Yet Another Coupler – YAC

1 August 2020

Contact: Moritz Hanke (DKRZ)
René Redler (MPI-M)
YAC – Team

Development Team

Moritz Hanke (DKRZ)
René Redler (MPI-M)
Teresa Holfeld (MPI-M, student assistant)
Maxim Yastremsky (MPI-M, student assistant)

With contributions from

Thomas Jahns (DKRZ)
Uwe Schulzweida (MPI-M)
Hendrik Bockelmann (DKRZ)
Jörg Behrens (DKRZ)
Sergey Kosukhin (MPI-M)
YAC – General remarks

YAC – Tutorial, Version 1.5.5

Energy Momentum

Carbon

Water

ECHAM6

OASIS3

HAMOCC

MPIOM

JSBACH2

MPI-ESM1.0
YAC – General remarks

**Diagram:**

- **ECHAM6**
  - **Energy Momentum**
  - **Carbon**
  - **Water**
  - **Energy Momentum**

- **libOASIS3-MCT.a**

- **HAMOCC**

- **MPIOM**

- **JSBACH3**
  - **Water**

- **MPI-ESM1.1**
YAC – General remarks

YAC - Tutorial, Version 1.5.5

Energy
Momentum

ICON atmosphere

Energy
Momentum

Carbon

Water

River
Runoff

ICON-ESM

HAMOCC

ICON ocean

JSBACH4

libyac.a
YAC – General remarks

- search
- interpolation
- configuration
- communication
- data exchange
YAC – General remarks

A coupling software not only for ICON

Parallel search on (almost) arbitrary grids on the sphere
Parallel interpolation
Parallel data exchange

Library
BSD License
Programming Language C (~95%)\(^1\)
Fortran and C user API
Programming based on standards (C, MPI, XML, NetCDF)
Git repository
Redmine
Autotools
Valgrind testing
Unit tests (~90% of lines covered)
Fortran and C examples plus toy models
XML coupling configuration file with GUI support

\(^1\) generated using David A. Wheeler's 'SLOCCount'
YAC – Search

required

- Unique global IDs for cells, vertices, edges
- Geographical positions \((\lambda, \phi)\) of vertices and points
- Halo points/cells have to be marked
- Ranks of respective processes

provided

- Initial scalable computation of global mapping
- Final scalable parallel interpolation specific search and calculation of interpolation weights

features

- Support for circles of latitude/longitude and great circles
- Search and interpolation in Cartesian coordinates
- Convex & moderately concave polygons
- Support for masked cells and points
YAC – Interpolation

Available 2-dimensional (horizontal) interpolation methods

- 1\textsuperscript{st} – order conservative remapping (\textit{conserv})
- 2\textsuperscript{nd} – order conservative remapping (\textit{conserv})
- Hybrid cubic spherical Bernstein-Bézier patch interpolation (\textit{bernstein_bezier})
- Patch recovery - polynomial fit (\textit{patch_recovery})
- Smoothed Patch recovery - polynomial fit (\textit{smooth_patch_recovery})
- Distance-weighted N-nearest-neighbour (\textit{n-nearest_neighbour})
- N-nearest-neighbour average (\textit{n-nearest_neighbour})
- Gauss-weighted N-nearest-neighbour (\textit{n-nearest_neighbour})
- Radial Basis Functions (\textit{radial_basis_function})
- Source Point to Target Point Mapping (\textit{source_to_target_map})
- Simple cell average (\textit{average})
- Distance-weighted cell average (\textit{average})
- Fixed value (\textit{fixed})
- File input (\textit{user_file})
example


1\textsuperscript{st}-order conservative remapping

plus patch recovery

plus fixed value
YAC – Interpolation stack

Step 1: 1st-order conservative remapping
Step 2: ... + patch recovery
Step 3: \[ ... + \text{fixed value} \]
**Initialisation Phase**

