Work report – Implementation of true area normalization in OASIS3

CERFACS Working Notes WN-CMGC-12-48

Li Yan and Sophie Valcke

Part1: Why we introduce the true areas?

If there is no problem with the definition of the cell boundaries, i.e. the areas computed by the location of grid cell vertices can totally represent true areas, everything will be fine.

In general, in order to satisfy conservative regridding, we require:

$$\sum_i (F_{si} \cdot A_{si}) = \sum_j (F_{dj} \cdot A_{dj}) \qquad (1)$$

where A is the area of an individual grid cell, s refers to the source grid, d the destination grid, and the sums are over all grid cells, i and j.

In order to satisfy (1), we can use the following formula to assign the destination grid value of F:

$$F_{dj} = \sum_i (F_{si} \cdot f_{ji}) \qquad (2)$$

where f_{ji} is the areal fraction of destination cell j intersected by source cell i.[1] If (2) is satisfied, then (1) will be satisfied.

Why if we satisfy (2), we satisfy (1), it could be proved from the aspect of f_{ji}'s definition:

$$f_{ji} = \frac{A_{dj} \cap A_{si}}{A_{dj}} \qquad (3)$$

By summing over all destination cells and applying (2), (3) in the derivation steps, we have:

$$\begin{aligned}
\sum_j (F_{dj} \cdot A_{dj}) &= \sum_j ((\sum_i (F_{si} \cdot f_{ji})) \cdot A_{dj}) && \text{<---(2) applied} \\
&= \sum_j ((\sum_i F_{si} \cdot (\frac{A_{dj} \cap A_{si}}{A_{dj}})) \cdot A_{dj}) && \text{<---(3) applied} \\
&= \sum_j (\sum_i F_{si} \cdot (\frac{A_{dj} \cap A_{si}}{A_{dj}}) \cdot A_{dj}) \\
&= \sum_j (\sum_i F_{si} \cdot (A_{dj} \cap A_{si})) \\
&= \sum_i (\sum_j F_{si} \cdot (A_{dj} \cap A_{si})) && (4) \\
&= \sum_i (F_{si} \cdot \sum_j (A_{dj} \cap A_{si})) \\
&= \sum_i (F_{si} \cdot A_{si})
\end{aligned}$$

which prove that (1) is indeed satisfied.

1.1 True area normalization in SCRIP

The above is followed by the current SCRIP code, if the areas could be correctly reflected, there will be no problem. However, the areas computed by the location of grid cell vertices might differ the true areas[2]; under such conditions, (1) is not satisfied strictly. If (1) is not satisfied, this can lead to non-conservation of important properties[3]. So we need improve the normalization. In order to achieve this goal, previous areal fraction f_{ji} will be modified to be f_{ji_new} by multiplying a ratio r:

$$r=\frac{\frac{\text{True source area}}{\text{False source area}}}{\frac{\text{True destination area}}{\text{False destination area}}}=\frac{\frac{A_{\text{si\_True}}}{A_{\text{si\_False}}}}{\frac{A_{\text{dj\_True}}}{A_{\text{dj\_False}}}} \quad (5)$$

and

$$f_{\text{ji\_new}}=f_{ji}\cdot r \quad (6)$$

the new areal fraction f_{ji_new} involving both the source and destination cell areas.

So (2) could be improved to be:

$$F_{dj}=\sum_i (F_{si}\cdot f_{\text{ji\_new}}) \quad (7)$$

After this modification, (1) will be satisfied strictly whatever cell boundary curves or bow to which side. This could be illustrated in theory, see Appendix 1.

Applying formula (5), (6), (7), true area normalization has been implemented in OASIS3. It should be noted that in OASIS3, there were previously 3 normalization options, DESTAREA, FRACAREA, FRACNNEI.[4] After true area normalization, these 3 normalization are respectively improved to be DESTARTR, FRACARTR, FRACNNTR. DESTARTR/FRACARTR/FRACNNTR are respectively the same as DESTAREA/FRACAREA/FRACNNEI, except that they introduce true areas as explained in (5), (6), and (7). There are 6 options in all. 3 old options are still preserved.

