

Integration of Biogeochemical Models Into the Most Recent FABM Framework

What is FABM

- Framework for Aquatic Biogeochemical Models (FABM)
- developing a biogeochemical model implemented in the FABM in the current version of GOTM .
- Can link other models coupling to a collection of biogeochemical model
- Biogeochemical models that contain the initialization, subroutines, and function code within modules
- FABM relies on object oriented features allowing model data to remain contained within the files.

Steps to implementing the biogeochemical models within FABM

Adding subdirectory

- Within fabm/src/models create a subdirectory with an identifier that describes the institution that the model is coming from.
- In our case that identifier is “uvic.”

Adding Files

- Within the uvic subdirectory include the model files from the mortenson 2017 code.
- The included files are:

~/gotm/code/extern/fabm/src/models/uvic/eco.F90

~/gotm/code/extern/fabm/src/models/uvic/icealgae.F90

~/gotm/code/extern/fabm/src/models/uvic/npzd_resolute.F90

~/gotm/code/extern/fabm/src/models/uvic/dic.F90

~/gotm/code/extern/fabm/src/models/uvic/dms.F90

~/gotm/code/extern/fabm/src/models/uvic/icedms.F90

Creation of uvic_model_library.F90

- Within the uvic subdirectory I have created a uvic_model_library.F90 File.

~/gotm/code/extern/fabm/src/models/uvic/uvic_model_library.F90

- Following the template for model_library files. The additions made were:
 - The inclusion of a use statement for the model files:

```

use uvic_eco
use uvic_icealgae
use uvic_dic
use uvic_npzd_resolute
use uvic_icedms
use uvic_dms

```

- To link the name of the included model to its derived type:

```

select case (name)
  case ('uvic_eco');          allocate(type_uvic_eco::model)
  case ('uvic_icealgae');     allocate(type_uvic_icealgae::model)
  case ('uvic_npzd_resolute'); allocate(type_uvic_npzd_resolute::model)
  case ('uvic_dic');          allocate(type_uvic_dic::model)
  case ('uvic_dms');          allocate(type_uvic_dms::model)
  case ('uvic_icedms');       allocate(type_uvic_icedms::model)

```

- The new version of FABM has a fabm_library.F90.in file that automates the generation of references to modules within institutes' directories
- Useful because when new models are added, no changes are required within the FABM code

Creation of CMakeLists.txt

- Within the uvic directory I have created a CMakeLists.txt file which has instructions for the source files to compile for CMake.
- It includes the files for all the models in the uvic directory and the uvic_model_library files.

~/gotm/code/extern/fabm/src/models/uvic/CMakeLists.txt

```

add_library(fabm_models_uvic OBJECT
    uvic_model_library.F90
    eco.F90
    icealgae.F90
    npzd_resolute.F90
    dic.F90
    dms.F90
    icedms.F90
)

add_dependencies(fabm_models_uvic fabm_base)

```

Addition to CmakeLists.txt

- This is the only addition within the FABM source that is needed:
 - Added name of the new directory to an existing CMake file
`~/gotm/code/extern/fabm/src/CMakeLists.txt`

```

# List of contributing institutes that should be included in compilation.
# When adding new institutes to the source tree, please add them here as well.
# You can exclude institute directories from compilation by commenting them out.
set(DEFAULT_INSTITUTES
    akvaplan      # Akvaplan-niva, Norway
    au            # University of Aarhus, Denmark
    bb            # Bolding & Bruggeman - formerly Bolding & Burchard
    csiro         # Commonwealth Scientific and Industrial Research Organisation, Australia
    ersem         # European Regional Seas Ecosystem Model
    examples      # Examples supplied with FABM itself
    gotm          # Models ported from original GOTM/BIO library
    iow           # Leibniz Institute for Baltic Sea Research, Germany
    jrc           # EC - Joint Research Centre, Ispra, Italy
    msi           # Marine Systems Institute, Tallinn University of Technology, Estonia
    niva          # Norwegian Institute for Water Research, Norway
    pclake        # The PCLake model - reference implementation
    pml           # Plymouth Marine Laboratory, United Kingdom
    selma         # Simple EcoLogical Model for the Aquatic - PROGNOS
    su            # Swansea University, United Kingdom
    uhh           # University of Hamburg, Germany
    uvic          # University of Victoria, Canada
)