- `yac_finit`
- `yac_fdef_comp`
- `yac_fdef_datetime`
- `yac_fget_localcomm`

**Grid Definition**

- `yac_fdef_subdomain`
- `yac_fdef_points`
- `yac_fdef_index_location`
- `yac_fdef_elements`
- `yac_fconnect_subdomains`
- `yac_fdef_mask`
- `yac_fdef_field`

**Search – End of Definition**

- `yac_fsearch`

**Data exchange**

- `yac_fget`
- `yac_fput`

**Termination**

- `yac_ffinalize`
YAC – Initialisation Phase

component initialisation

CALL yac_finit ( “coupling.xml”, “coupling.xsd” )
• will call MPI_INIT if not been called already

CALL yac_fdef_comp ( component_name, component_id )
• local operations for initialising of YAC-internal data structures
• needs to be called by all processes

CALL yac_fdef_datetime ( start_datetime = start_of_run_in_iso_format,
                          end_datetime  = end_of_run_in_iso_format )
• overwrites start and end date set in coupling.xml
• if required it has to be called before calling yac_fdef_field
• time management inside yac using mtime
YAC – Definition Phase

grid definition

CALL yac_fdef_subdomain ( component_id, grid_name, subdomain_id )
grid definition

CALL yac_fdef_elements ( subdomain_id,
                         nbr_of_horizontal_vertices,
                         nbr_of_horizontal_cells,
                         nbr_vertices_per_cell,
                         array_of_longitudes,
                         array_of_latitudes,
                         connectivity )

overloaded with respect to
   - data type for coordinate arrays
   - grid types
YAC – Definition Phase

grid definition

CALL yac_fdef_points ( subdomain_id,
nbr_of_horizontal_points,
CELL,
array_of_longitudes,
array_of_latitudes,
point_id )

overloaded with respect to
- data type for coordinate arrays
- grid types
grid definition

CALL yac_fdef_index_location ( subdomain_id, 
nbr_of_indices, 
CELL, 
array_of_global_indices, 
array_of_ranks )
YAC – Definition Phase

grid definition

CALL yac_fconnect_subdomains ( component_id, 
nbr_subdomain_ids, 
array_of_subdomain_ids, 
domain_id )
YAC – Definition Phase

mask definition

CALL yac_fdef_mask ( size_of_mask_array, mask_array, point_id, mask_id )

overloaded with respect to
- data type for mask array
YAC – Definition Phase

field definition

CALL yac_fdef_field ( field_name, component_id, domain_id, array_of_cell_point_ids, array_of_cell_mask_ids, nbr_point_set_per_subdomain, field_id )
YAC – Communication phase

search

CALL yac_fsearch ( nbr_of_components, array_of_component_ids, nbr_of_fields, array_of_field_ids, error_status )

- includes collective MPI operations
- needs to be called by all processes
- accesses the coupling configuration
- invokes the neighbourhood search
- does the communicator splitting

CALL yac_fget_localcomm ( local_mpi_communicator, component_id )
YAC – Exchange Phase

data exchange

as it is implemented in ICON

Beginning of timestep

End of timestep

dynamics & physics

acc

put

get

YAC – Tutorial, Version 1.5.5
YAC – Exchange Phase

data exchange

as it is implemented in ICON

Atmosphere

Ocean

from restart

… to next job

to restart

from restart

… to next job

to restart
YAC – Exchange Phase

data exchange

CALL yac_fput ( field_id,
               nbr_horizontal_points,
               collection_size,
               nbr_pointsets,
               nbr_subdomains,
               send_field,
               info,
               error_flag )

• to be called at every time step
• at the “source timestep” interval specified in the xml file
• accumulation/averaging done inside yac_fput
YAC – Exchange Phase

data exchange

! field_id(6) : Temperature

DO i_blk = 1, patch_horz%nblks_c
  nn = (i_blk-1)*nproma
  DO n = 1, nproma
    buffer(nn+n,1) = &
    ocean_state%p_prog(nold(1))%tracer(n,1,i_blk,1) + tmelt
  ENDDO
ENDDO

