interpweight: Re-organization of ORCHIDEE's interpolation

L. Fita

[1] *Laboratoire de Météorologie Dynamique (LMD), CNRS, UPMC, Jussieu, Paris, Francia*

ORCHIDEE meeting – Palaiseau, 13 October 2016

Contact: **lluis.fita@lmd.jussieu.fr**

Overview: HEAT project

- HEAT project aims to develop a new Global Climate Model (GCM) using different components:
  - New atmospheric dynamical core called DYNAMICO
  - LMD atmospheric physics from LMDZ GCM
  - IPSL land model: ORCHIDEE
  - ocean model: NEMO
  - IPSL atmospheric chemistry: INCA
  - ...

Overview: HEAT project

- HEAT project aims to develop a new Global Climate Model (GCM) using different components:
  - New atmospheric dynamical core called DYNAMICO
  - LMD atmospheric physics from LMDZ GCM
  - IPSL land model: ORCHIDEE
  - ocean model: NEMO
  - IPSL atmospheric chemistry: INCA
  - ...
- DYNAMICO uses an unstructured grid and XIOS libraries for I/O

Overview: HEAT project

- HEAT project aims to develop a new Global Climate Model (GCM) using different components:
  - New atmospheric dynamical core called DYNAMICO
  - LMD atmospheric physics from LMDZ GCM
  - IPSL land model: ORCHIDEE
  - ocean model: NEMO
  - IPSL atmospheric chemistry: INCA
  - ...
- DYNAMICO uses an unstructured grid and XIOS libraries for I/O
- All the components have to be prepared to be able to work on an unstructured grid (or not with OASIS) and with XIOS libraries
  - Components' code have to use XIOS
  - Input and initialization files have to be understandable by XIOS

ORCHIDEE to DYNAMICO

- Due to efficiency constrains ORCHIDEE will be directly coupled to DYNAMICO

ORCHIDEE to DYNAMICO

- Due to efficiency constrains ORCHIDEE will be directly coupled to DYNAMICO
- Current trunk version of ORCHIDEE uses XIOS for the output

ORCHIDEE to DYNAMICO

- Due to efficiency constrains ORCHIDEE will be directly coupled to DYNAMICO
- Current trunk version of ORCHIDEE uses XIOS for the output
- ORCHIDEE should be prepared to use XIOS also for the input data

ORCHIDEE to DYNAMICO

- Due to efficiency constrains ORCHIDEE will be directly coupled to DYNAMICO
- Current trunk version of ORCHIDEE uses XIOS for the output
- ORCHIDEE should be prepared to use XIOS also for the input data
- Planned steps:
  1. Prepare ORCHIDEE to use XIOS variables-like outcome:
     Do not add any new feature. JUST re-organization & standardization of the existing code
  2. Prepare ORCHIDEE forcing files for XIOS

ORCHIDEE to DYNAMICO

- Due to efficiency constrains ORCHIDEE will be directly coupled to DYNAMICO
- Current trunk version of ORCHIDEE uses XIOS for the output
- ORCHIDEE should be prepared to use XIOS also for the input data
- Planned steps:
  1. Prepare ORCHIDEE to use XIOS variables-like outcome:
     Do not add any new feature. JUST re-organization & standardization of the existing code
  2. Prepare ORCHIDEE forcing files for XIOS
- Work done in a new branch:
  `svn://forge.ipsl.jussieu.fr/orchidee/branches/ORCHIDEE-DYNAMICO/ORCHIDEE`

ORCHIDEE to DYNAMICO

- Due to efficiency constrains ORCHIDEE will be directly coupled to DYNAMICO
- Current trunk version of ORCHIDEE uses XIOS for the output
- ORCHIDEE should be prepared to use XIOS also for the input data
- Planned steps:
  1. Prepare ORCHIDEE to use XIOS variables-like outcome:
     Do not add any new feature. JUST re-organization & standardization of the existing code
  2. Prepare ORCHIDEE forcing files for XIOS
- Work done in a new branch:
  `svn://forge.ipsl.jussieu.fr/orchidee/branches/ORCHIDEE-DYNAMICO/ORCHIDEE`
- More information at:
  `https://forge.ipsl.jussieu.fr/orchidee/wiki/DevelopmentActivities/ORCHIDI`