1.2 Extra nearest-neighbour option

Note that the FRACNNEI option was changed so to ensure that unmasked destination cells that don't intersect any source cell (masked or unmasked) get nearest-neighbour value. This is about “tricky” cells. Tricky cell is defined as unmasked destination grid cell that intersects only masked source cells or doesn't intersect any source cell at all. Previously, the program picked up tricky cells by determining whether the corresponding sourcecells were all masked or not. So it missed tricky cells that don't intersect any source cell at all. After modification, the computing idea is reverse from before. It picks up tricky cells by determining unmasked destination cells with all source cells masked or without any link at all. So all tricky cells are now included.

For ORCA 2 degree grid (TORC) and Gaussian Reduced T42 (BT42), when BT42 is the destination cell, some tricky cells fall "outside" the ocean grid, i.e. they are located further south than the last line of the ocean grid. So they do not intersect masked source cells as they do not intersect any cells at all. This means that before modification these tricky cells did not get any value at all. And it means that it is better after modification because it will assign a nearest-neighbour value even for these tricky cells. Before and after modification, the number of tricky cells are 8 vs 25. So there are 17 tricky cells fall "outside" the ocean grid.

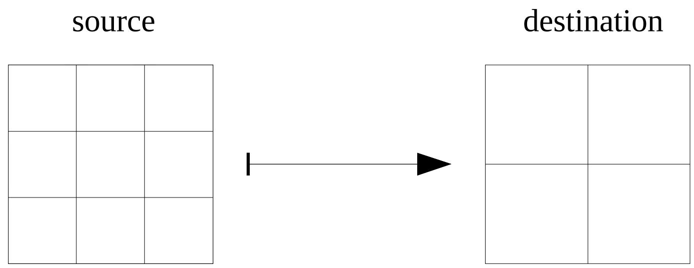
For ORCA 1 degree grid (TNE1) and Gaussian Reduced T127 (T127), there is no such problem.

Part 2: Simple grid results

The derivation to illustrate this issue may be weak, so we use simple grid to confirm the true area normalization's effectiveness.

2.1 construct the simple grid

Simple grids are constructed like this: ocean (here after ocn) grid is 3*3 cells, atmospheric (hereafter atm) grid is 2*2 cells. Ocn grid is as the source, atm the destination. They cover the same domain of the sphere (0~60N, 10E~70E). For ocn, every grid covers 20°*20°, atm 30°*30°. To be simple, all the cells are non-masked.



If the cell boundary is really along great circles, and parallels, on the sphere, the true area of each cell is respectively, see the following tables:

ocn grid cells : true areas (units:m^2)

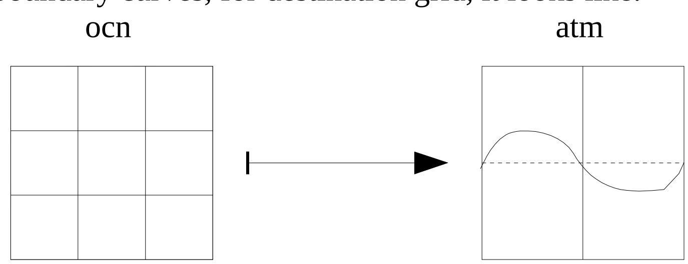
3162935	3162935	3162935
4261411	4261411	4261411
4845898	4845898	4845898

atm grid cells : true areas (units:m^2)

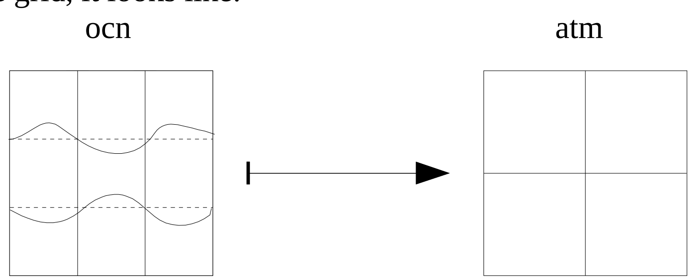
7779024	7779024
10626342	10626342

The sum area of all grid cells is 36810732. For ocn and atm, the sum area is identical.