CALL yac_fput ( field_id(6), nbr_hor_points, 1, &
                1, 1, buffer(1:nbr_hor_points,1), &
                info, ierror )
YAC – Exchange Phase

data exchange

CALL yac_fget ( field_id,
nbr_horizontal_points,
collection_size,
nbr_pointsets,
nbr_subdomains,
recv_field,
info,
error_flag )

• to be called at every time step
• at the “target timestep” interval specified in the xml file
• check the returned info argument
YAC – Exchange Phase

data exchange

Return values for the info argument

```plaintext
enum, bind(c)

enumerator :: NONE = 0
enumerator :: COUPLING = 1
enumerator :: RESTART = 2
enumerator :: GET_FOR_RESTART = 3
enumerator :: PUT_FOR_RESTART = 4
enumerator :: GET_FOR_CHECKPOINT = 5
enumerator :: PUT_FOR_CHECKPOINT = 6
enumerator :: OUT_OF_BOUND = 7

end enum
```
CALL \texttt{yac\_fget} ( \texttt{field\_id(1)}, \texttt{nbr\_hor\_points}, 2, \&
& 1, 1, \texttt{buffer}(1:\texttt{nbr\_hor\_points},1:2), \&
& \texttt{info}, \texttt{ierror} )

IF (\texttt{info} > 0 \ . \& \texttt{info} < 7 ) THEN
DO i\_blk = 1, patch\_horz\%nblks\_c
  nn = (i\_blk-1)*nproma
  DO n = 1, nproma
    atmos\_fluxes\%stress\_xw(n,i\_blk) = \texttt{buffer}(nn+n,1)
    atmos\_fluxes\%stress\_x (n,i\_blk) = \texttt{buffer}(nn+n,2)
  ENDDO
ENDDO
CALL sync\_patch\_array ( \ldots , atmos\_fluxes\%stress\_xw(:, :) )
CALL sync\_patch\_array ( \ldots , atmos\_fluxes\%stress\_x (:,:) )
ENDIF
YAC – Termination Phase

termination of coupling

CALL yac_ffinalise()

- frees all internal data structures related to coupling
- MPI communicators may no longer be available
- will call MPI_FINALIZE
  - if MPI_INIT has been called by yac_finit
  - if MPI_FINALIZE has not already been called
Recommended calling sequence

CALL MPI_init ( ... )

CALL yac_finit ( ... )

CALL yac_finit_comp ( ... )

CALL yac_fsearch ( ... )

CALL yac_fget_local_comm ( ... )

CALL yac_fffinalise ( )

CALL MPI_finalize ( ... )
<?xml version="1.0" encoding="UTF-8"?>
<component
xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.component.xsd">
  <id>1</id>
  <name>atmo</name>
  <model>ICON</model>
  <simulated>atmosphere</simulated>
  <transient_grid_refs>
    <transient_grid_ref collection_size="2" grid_ref="1" id="1" transient_ref="1"/>
    <transient_grid_ref collection_size="2" grid_ref="1" id="2" transient_ref="2"/>
    <transient_grid_ref collection_size="3" grid_ref="1" id="3" transient_ref="3"/>
    <transient_grid_ref collection_size="4" grid_ref="1" id="4" transient_ref="4"/>
    <transient_grid_ref collection_size="4" grid_ref="1" id="5" transient_ref="5"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="6" transient_ref="6"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="7" transient_ref="7"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="8" transient_ref="8"/>
    <transient_grid_ref collection_size="5" grid_ref="1" id="9" transient_ref="9"/>
  </transient_grid_refs>
  <transients>
    <transient id="1" transient_standard_name="surface_downward_eastward_stress"/>
    <transient id="2" transient_standard_name="surface_downward_northward_stress"/>
    <transient id="3" transient_standard_name="surface_fresh_water_flux"/>
    <transient id="4" transient_standard_name="total_heat_flux"/>
    <transient id="5" transient_standard_name="atmosphere_sea_ice_bundle"/>
    <transient id="6" transient_standard_name="sea_surface_temperature"/>
    <transient id="7" transient_standard_name="eastward_sea_water_velocity"/>
    <transient id="8" transient_standard_name="northward_sea_water_velocity"/>
    <transient id="9" transient_standard_name="ocean_sea_ice_bundle"/>
  </transients>
  <grids>
    <grid id="1" alias_name="grid1"/>
  </grids>
</component>
YAC – Component XML configuration

