Recent enhancements to XIOS

(v2.0 Andrew Coward/Antony Siahaan August 2014)

Thanks to new updates in xios, there are several new, interesting functionalities available.

- Access to the new functionality does not require any change in the NEMO source code. you just have to change iodef.xml and field_def.xml.
- Tested with xios rev 482.

Here is a description of the new functionalities:

1) examples of simple computations (iodef.xml only):

```
<field field_ref="sst" name="tosK" unit="degK" > sst+273.15 </field>
<field field_ref="sst" name="tostos" unit="degC2" > sst*sst </field>
<field field_ref="sst" name="tos_sos" > sst + sss </field>
```

2) examples of more complex computations (needs the modification of iodef.xml and field_def.xml)

example 1: monthly average of daily maximum of the SST

define sstmax in field_def.xml (for example just below the definition of sst)

```
<field id="sstmax" field_ref="sst" long_name="sea surface temperature" operation="maximum" />
```

edit iodef.xml to use sstmax in a monthy file:

```
 <file_group id="1m" output_freq="1m"  output_level="10" enabled=".TRUE."> <!-- 1m files -->
  <file id="file1" name_suffix="_grid_T" description="ocean T grid variables" >
   <field field_ref="sstmax" name="tos_1dmax" operation="average" freq_op="1d" > @sstmax </field>
  </file>
 </file_group>
```

N.B. the use of @ before the name of the variable tells XIOS to do the temporal operation, at frequency freq_op, defined in the variable definition (in this example: maximum and 1d), before doing the temporal operation, at frequency output_freq, defined when listing the variables in the file (here: average and 1m)

example 2: daily weighted temporal average

define toce_wei in field_def.xml (for example just bellow the definition of toce)

```
<field id="toce_wei" long_name="weighted temperature" unit="degC" grid_ref="grid_T_3D" > toce * cellthc </field>
```

use toce_wei in a daily file:

```
<field field_ref="toce_wei"  name="toce_wei" operation="instant" freq_op="1d" > @toce_wei / @cellthc </field>
```

example 3: daily daily sst variance

```
<field field_ref="sst"  name="tosvar" operation="instant" freq_op="1d" > @sst2 - (@sst * @sst) </field>
```

note: in this example we use sst2 that is already computed in NEMO. this feature becomes obsolete in the code and will be replaced by a simple definition in field_def.xml:

```
<field id="sst2" long_name="square of sea surface temperature" unit="degC2" > sst * sst </field>
```

3) examples of change of precision:

```
<field field_ref="sst" name="tos_r8" prec="8" />
    <!-- integer 2  with add_offset and scale_factor attributes -->
<field field_ref="sss" name="sos_i2" prec="2" add_offset="20." scale_factor="1.e-3" />
```

4) examples of adding user defined attributes:

```
<file_group id="1d" output_freq="1d" output_level="10" enabled=".TRUE."> <!-- 1d files -->
 <file id="file1" name_suffix="_grid_T" description="ocean T grid variables" >
    <field field_ref="sst" name="tos" >
        <variable id="my_attribute1" type="string" >blabla</variable>
        <variable id="my_attribute2" type="integer" >3</variable>
        <variable id="my_attribute3" type="float" >5.0</variable>
    </field>
    <variable id="my_global_attribute" type="string" >blablaglobal</variable>
  </file>
 </file_group>
```

Example Application:

Steps to obtain mean tracer fields from NEMO v3.6 runs using the non-linear free surface (key_vvl)

Introduction

From NEMO v3.6 onwards, mean tracer fields from any runs using the non-linear free surface will be thickness weighted means. Unwary users may suddenly find that vertical profiles of tracer values appear distinctly unphysical. A typical example would be apparently normal surface temperatures (assuming a surface layer ~1m) but temperatures increasing to several hundreds of degrees at depths. This is not a model error but is by design because, with cell thicknesses varying in time, the correct cell average is:

```
<e3t*T>/<e3t>
```

where <> denotes the temporal averaging operator. To recover the expected average tracer values, the average fields appearing in the grid_T diagnostic output need to be divided by the average cell thickness for the same averaging period. The average cell thickness is a standard output field (field_ref=cellthc) which can be included in the mean output fields by the simple addition of a line such as:

```
<field field_ref="cellthc" name="e3t" />
```

To each mean file id ithe field is required in. If the mean e3t field is present, the required scaling operation can be carried out with common tools such as ncap2.