If the cell boundary curves, for destination grid, it looks like:



for source grid, it looks like:



As a result of boundary curves, the true area will change, that means areas computed by location of vertices (hereafter call this computed areas as false areas) can not represent true areas. Because in the SCRIP code, the false areas are computed under the assumption that the vertices are connected in some simple way (e.g. along great circles, along parallels), it doesn't consider curving boundary conditions.

If the boundary bows slightly/heavily, the true area changes slightly/heavily correspondingly. For example, the atm grid cell boundary bows slightly, hence true area changes by 1% of lower left corner cell area. That's 100000 m^2, grossly 1% of 10726342. True areas will be then:

atm grid cells : true areas (units:m^2)

7679024 (=7779024 - 100000)	7879024 (=7779024 + 100000)
10726342 (=10626342 + 100000)	10526342 (=10626342 - 100000)

Of course, the total area of atm grid are kept unchanged: 36810732. Other area changes are similar(e.g. 10%, 20% and for ocn grid). Detailed area data, see Appendix 2.

2.2 integrated flux results

In order to estimate conservation, integrated flux before and after interpolation by OASIS3, with and without the true area correction, are calculated. The input integrated flux is $\sum_i (F_{si} \cdot A_{si})$, the output is $\sum_j (F_{dj} \cdot A_{dj})$. Values of the flux on each cell of input are given by the function:

$$F_{si} = 2 - \cos[\frac{1}{1.2} \cdot \arccos(\cos(lat_{si}) \cdot \cos(lon_{si}))] \quad (8).$$

where lat_{si} and lon_{si} are the latitude and the longitude of the grid point in the centre of the cell si.

Table 1: Integrated flux (change destination area)

	Normalization	Normalization option	Area change degree			
			0%	1%	10%	20%
Input	\	\	46551344			
Output	With true areas	FRACNNTR	46551344	46551344	46551344	46551344
	No true areas	FRACNNEI	46551344	46515648	46508452	46465560

Table 2: Integrated flux (change source area)

	Normalization	Normalization option	Area change degree			
			0%	1%	10%	20%
Input	\	\	46551344	46549256	46525240	46499136
Output	With true areas	FRACNNTR	46551344	46549256	46525240	46499136
	No true areas	FRACNNEI	46551344	46551344	46551344	46551344

The results show that with true area normalization, output integrated flux is always identical with input integrated flux, formula (1) is satisfied. Without such normalization, if area changes (boundary curves), formula (1) is not satisfied. In other words, true area normalization is needed to ensure conservation. Note that in this simple configuration where all the grid cells are non-masked, these 3 normalization options (DESTAREA, FRACAREA, FRACNNEI) would give the same results. So choose an option arbitrarily on behalf of “NO true area” option. Here we choose FRACNNEI vs its improved FRACNNTR.

Part3: Real grid results

Two set of grids are used. One set is Gaussian Reduced T42 (BT42) for atm and ORCA 2 degree (TORC) for ocn. The size of BT42 is 6232*1, TORC 182*149. The other set is Gaussian Reduced T127 (T127) for atm and ORCA 1 degree (TNE1) for ocn. The size of T127 is 24572*1, TNE1 362*292.

For real grid, we conduct interpolation from ocn to atm, also from atm to ocn.

Note that here the results shown for FRACNNEI option were obtained with the code that has been modified to ensure that non-masked destination cells that don't intersect any source cell (masked or unmasked) also get nearest-neighbour value (see 1.2 above).