interpweight: Main objectives

1. Prepare ORCHIDEE’s interpolation code for XIOS

interpweight: Main objectives

1. Prepare ORCHIDEE’s interpolation code for XIOS
2. Generalize and flexibilize interpolation

interpweight: Main objectives

1. Prepare ORCHIDEE’s interpolation code for XIOS
2. Generalize and flexibilize interpolation
3. Without introducing any new piece of code, just re-organize it

interpweight: Main objectives

1. Prepare ORCHIDEE's interpolation code for XIOS
2. Generalize and flexibilize interpolation
3. Without introducing any new piece of code, just re-organize it
4. How works current interpolation
   - Uses `'aggregate_p'`
   - Has a specific/particular interpolation for each file
   - Different ranks 2D, 3D, 4D and kind of variables: unique value per grid point (`carteveg5km.nc`), multiple fractions per grid point (`PFTmap.nc`), continuous fields (`reftemp.nc`)
   - Perform mainly three steps (file depending):
     (a) Modification of values from file
     (b) Creation of land/sea mask using values from file
     (c) Interpolate values

interpweight: Main objectives

1. Prepare ORCHIDEE’s interpolation code for XIOS
2. Generalize and flexibilize interpolation
3. Without introducing any new piece of code, just re-organize it
4. General aim of ’interpweight’:
   - Join different ‘methods’ of each step in all purpose routines
   - Get some key information (range of values, dimensions, ...) directly from data
   - Prepare ORCHIDEE’s subroutines to use same output as the one provided by XIOS
   - Completely disappear once XIOS is introduced

interpweight: steps' subroutines

1. Modification of initial values: *Removing that wrong values from the file*
   `interpweight_modifying_input[1/2/3/4]D`: subroutines to modify 1D, 2D, 3D or 4D input data from file, using different methods: `noneg`

interpweight: steps' subroutines

1. Modification of initial values: *Removing that wrong values from the file*
   `interpweight_modifying_input[1/2/3/4]D`: subroutines to modify 1D, 2D, 3D or 4D input data from file, using different methods: `noneg`
2. Masking input: *Compute 'on fly' the values of the land/sea mask taking the values in the file*
   `interpweight_masking_input[1/2/3/4]D`: subroutines to compute the mask using data from input file using different methods: `nomask`, `mbelow`, `mabove`, `msumrange`, `var`

interpweight: steps' subroutines

1. Modification of initial values: *Removing that wrong values from the file*
   `interpweight_modifying_input[1/2/3/4]D`: subroutines to modify 1D, 2D, 3D or 4D input data from file, using different methods: `noneg`
2. Masking input: *Compute 'on fly' the values of the land/sea mask taking the values in the file*
   `interpweight_masking_input[1/2/3/4]D`: subroutines to compute the mask using data from input file using different methods: `nomask`, `mbelow`, `mabove`, `msumrange`, `var`
3. Area weights: Get coincident areas from input file to the target projection using `aggregate_p`
   No changes on it

interpweight: steps' subroutines (con't)

4. Interpolate: Use obtained areas and interpolate values at each grid point at the same format as it will be provided by XIOS
   - `interpweight_provide_fractions[1/2/3/4]D:` Perform interpolation for fraction/category data
   - `interpweight_provide_interpolation[2/4]D:` Perform interpolation for continuous data
   - `variableusetypes:` Variable to provide the values along the additional dimension to perform the interpolation. (e.g.: in case the 13 pfts are: 1,3,6,18,23,34,35,39,48,...)

interpweight: steps' subroutines (con't)

4. Interpolate: Use obtained areas and interpolate values at each grid point at the same format as it will be provided by XIOS
   - `interpweight_provide_fractions[1/2/3/4]D`: Perform interpolation for fraction/category data
   - `interpweight_provide_interpolation[2/4]D`: Perform interpolation for continuous data
   - `variableusetypes`: Variable to provide the values along the additional dimension to perform the interpolation. (e.g.: in case the 13 pfts are: 1,3,6,18,23,34,35,39,48,...)
5. A new variable is added which explains the 'availability' of data to perform the interpolation (0, 1). When it is negative it means that there was no data for that grid point

$$availability = \frac{\sum_{i=1}^{Npts} A_i^{source}}{Area\ target\ grid\ point} \tag{1}$$