Users may question why the tracer fields continue to be labelled as entries such as "sst" or "vosaline" when they need additional post-processing to return these actual fields. One, pragmatic, answer is that doing so removes the need to employ different XML files with linear and non-linear free-surface runs. A better answer, though, is that they were left in this state in anticipation of the updates to the XIOS package that would allow the IO servers to perform the rescaling operation at the point of output. These enhancements have now been released (see top section) and, from revision 482, the XIOS servers can be instructed to perform arithmetic operations with combinations of output fields. This note describes the steps necessary to enable this functionality.

Step 1: Update XIOS to the latest version (or at least rev 482) and recompile. Details will depend on your local installation but a typical procedure might be:

```
cd $XIOS_HOME
svn update
make_xios --prod --arch your_architecture --netcdf_lib netcdf4_par --job 8
```

Step 2: Make sure any local copies or symbolic links to xios_server.exe are updated to the new version

Step 3: Add a new output field of surface cell thickness for use with surface tracer fields. This is necessary because, although the information is already present in the 3D cell thickness field, the XIOS arithmetic operations can only be performed between identical shape arrays. Adding a new output field requires four sub-steps:

a. Insert an `iom_put` statement in `domvvl.F90`. I.e.:

```
CALL iom_put( "cellthc" , fse3t_n  (:,:,:) )
```

becomes:

```
CALL iom_put( "cellthc" , fse3t_n  (:,:,:) )
CALL iom_put( "cellthc2d" , fse3t_n  (:,:,1) )
```

b. Add a definition of the new field to `field_def.xml`. I.e.:

```
<!-- variables available with key_diaar5/key_vvl -->
<field id="cellthc"   long_name="Cell thickness"            unit="m"  grid_ref="grid_T_3D"/>
```

becomes:

```
<!-- variables available with key_diaar5/key_vvl -->
<field id="cellthc"   long_name="Cell thickness"            unit="m"  grid_ref="grid_T_3D"/>
<field id="cellthc2d" long_name="Sea surface cell thickness" unit="m"                     />
```

c. Add the new field to the `iodef.xml` file for the target experiment. I.e.:

```
<field field_ref="cellthc"    name="e3t"   />
<field field_ref="cellthc2d"  name="e3t2d" />
```

d. Recompile nemo

Step 4: Change entries in the `iodef.xml` file for those entries that need to be rescaled. A typical set would be:

```
    .          .
<file id="file1" name_suffix="_grid_T" description="ocean T grid variables" >
  <field field_ref="sst"     name="sst"      long_name="sea_surface_temperature"         operation="instant" freq_op="1m" > @sst / @cellthc2d </field>
  <field field_ref="sss"     name="sss"      long_name="sea_surface_salinity"            operation="instant" freq_op="1m" > @sss / @cellthc2d </field>
  .
  <field field_ref="toce"    name="potemp"   long_name="sea_water_potential_temperature" operation="instant" freq_op="1m" > @toce / @cellthc </field>
  <field field_ref="soce"    name="salin"    long_name="sea_water_salinity"              operation="instant" freq_op="1m" > @soce / @cellthc </field>
 .
```

Step 5: Run the model and confirm that the output is scaled correctly.

Steps to add output files for a zoom area

Step 1: Add the zoom area in the `domain_def.xml` inside the `"<domain_definition>"` section, e.g. :

```
<domain_definition>
 <domain_group id="grid_T">
  <!--   example of hand defined zoom   -->
  <domain id="zoomArea" zoom_ibegin="160" zoom_jbegin="160" zoom_ni="20" zoom_nj="20" />
```