Table 3: TORC---->BT42

Integrated flux : $\sum(F \cdot A)$: for non-masked points				
Input(Units:E+14)		Output(Units:E+14)		
6.664670	Without true area	DESTAREA	FRACAREA	FRACNNEI
		6.566796	6.792754	6.816145
	With true area	DESTARTR	FRACARTR	FRACNNTR
		6.566998	6.792935	6.816327

Table 4: BT42----> TORC

Integrated flux : $\sum(F \cdot A)$: for non-masked points				
Input(Units:E+14)		Output(Units:E+14)		
6.815671	Without true area	DESTAREA	FRACAREA	FRACNNEI
		6.566926	6.651047	6.664681
	With true area	DESTARTR	FRACARTR	FRACNNTR
		6.566698	6.651209	6.664842

Table 5: TNE1---->T127

Integrated flux : $\sum(F \cdot A)$: for non-masked points				
Input(Units:E+14)		Output(Units:E+14)		
6.750693	Without true area	DESTAREA	FRACAREA	FRACNNEI
		4.409293	6.713571	6.726474
	With true area	DESTARTR	FRACARTR	FRACNNTR
		4.856867	6.927643	6.940548

Table 6: T127----> TNE1

Integrated flux : $\sum(F \cdot A)$: for non-masked points				
Input(Units:E+14)		Output(Units:E+14)		
6.726511	Without true area	DESTAREA	FRACAREA	FRACNNEI
		6.761301	6.747849	6.750715
	With true area	DESTARTR	FRACARTR	FRACNNTR
		6.627720	6.627696	6.630562

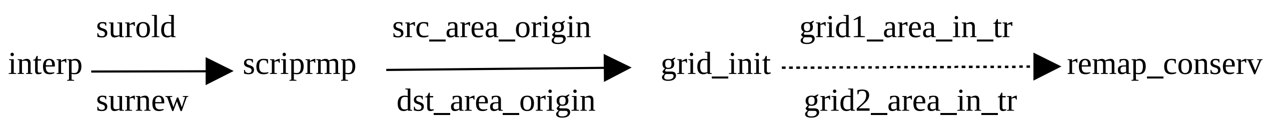
The results show that true area normalization is effective for all new options. However, we see here that the integrated fluxes before and after the interpolation are not strictly equal even with the new options. This is in fact not surprising as the true area normalization focuses only on the cell boundary curving issue and in the present case, the significant difference between the source and target integrated flux is due to the non-matching land-sea mask.

Note that under DESTAREA/DESTARTR option, when TNE1---->T127, the output and input show big difference, 4.409293/4.856867 vs 6.750693. This is because near the north pole, there are several cells have extreme negative value (about -3000 and -2000), that cause the much smaller integrated flux. These abnormal values are linked to specific SCRIP problems near the pole. However, for completeness, we show these results anyway.

Part4: Modifications in code

In the whole OASIS3 directory, there are five parts: main code, related libraries, utilities to compile OASIS3, examples to offer running environments and documentation[5]. Our modifications are done in main code and SCRIP library. Main idea is to transfer "true areas" values read in from the auxiliary NetCDF file areas.nc to remapping program in the SCRIP library. The flow chart is as follows:

Main stream:



The interp, scriprmp, grid_init and remap_conserv are subroutines in interp.F, scriprmp.F, grids.f and remap_conserv.F. The interp.F belongs to main code, other three programs belong to SCRIP library. Firstly, the true areas data transfer from interp to scriprmp, and then to grid_init, to save the true areas in save variables grid1_area_in_tr and grid2_area_in_tr. At last, the two save variable arrays will be used in the computation of norm_factor in remap_conserv.F. The dashed line arrow means that subroutine grid_init does not really call remap_conserv, grid_init just saves the true areas as the save variable in a module that is used after.