<?xml version="1.0" encoding="UTF-8"?>
<component
 xmlns="http://www.w3schools.com"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.w3schools.component.xsd">
  <id>1</id>
  <name>atmo</name>
  <model>ICON</model>
  <simulated>atmosphere</simulated>
  <transient_grid_refs>
    <transient_grid_ref collection_size="2" grid_ref="1" id="1" transient_ref="1"/>
    <transient_grid_ref collection_size="2" grid_ref="1" id="2" transient_ref="2"/>
    <transient_grid_ref collection_size="3" grid_ref="1" id="3" transient_ref="3"/>
    <transient_grid_ref collection_size="4" grid_ref="1" id="4" transient_ref="4"/>
    <transient_grid_ref collection_size="4" grid_ref="1" id="5" transient_ref="5"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="6" transient_ref="6"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="7" transient_ref="7"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="8" transient_ref="8"/>
    <transient_grid_ref collection_size="5" grid_ref="1" id="9" transient_ref="9"/>
  </transient_grid_refs>
  <transients>
    <transient id="1" transient_standard_name="surface_downward_eastward_stress"/>
    <transient id="2" transient_standard_name="surface_downward_northward_stress"/>
    <transient id="3" transient_standard_name="surface_fresh_water_flux"/>
    <transient id="4" transient_standard_name="total_heat_flux"/>
    <transient id="5" transient_standard_name="atmosphere_sea_ice_bundle"/>
    <transient id="6" transient_standard_name="sea_surface_temperature"/>
    <transient id="7" transient_standard_name="eastward_sea_water_velocity"/>
    <transient id="8" transient_standard_name="northward_sea_water_velocity"/>
    <transient id="9" transient_standard_name="ocean_sea_ice_bundle"/>
  </transients>
  <grids>
    <grid id="1" alias_name="grid1"/>
  </grids>
</component>
<name>atmo</name>

<model>ICON</model>

<simulated>atmosphere</simulated>

CALL YAC_fdef_comp ( "atmo", comp_id )
<?xml version="1.0" encoding="UTF-8"?>
<component>
    <xmlns>http://www.w3schools.com</xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.component.xsd">
        <id>1</id>
        <name>atmo</name>
        <model>ICON</model>
        <simulated>atmosphere</simulated>
        <transient_grid.refs>
            <transient_grid_ref collection_size="2" grid_ref="1" id="1" transient_ref="1"/>
            <transient_grid_ref collection_size="2" grid_ref="1" id="2" transient_ref="2"/>
            <transient_grid_ref collection_size="3" grid_ref="1" id="3" transient_ref="3"/>
            <transient_grid_ref collection_size="4" grid_ref="1" id="4" transient_ref="4"/>
            <transient_grid_ref collection_size="4" grid_ref="1" id="5" transient_ref="5"/>
            <transient_grid_ref collection_size="1" grid_ref="1" id="6" transient_ref="6"/>
            <transient_grid_ref collection_size="1" grid_ref="1" id="7" transient_ref="7"/>
            <transient_grid_ref collection_size="1" grid_ref="1" id="8" transient_ref="8"/>
            <transient_grid_ref collection_size="5" grid_ref="1" id="9" transient_ref="9"/>
        </transient_grid.refs>
        <transients>
            <transient id="1" transient_standard_name="surface_downward_eastward_stress"/>
            <transient id="2" transient_standard_name="surface_downward_northward_stress"/>
            <transient id="3" transient_standard_name="surface_fresh_water_flux"/>
            <transient id="4" transient_standard_name="total_heat_flux"/>
            <transient id="5" transient_standard_name="atmosphere_sea_ice_bundle"/>
            <transient id="6" transient_standard_name="sea_surface_temperature"/>
            <transient id="7" transient_standard_name="eastward_sea_water_velocity"/>
            <transient id="8" transient_standard_name="northward_sea_water_velocity"/>
            <transient id="9" transient_standard_name="ocean_sea_iceBundle"/>
        </transients>
        <grids>
            <grid id="1" alias_name="grid1"/>
        </grids>
    </component>
<transient_grid_refs>
    <transient_grid_ref collection_size="2" grid_ref="1" id="1" transient_ref="1"/>
    <transient_grid_ref collection_size="2" grid_ref="1" id="2" transient_ref="2"/>
    <transient_grid_ref collection_size="3" grid_ref="1" id="3" transient_ref="3"/>
    <transient_grid_ref collection_size="4" grid_ref="1" id="4" transient_ref="4"/>
    ...
    <transient_grid_ref collection_size="5" grid_ref="1" id="9" transient_ref="9"/>
</transient_grid_refs>