Step 2: For normal variables the defined zoom region can be used to output data for just that region. For example:

```
<file_group id="3h" output_freq="3h"  output_level="10" enabled=".TRUE." split_freq="5d" sync_freq="1d"> <!-- 3h files -->
   <file id="file31" name_suffix="_zoom_T" description="ocean T grid variables" >
      <field_group id="field_3h_zoomT"  domain_ref="zoomArea" >
         <field field_ref="toce" name="potemp" long_name="sea_water_potential_temperature" />
      </field_group>
   </file>
</file_group>
```

but the system will fail if any naive attempt is made to output derived variables using the arithmetic operations. For example:

```
<file_group id="3h" output_freq="3h"  output_level="10" enabled=".TRUE." split_freq="5d" sync_freq="1d"> <!-- 3h files -->
   <file id="file31" name_suffix="_zoom_T" description="ocean T grid variables" >
      <field_group id="field_3h_zoomT"  domain_ref="zoomArea" >
       <field field_ref="toce" name="potemp" long_name="sea_water_potential_temperature" operation="instant" freq_op="3h" > @toce / @cellthc </field>
      </field_group>
   </file>
</file_group>
```

will fail. This is because, without additional specification, the expression operates on the global domain but the zoom attribute has been passed by group inheritance and the resultant field is larger than expected. Fortunately these difficulties can be avoided by defining intermediate fields with the appropriate zoom attribute that can be used in the expression. Steps 3 and 4 provide an example of how this can be achieved:

Step 3: Define a new field group in `field_def.xml` for all variables to be used in expressions in the zoom area. This is done by referencing the corresponding `field_ref` and `domain_ref.`

```
<field_group id="groupZoom" >
    <field id="zoom_sst"       field_ref="sst"       domain_ref="zoomArea" />
    <field id="zoom_toce"      field_ref="toce"      domain_ref="zoomArea" />
    <field id="zoom_cellthc"   field_ref="cellthc"   domain_ref="zoomArea" />
    <field id="zoom_cellthc2d" field_ref="cellthc2d" domain_ref="zoomArea" />
</field_group>
```

Step 4: Use the new field group in the definition of output file for the zoom area in `iodef.xml` by using the keyword `group_ref` to reference it.

```
<file_group id="3h" output_freq="3h"  output_level="10" enabled=".TRUE." split_freq="5d" sync_freq="5d"  >
   <file id="file31" name_suffix="_zoom_T" description="ocean T grid variables" >
      <field_group id="field_3h_zoomT" group_ref="groupZoom"  domain_ref="zoomArea" >
         <field field_ref="zoom_sst"  name="sst_zoom"    long_name="sea_surface_temperature"         operation="instant" freq_op="3h">  @zoom_sst  / @zoom_cellthc2d </field>
         <field field_ref="zoom_toce" name="potemp_zoom" long_name="sea_water_potential_temperature" operation="instant" freq_op="3h">  @zoom_toce / @zoom_cellthc   </field>
      </field_group>
   </file>
</file_group>
```

If the thickness field `cellthc2d` does not yet exist, then it can be added following the procedure described in the previous section. Bear in mind that 2D fields, e.g. sea surface temperature (`sst`) much be operated with surface cell thickness (`cellthc2d`) instead of cell thickness (`cellthc`).

Adding global attributes for the zoom area

XIOS does not automatically add sufficient information in the global attributes of a zoom file to clearly identify the region. It is recommended that this data is added explicitly by using the new "user defined attribute" mechanism. To do this, attributes can be added in the same file group defined for the zoom area using the `<variable>` context.

```
<file_group id="3h" output_freq="3h"  output_level="10" enabled=".TRUE." split_freq="5d" sync_freq="5d"  >
  <file id="file31" name_suffix="_zoom_T" description="ocean T grid variables" >
     <field_group id="field_3h_zoomT" group_ref="groupZoom"  domain_ref="zoomArea" >
        <field field_ref="zoom_sst"  name="sst_zoom"    long_name="sea_surface_temperature"         operation="instant" freq_op="3h">  @zoom_sst  / @zoom_cellthc2d </field>
        <field field_ref="zoom_toce" name="potemp_zoom" long_name="sea_water_potential_temperature" operation="instant" freq_op="3h">  @zoom_toce / @zoom_cellthc   </field>
     </field_group>

     <variable id="DOMAIN_zoom_ibegin"   type="integer">160</variable>
     <variable id="DOMAIN_zoom_jbegin"   type="integer">160</variable>
     <variable id="DOMAIN_zoom_ni"       type="integer">20</variable>
     <variable id="DOMAIN_zoom_nj"       type="integer">20</variable>

  </file>
</file_group>
```