Besides the main stream modification above, some other modifications are done. One aspect is for all the subroutines which have called grid_init (scriprmp_vector, calc_remap_matrix, in scriprmp_vector.F, vector.F90), the other is for subroutines that treat the different normalization options (in scrip.F, remap_vars.F, remap_write.F).

In summary, after modification on FRACNNEI/FRACNNTR option, there are 2 routines modified in main code, 10 routines modified in SCRIP library. The algorithm to calculate tricky cells in fracnnei.f are totally changed. We rewrite it and split the fracnnei.f into 2 files: fracnnei_vmm.f and fracnnei.f. Also, the makefile is correspondingly changed.

Notes:

[1]Note that there is another way to assign the destination grid value of *F*, if areal fraction *f* represent the fraction of source grid cell *i* that contribute to destination cell *j (see "Preserving Global Integrals When Regridding", Karl E. Taylor, 2010)*. However, because in current SCRIP code this way is chosen, we follow SCRIP's form.

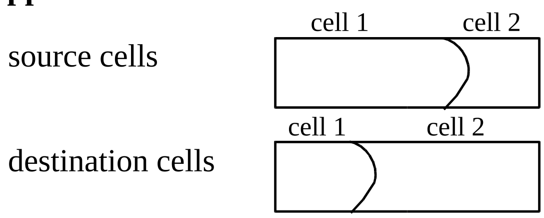
[2]Because when we compute the areas, we don't know how the vertices are connected to each other. We just construct the boundary in some simple way(e.g. The SCRIP assumes the boundary is linear in the longitude latitude space as in). This assumption may not reflect true boundaries.

[3]The more detailed saying is “This can lead to spurious time variations in a global mean quantity. It can also result in non-conservation of important properties when passing fluxes from one model component to another in some coupled climate models.” *(see “Preserving Global Integrals When Regridding”, Karl E. Taylor, 2010)*

[4]“FRACAREA: The sum of the non-masked source cell intersected areas is used to normalise each target cell field value: the flux is not locally conserved, but the flux value itself is reasonable.
DESTAREA: The total target cell area is used to normalise each target cell field value even if it only partly intersects non-masked source grid cells: local flux conservation is ensured, but unreasonable flux values may result.
FRACNNEI: as FRACAREA, except that at least the source nearest unmasked neighbour is used for unmasked target cells that intersect only masked source cells. ”
All these are quoted from *OASIS3 user Guide, p35(S. Valcke, 2010)*. Here the “unmasked target cells that intersect only masked source cells” are the “tricky” cells later mentioned.

[5]The whole OASIS3 directory structure, see *OASIS3 user Guide, p5, S. Valcke, 2010.*

Appendix 1: Illustration that new areal fraction fji_new guarantees the conservation



Consider two grids shown above, each with only two cells. Suppose the external grid cell corners are the same for the two grids, but suppose that the true cell boundary between the two cells bows toward the center of cell 2 on the source grid and on the destination grid. Let's suppose that the false cell boundaries are straight lines.
So the total source flux is

$$F_{s1}\times A_{\mathrm{s1\_true}}+F_{s2}\times A_{\mathrm{s2\_true}} \quad \text{(A1)}$$

the total destination flux is

$$F_{d1}\times A_{\mathrm{d1\_true}}+F_{d2}\times A_{\mathrm{d2\_true}} \quad \text{(A2)}$$

In addition, areal fraction f_{ji} could be defined:

$$f_{ji}=\frac{A_{\mathrm{dj\_False}}\cap A_{\mathrm{si\_False}}}{A_{\mathrm{dj\_False}}} \quad \text{(A3)}$$

here in (A3) all the areas are false areas, because for original areal fraction f_{ji}, all the areas are computed areas by vertices, no true areas are introduced.
And correspondingly new areal fraction f_{ji_new} could be written as:

$$f_{\mathrm{ji\_new}}=f_{ji}\times r=\frac{A_{\mathrm{dj\_False}}\cap A_{\mathrm{si\_False}}}{A_{\mathrm{dj\_False}}}\times\frac{\dfrac{A_{\mathrm{si\_True}}}{A_{\mathrm{si\_False}}}}{\dfrac{A_{\mathrm{dj\_True}}}{A_{\mathrm{dj\_False}}}}=\frac{A_{\mathrm{dj\_False}}\cap A_{\mathrm{si\_False}}}{A_{\mathrm{dj\_True}}}\times\frac{A_{\mathrm{si\_True}}}{A_{\mathrm{si\_False}}} \quad \text{(A4)}$$