CALL yac_fget ( field_id, nbr_hor_points, 2, &
                 1, 1, buffer(1:nbr_hor_points,1:2), &
                 info, ierror )

CALL yac_fput ( field_id, nbr_hor_points, 5, &
                 1, 1, buffer(1:nbr_hor_points,1:5), &
                 info, ierror )
<?xml version="1.0" encoding="UTF-8"?>
<component
xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.component.xsd">
  <id>1</id>
  <name>atmo</name>
  <model>ICON</model>
  <simulated>atmosphere</simulated>
  <transient_grid_refs>
    <transient_grid_ref collection_size="2" grid_ref="1" id="1" transient_ref="1"/>
    <transient_grid_ref collection_size="2" grid_ref="1" id="2" transient_ref="2"/>
    <transient_grid_ref collection_size="3" grid_ref="1" id="3" transient_ref="3"/>
    <transient_grid_ref collection_size="4" grid_ref="1" id="4" transient_ref="4"/>
    <transient_grid_ref collection_size="4" grid_ref="1" id="5" transient_ref="5"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="6" transient_ref="6"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="7" transient_ref="7"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="8" transient_ref="8"/>
    <transient_grid_ref collection_size="5" grid_ref="1" id="9" transient_ref="9"/>
  </transient_grid_refs>
  <transients>
    <transient id="1" transient_standard_name="surface_downward_eastward_stress"/>
    <transient id="2" transient_standard_name="surface_downward_northward_stress"/>
    <transient id="3" transient_standard_name="surface_fresh_water_flux"/>
    <transient id="4" transient_standard_name="total_heat_flux"/>
    <transient id="5" transient_standard_name="atmosphere_sea_ice_bundle"/>
    <transient id="6" transient_standard_name="sea_surface_temperature"/>
    <transient id="7" transient_standard_name="eastward_sea_water_velocity"/>
    <transient id="8" transient_standard_name="northward_sea_water_velocity"/>
    <transient id="9" transient_standard_name="ocean_sea_ice_bundle"/>
  </transients>
  <grids>
    <grid id="1" alias_name="grid1"/>
  </grids>
</component>
<transients>
  <transient id="1">transient_standard_name="surface_downward_eastward_stress"/></transient>
  <transient id="2">transient_standard_name="surface_downward_northward_stress"/></transient>
  <transient id="3">transient_standard_name="surface_fresh_water_flux"/></transient>
  <transient id="4">transient_standard_name="total_heat_flux"/></transient>
  ...
  <transient id="9">transient_standard_name="ocean_sea_ice_bundle"/></transient>
</transients>

CALL yac_fdef_field &
  ( "surface_downward_eastward_stress", &
    component_id, subdomain_id, point_id, &
    mask_id, 1, field_id(1) )
  ...

