to  $\approx 0.04$  while the reference is in most cases by multiple orders of magnitude closer to 0.



**Figure 7** Misfits for scaled-consistent mapping in preCICE and non-conservative reference.

Only for the nearest-neighbour and nearest-projection mappings with the harmonic test function we observe that the reference deviates from 0 and that the preCICE results follow those. For all other cases there is a pattern visible independent of mapping and test function. A strongly correlated pattern can be found in the l-max results for preCICE, see figure 8. Whenever, l-min is less than 0 then l-max is greater than 0 and vice versa. This can be interpreted as the



**Figure 8** l-min and l-max correlation for scaled-consistent preCICE mapping with non-conservative reference.

effects of the scaled-consistent mapping by preCICE. To verify this claim, we analyze the results in further detail. We observe pairs (-l-min, +l-max) and (+l-min, -l-max). Following the definition of these metrics:

- -l-min: minimum is larger
- +l-max: maximum is larger
- +l-min: minimum is smaller
- -l-max: maximum is smaller

Based on this we infer, that for (-l-min, +l-max) all values were scaled up by a factor  $s > 1$  and for (+l-min, -l-max) scaled down by a factor  $0 < s < 1$ . When checking the minimum and maximum values of the respective meshes, this holds. This scaling occurs due to the nature of the scaled-consistent mapping in preCICE, as described in 5.

MASK	torc	nogt
icos	4.096	4.016
bggd	4.090	4.001
sse7	4.061	3.983
torc	3.941	—
nogt	—	3.844

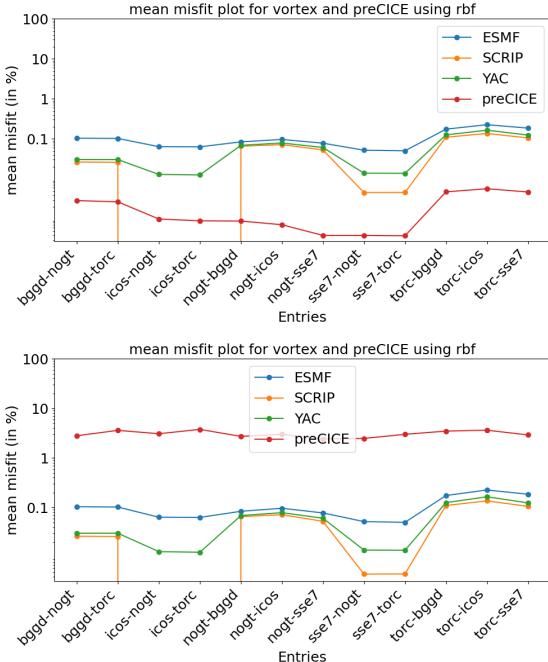
**Table 2** Filtered center mesh areas. All values are scaled by  $10^{14}$

Mapping	src area	tgt area	src/tgt	$s_{\text{real}}$
bggd-nogt	4.001	3.844	1.041	> 1
bggd-torc	4.090	3.941	1.038	> 1
icos-nogt	4.016	3.844	1.045	> 1
icos-torc	4.096	3.941	1.039	> 1
nogt-bggd	3.844	4.001	0.961	< 1
nogt-icos	3.844	4.016	0.957	< 1
nogt-sse7	3.844	3.983	0.965	< 1
sse7-nogt	3.983	3.844	1.036	> 1
sse7-torc	4.061	3.941	1.030	> 1
torc-bggd	3.941	4.090	0.964	< 1
torc-icos	3.941	4.096	0.962	< 1
torc-sse7	3.941	4.061	0.970	< 1

**Table 3** Overview of mapping pairs with their respective mesh areas and their quotient, being an approximation for the scaling factor  $s$ . The resulting fractions are compared to the observed scaling factors  $s_{\text{real}}$

This is also supported by data, see the tables about mesh areas and area fractions. The first table contains the surface areas of all relevant meshes. When applying this information to all considered mapping pairs one can see that whenever the fraction of source mesh area to target mesh area is greater than one, so is the inferred scaling factor based on the l-min/l-max relation.