Derive (A2), we have:

$$\begin{aligned}&F_{d1}\times A_{\mathrm{d1\_true}}+F_{d2}\times A_{\mathrm{d2\_true}}\\&=(F_{s1}\times f_{\mathrm{11\_new}}+F_{s2}\times f_{\mathrm{12\_new}})\times A_{\mathrm{d1\_True}}+(F_{s1}\times f_{\mathrm{21\_new}}+F_{s2}\times f_{\mathrm{22\_new}})\times A_{\mathrm{d2\_True}}\\&=F_{s1}\times f_{\mathrm{11\_new}}\times A_{\mathrm{d1\_True}}+0+F_{s1}\times f_{\mathrm{21\_new}}\times A_{\mathrm{d2\_True}}+F_{s2}\times f_{\mathrm{22\_new}}\times A_{\mathrm{d2\_True}}\end{aligned}$$

because $f_{21} = f_{21\_new} = 0$ as no parts of the source cell 2 intersect the destination cell 1.

applying (A4) here,

$$f_{\mathrm{11\_new}}\times A_{\mathrm{d1\_True}}+f_{\mathrm{21\_new}}\times A_{\mathrm{d2\_True}}=(A_{\mathrm{d1\_False}}\cap A_{\mathrm{s1\_False}})\times\frac{A_{\mathrm{s1\_True}}}{A_{\mathrm{s1\_False}}}+(A_{\mathrm{d2\_False}}\cap A_{\mathrm{s1\_False}})\times\frac{A_{\mathrm{s1\_True}}}{A_{\mathrm{s1\_False}}}$$

which can also be expressed, in the particular case illustrated above as:

$$=A_{\mathrm{d1\_False}}\times\frac{A_{\mathrm{s1\_True}}}{A_{\mathrm{s1\_False}}}+(A_{\mathrm{s1\_False}}-A_{\mathrm{d1\_False}})\times\frac{A_{\mathrm{s1\_True}}}{A_{\mathrm{s1\_False}}}=A_{\mathrm{s1\_False}}\times\frac{A_{\mathrm{s1\_True}}}{A_{\mathrm{s1\_False}}}=A_{\mathrm{s1\_True}}$$

similarly,

$$f_{\mathrm{22\_new}}\times A_{\mathrm{d2\_True}}=(A_{\mathrm{d2\_False}}\cap A_{\mathrm{s2\_False}})\times\frac{A_{\mathrm{s2\_True}}}{A_{\mathrm{s2\_False}}}=A_{\mathrm{s2\_False}}\times\frac{A_{\mathrm{s2\_True}}}{A_{\mathrm{s2\_False}}}=A_{\mathrm{s2\_True}}$$

so (A2) = (A1) using f_{ji_new} .

The result demonstrates that if we use new areal fraction f_{ji_new}, the conservation is ensured, at least in this special case.

Appendix 2: true area data after changed in simple grid

Atm area changed by 10%

6779024	8779024
11626342	9626342

Atm area changed by 20%

5779024	9779024
12626342	8626342

Ocn area changed by 1%

3122935	3202935	3122935
4341411	4181411	4341411
4805898	4885898	4805898

Ocn area changed by 10%

2662935	3662935	2662935
5261411	3261411	5261411
4345898	5345898	4345898

Ocn area changed by 20%

2162935	4162935	2162935
6261411	2261411	6261411
3845898	5845898	3845898