CALL yac_fdef_field &
  ( "ocean_sea_ice_bundle", &
    component_id, subdomain_id, point_id, &
    mask_id, 1, field_id(9) )
<?xml version="1.0" encoding="UTF-8"?>
<component
  xmlns="http://www.w3schools.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3schools.component.xsd">
  <id>1</id>
  <name>atmo</name>
  <model>ICON</model>
  <simulated>atmosphere</simulated>
  <transient_grid_refs>
    <transient_grid_ref collection_size="2" grid_ref="1" id="1" transient_ref="1"/>
    <transient_grid_ref collection_size="2" grid_ref="1" id="2" transient_ref="2"/>
    <transient_grid_ref collection_size="3" grid_ref="1" id="3" transient_ref="3"/>
    <transient_grid_ref collection_size="4" grid_ref="1" id="4" transient_ref="4"/>
    <transient_grid_ref collection_size="4" grid_ref="1" id="5" transient_ref="5"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="6" transient_ref="6"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="7" transient_ref="7"/>
    <transient_grid_ref collection_size="1" grid_ref="1" id="8" transient_ref="8"/>
    <transient_grid_ref collection_size="5" grid_ref="1" id="9" transient_ref="9"/>
  </transient_grid_refs>
  <transients>
    <transient id="1" transient_standard_name="surface_downward_eastward_stress"/>
    <transient id="2" transient_standard_name="surface_downward_northward_stress"/>
    <transient id="3" transient_standard_name="surface_fresh_water_flux"/>
    <transient id="4" transient_standard_name="total_heat_flux"/>
    <transient id="5" transient_standard_name="atmosphere_sea_ice_bundle"/>
    <transient id="6" transient_standard_name="sea_surface_temperature"/>
    <transient id="7" transient_standard_name="eastward_sea_water_velocity"/>
    <transient id="8" transient_standard_name="northward_sea_water_velocity"/>
    <transient id="9" transient_standard_name="ocean_sea_ice_bundle"/>
  </transients>
  <grids>
    <grid id="1" alias_name="grid1"/>
  </grids>
</component>
YAC – Component XML configuration

<grids>
  <grid id="1" alias_name="grid1"/>
</grids>

CALL yac_fdef_subdomain ( component_id, "grid1", subdomain_id )
<transient_grid.refs>
  <transient_grid_ref collection_size="2" grid_ref="1" id="1" transient_ref="1"/>
  <transient_grid_ref collection_size="2" grid_ref="1" id="2" transient_ref="2"/>
  <transient_grid_ref collection_size="3" grid_ref="1" id="3" transient_ref="3"/>
  <transient_grid_ref collection_size="4" grid_ref="1" id="4" transient_ref="4"/>
  ...
  <transient_grid_ref collection_size="5" grid_ref="1" id="9" transient_ref="9"/>
</transient_grid.refs>

<transients>
  <transient id="1" transient_standard_name="surface_downward_eastward_stress"/>
  <transient id="2" transient_standard_name="surface_downward_northward_stress"/>
  <transient id="3" transient_standard_name="surface_fresh_water_flux"/>
  <transient id="4" transient_standard_name="total_heat_flux"/>
  ...
  <transient id="9" transient_standard_name="ocean_sea_ice_bundle"/>
</transients>

<grids>
  <grid id="1" alias_name="grid1"/>
</grids>
# YAC – XML configuration

## New Coupling

### Transients

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Grid</th>
<th>Collect. Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>total_heat_flux</code></td>
<td>grid1</td>
<td>4</td>
</tr>
<tr>
<td><code>atmosphere_sea_ice_bundle</code></td>
<td>grid1</td>
<td>4</td>
</tr>
<tr>
<td><code>sea_surface_temperature</code></td>
<td>grid1</td>
<td>1</td>
</tr>
<tr>
<td><code>eastward_sea_water_velocity</code></td>
<td>grid1</td>
<td>1</td>
</tr>
<tr>
<td><code>northward_sea_water_velocity</code></td>
<td>grid1</td>
<td></td>
</tr>
</tbody>
</table>

### Basic Settings

- **Calendar**: proleptic-gregorian
- **Start date**: 1800-01-01T00:00:00.000
- **End date**: 2100-01-01T00:00:00.000
- **Timestep unit**: second

