

ESMF Python Interface JST call #2 summary

Date: December 6, 2012

Participants:

Eric Hutton - CSDMS

Karl Smith - NOAA

Lucy Smedstadt - NRL

Alex Pletzer - Tech-X

Arlindo daSilva - NASA

ESMF Core Team

Ryan O'Kuinghtons, Gerhard Theurich, Robert Oehmke

Minutes:

Grid coordinates and masks:

- overload grid constructors and/or have additional methods for `add_coords` and `add_items`?

- domain decomposition requires that the allocation and setting of coordinates happen in separate calls

- there is more functionality in the `add_coords` calls than just a stagger location which would make it a complicated looking constructor

- we could use `kwargs` to simplify the constructor interface

** ideal solution is to supply overloaded constructors for simple cases and also retain separate `add_coords()` and `add_mask()` interfaces for the more complicated cases (see below)

- mask could be handled by an `add_mask` call that both allocates the space and adds the data values

- this uses twice the memory as the three step process

minimum of 3 steps required to create grid coordinates and mask arrays:

- allocate

- get local bounds

- set

** replace static `ESMP_GridAddCoords()` with `coords = grid.add_coords(stagger, data)`

- at minimum sets the bounds and `coords` allocations as properties of the grid

- can optionally set data (double memory case)

- but user could also fill in the returned numpy array with coordinates at a later time

Field:

- the numpy array class should be used as a base class of the Field object
- base class could even be a masked array
- this is for the Field only, not the Grid
- this allows a LOT of functionality (histograms, slicing, etc.)

Python vs. Fortran/C allocation:

- We could explore passing Python allocated arrays through to the Fortran create from pointers interfaces of ESMF
- instead of the current allocate and copy paradigm, create with decomposition info

Question: Is it better to receive bounds on Python side and then allocate and array to pass to ESMF or to receive bounds and allocation from ESMF and only set the data on the Python side?

- Arlindo says that it doesn't matter, as long as the use is pythonic (e.g. need the ability to look at the shape to fill in the data)

Other issues:

- Arlindo suggests using the numpy c api to interface directly with Fortran where possible, rather than going through the c interface with ctypes
- we are currently doing everything through the C interface for consistency and because it helps to develop that interface as well as the Python side
- Alex asked about backward compatibility of present interface
- no, but the current tarballs will continue to be available

Actions:

- Arlindo will send an example of a Field derived class of Numpy Array
- Ryan will branch ESMP and start development of the new interface!