Using the information from table 3 we can assume that the scaling will change the data by up to approximately  $\pm 4$  percent-points. When checking the change from the consistent to scaled-consistent mapping results of mean-misfits, then we can see that the data lies within this threshold, figure 9. A similar effect can be seen in the global target conservation, see figure 10. The deviation indicated here is also within the bounds of 4%. On the other hand, we observe a proportional reduction in the global source conservation. Thereby, improving the results of preCICE. However, those are still more than 2 orders of magnitude worse than the reference.

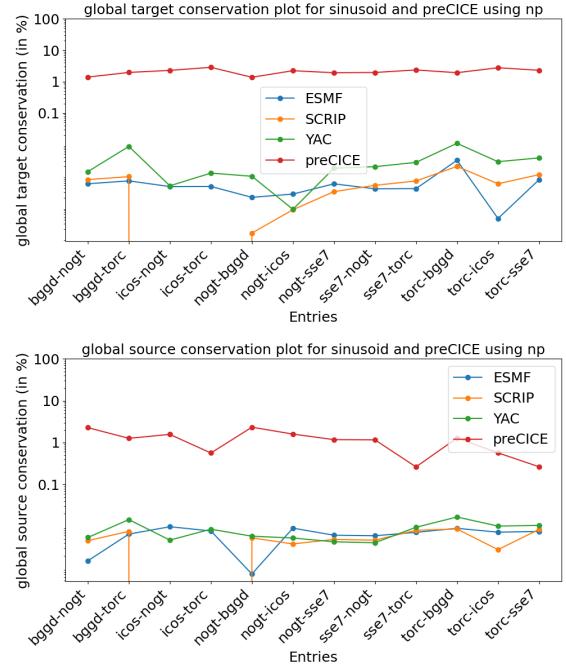


**Figure 9** Mean misfit change from consistent to scaled-consistent mapping. Scaled-consistent mapping within 4 percent-points of error. Top: consistent Bottom: scaled-consistent

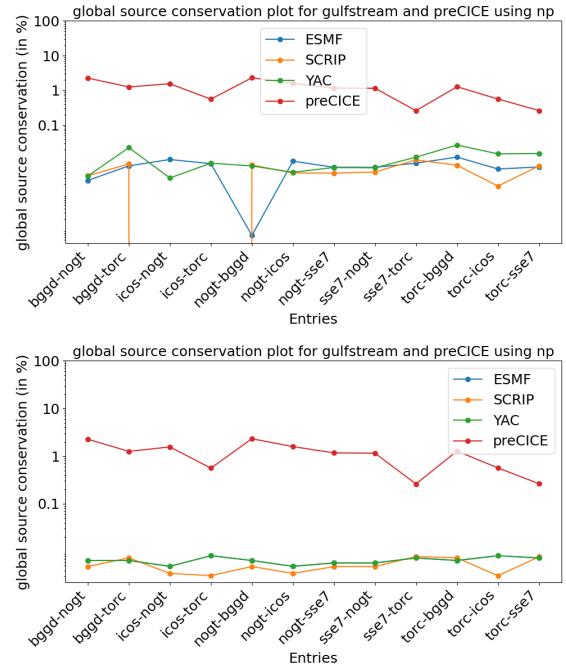
As the final step, we compare the results of the scaled-consistent mapping with the conservative mappings of the reference. In terms of the misfit and l-min/max metrics, no significant changes are visible relative to the comparison conducted with the non-conservative reference mappings. The mean-misfits of preCICE remain at a higher level and l-min/max display the same correlation. The only noticeable change is in the source conservation metric. Here, the reference results are now lower, such that preCICE is overall worse by approximately factor 2 to 4, see figure 11. This holds for all mapping types independently of test function.

## 5 Conclusion

Overall, we can determine, that preCICE was able to reproduce comparable results as the reference couplers in the regridding benchmark, in which deviations are within 4 percent points. With respect to consistent mappings it yielded very similar misfits for the nearest-neighbour as well as nearest-projection methods and achieved even lower misfits,  $\approx 2$  orders of magnitude, for the rbf mapping. Regarding the conservation of integral properties, we observed slightly higher errors, which are the cause for the 4 percent error margin. Here, the reference displayed



**Figure 10** Global source and target conservation of scaled-consistent mapping.



**Figure 11** Global source conservation of scaled-consistent mapping with preCICE and conservative reference.

errors close to 0 and about 4 percent for preCICE. Putting this into perspective, the errors are still in lower single digits. However, one must determine, whether such a deviation is acceptable for the current use case.