---

Max-Planck-Institut für Meteorologie

YAC – Tutorial, Version 1.5.5
YAC – XML configuration

Coupling for eastward\_sea\_water\_velocity (grid1 -> grid1)

- Enforce write weight file
- Choose preferred interpolation method:
  - Use source mask
  - Use target mask

Option 0: n-nearest\_neighbor
- n: 1

Weighted: ARITHMETIC\_AVERAGE

Option 1: fixed\_value
- user value: -999

Option 2: none selected

Add more interpolations

Save

Close
YAC – XML configuration

Coupling for eastward_sea_water_velocity (grid1 -> grid1)

Source timestep: 3600 seconds

Target timestep: 1.200 seconds

Coupling period: 3.600 seconds

Operation: accumulate

Source Time Lag: 1 model timestep(s)

Target Time Lag: 2 model timestep(s)

Forget default parameters

Save Close
YAC – XML configuration

Source time step

• time interval between two consecutive calls to yac_fput

Target time step

• time interval between two consecutive calls to yac_fget

Requirement: Source or target time step must be equal to or an integer multiple of the other.

Coupling period

• Time interval at which data are exchanged (with internal calls to MPI_SEND and MPI_RECV)

Requirement: Coupling period must be an integer multiple of the source/target time step
YAC – Examples

ICON_toy_cube.c  toy_icon_[atm/ocn].c
ICON_toy_reg2d.c  toy_mpiom_ocn.c
ICON_toy_unstruct.c  toy_reg2d_[atm/ocn].c
ICON_toy_unstruct_2.c  toy_woa_ocn.c

dummy_atmosphere.F90  dummy_atmosphere_c.c
dummy_io.F90  dummy_io_c.c
dummy_ocean.F90  dummy_ocean_c.c
ICON Implementation

Atmosphere  mo_interface_echam_ocean.f90

Initialisation, definition and search:  construct_atmo_coupler
Exchange of coupling fields:  interface_echam_ocean
Termination phase:  destruct_atmo_coupler

Land – HD model  mo_interface_hd_ocean.f90

Definition:  jsb_fdef_hd_fields
Exchange of runoff:  interface_hd_ocean

Ocean  mo_ocean_coupling.f90

Initialisation, definition and search:  construct_ocean_coupling
Exchange of coupling field:  couple_ocean_toatmo_fluxes
Termination phase:  destruct_ocean_coupling
YetAnotherCoupler 1.5.0

Related Pages

Here is a list of all related documentation pages:

- Sphere Partitioning Algorithm
- Polygon clipping in YAC
- Example on how to use XML routines from config_xml.h
- Configuration examples for different systems
- Tips’n Tricks for developers
- Description of how to build and run the Java GUI
- The c interface (yac_interface.h)
- The Fortran interface (yac_finterface.f90 and mo_yac_finterface.f90)
- Patch Recovery in YAC
- Issue with Patch Recovery in YAC
- Condensed release information
- Todo List
checkout ICON

The default configure will compile and build icon with yac. In order to deactivate the compilation with yac

./configure … --disable-yac …

./build_command

**Known issues:**

- **pgc**: fails with communicator_local.c
- **nag**: `-Wc,-O3 -Wc,-march=native -float-store` causes internal compiler error on Debian wheezy
- **MPI**: `MPI_(un)pack_external`
Doxygen

http://dkrz-sw.gitlab-pages.dkrz.de/yac/

Source Code (version 1.5.5)

```
git clone -b 'release-1.5.5' --single-branch --depth 1 \
git@gitlab.dkrz.de:YAC/YAC.git
```

Latest version (untagged)

```
git clone git@gitlab.dkrz.de:YAC/YAC.git
```

Documentation with further Links

- https://www.geosci-model-dev.net/9/2755/2016/
- https://doi.org/10.5676/dwd_pub/nwv/icon_003
- https://code.zmaw.de/projects/mpiesm-2/wiki/ICON_Coupled_Model_Development