For the conservative mappings we can determine, that by using the scaled-consistent method in preCICE a reduction in the error metric for the preservation of integral quantities by about 2 or 3 percent points is achievable. This comes at an equal cost of accuracy with respect to misfits.

Therefore, we conclude that, preCICE is able to achieve highly accurate results in reproducing the test function on the target mesh, but does not perform equally well when preserving integral quantities on the source mesh. As an option to remediate that, one could implement the conservation techniques, that were used in the reference mappers, as a new feature in preCICE.

## 6 Outlook

In this work we only discussed the process of including preCICE in a ESM related mapping problem and evaluated it with respect to different accuracy metrics. As further steps, a detailed performance analysis in terms of runtime should be performed. Since preCICE is highly optimized for heavily parallel operations, it may be of value to evaluate different scenarios on distributed systems.

## 7 Reproducability Note

For detailed instructions see the documentation in our repository. All data is available in: <https://github.com/Snapex2409/ESM-NC-VTK-Tool/tree/main>. The initial readme file contains an overall overview, while the file reproducibility.md has more detailed instructions. All figures created by the benchmark are available in the *images* folder. The initial data is in the *data* folder.

## References

- [1] S. Valcke, A. Piacentini, and G. Jonville, “Benchmarking regridding libraries used in earth system modelling,” *Mathematical and Computational Applications*, vol. 27, no. 2, 2022. [Online]. Available: <https://www.mdpi.com/2297-8747/27/2/31>
- [2] G. Chourdakis, K. Davis, B. Rodenberg, M. Schulte, F. Simonis, B. Uekermann, G. Abrams, H. Bungartz, L. Cheung Yau, I. Desai, K. Eder, R. Hertrich, F. Lindner, A. Rusch, D. Sashko, D. Schneider, A. Totounferoush, D. Volland, P. Vollmer, and O. Koseomur, “preCICE v2: A sustainable and user-friendly coupling library [version 2; peer review: 2 approved],” *Open Research Europe*, vol. 2, no. 51, 2022. [Online]. Available: <https://doi.org/10.12688/openreseurope.14445.2>
- [3] D. Schneider, M. K. Yurt, F. Simonis, and B. Uekermann, “Aste: An artificial solver testing environment for partitioned coupling with precice,” *Journal of Open Source Software*, vol. 9, no. 103, p. 7127, 2024. [Online]. Available: <https://doi.org/10.21105/joss.07127>
- [4] U. Unidata, “Netcdf,” 2020. [Online]. Available: <https://doi.org/10.5065/D6H70CW6>
- [5] P. Jones, “A user’s guide for scrip: A spherical coordinate remapping and interpolation package, version 1.4,” 01 1998.
- [6] M. Hanke, R. Redler, T. Hofeld, and M. Yastremsky, “Yac 1.2.0: new aspects for coupling software in earth system modelling,” *Geoscientific Model Development*, vol. 9, no. 8, pp. 2755–2769, 2016. [Online]. Available: <https://gmd.copernicus.org/articles/9/2755/2016/>
- [7] N. Collins, G. Theurich, C. DeLuca, M. Suarez, A. Trayanov, V. Balaji, P. Li, W. Yang, C. Hill, and A. da Silva, “Design and implementation of components in the earth system modeling framework,” *The International Journal of High Performance Computing Applications*, vol. 19, no. 3, pp. 341–350, 2005. [Online]. Available: <https://doi.org/10.1177/1094342005056120>
- [8] Y. Meurdesoif, “The xml-io-server.” [Online]. Available: <https://forge.ipsl.jussieu.fr/ioserver>
- [9] PROJ contributors, “PROJ coordinate transformation software library,” 2025. [Online]. Available: <https://proj.org/>
- [10] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit (4th ed.)*. Kitware, 2006.