```

Changes made to the variables.yaml file

~/gotm/code/extern/fabm/util/standard_variables/variables.yaml

- Added additional standard variable that were included in the mortenson2017 version of the code

Execute:

python parse_standard_variables.py

Within ~/gotm/code/extern/fabm/util/standard_variables

To parse the variables.yaml file

Which generated the necessary files for the inclusion of the standard variables in FABM

- (only need to rerun the python script if making additions to the variable.yaml)

python parse_standard_variables.py

Changes made to the gotm.yaml file

- Within the gotm.yaml under fabm:

~/gotm/cases/resolute/gotm.yaml

```
fabm:
  use: true
  freshwater_impact: true
  feedbacks:
    shade: true #false
    albedo: true #false
    surface_drag: true #false
  repair_state: false
  input:
  numerics:
    ode_method: 3
    split_factor: 1
    w_adv_discr: 6
    cnpar: 1
```

- Set use to true
- Set feedback settings to true to align with the settings in the mortenson2017 resolute test case.
- Include other settings from Mortenson2017
- Reading of these parameters is already done within gotm_fabm.F90

Creation of fabm.yaml

- Created fabm.yaml within resolute test case directory

- Begins with an instances element, and below you include the model blocks beginning with an identifier for the model, the path to the model within the models directory, and names and values of parameters to be read in by the model
- The use statement is optional, but allows the user to turn on and off models.
- Each model has its own block of 'parameters'.
- If parameters are not given values within the yaml, the default value provided in the registering of the variables is used.
- Included use_name_of_model parameter to the models that require the option to turn other models on and off.

```
instances:
  uvic_eco: # University of Victoria Ecosystem model
    use: true
    model: uvic/uvic_eco
    parameters:
      use_icealgae: true
      ac : 0.03           # light attenuation coefficient [m-1]
      f_seed: 0.1
```

- Some variables that are registered with:

```
call self%register_state_variable(self%
```

are declared under an 'initialize:' block under the model identifier.

```
initialization:
  no3: 7.2
  sil: 14.7
  ia: 0.017
  nh4: 0.01
```

Additions to gotm_fabm.F90

~/gotm/code/src/fabm/gotm_fabm.F90

- Included fetch from the addition of timestep to the standard variables within the init_var_gotm_fabm subroutine

```
! !INTERFACE:
subroutine init_var_gotm_fabm(nlev)
```

```
call model%link_scalar(standard_variables%number_of_days_since_start_of_the_year,decimal_year)
call model%link_scalar(standard_variables%timestep,dt)
```

Additions to gotm.F90

- Linked ice data to FABM

```
! initialize FABM module
#ifdef _FABM_

if (fabm_calc) then
  call init_gotm_fabm(nlev,dt,fm)
```

```
call model_fabm%link_horizontal_data(standard_variables_fabm%sea_ice_thickness,ice_hi)
call model_fabm%link_horizontal_data(standard_variables_fabm%snow_thickness,ice_hs)
call model_fabm%link_horizontal_data(standard_variables_fabm%topmelt,ice_uvic_topmelt)
call model_fabm%link_horizontal_data(standard_variables_fabm%f_melt,ice_uvic_Amelt)
call model_fabm%link_horizontal_data(standard_variables_fabm%topgrowth,ice_uvic_topgrowth)
call model_fabm%link_horizontal_data(standard_variables_fabm%termelt,ice_uvic_termelt)
call model_fabm%link_horizontal_data(standard_variables_fabm%tendency_of_sea_ice_thickness,d
call model_fabm%link_horizontal_data(standard_variables_fabm%tendency_of_sea_ice_thickness,d
```

- Some of which is referencing the standard variables previously included in the variables.yaml
- Changed some calls to mode_fabm to align with the structure of variables in variables.yaml
-

Changed made within the model code files

- Similar changes made for every model brought in under the uvic subdirectory:
- For the initialization variables changed method of reading in parameter values, from namelist to yaml
- Changed method of setting default values for read in parameters
- Changed method of assigning the parameters to the model object
- Included new source: source = do_horizontal for call self%register_horizontal_diagnostic_variable
- Changed if statement to use other model data to my change to use_name_of_model in the fabm.yaml
- Changed structure of call to self%register_dependecny to account for different structure of subroutines in new version of FABM

Steps for Compilation and example output

Obtain source code

`git@github.com:PatrickFarnole/code.git`

Create necessary build directories

```
mkdir ~/build
cd build
mkdir fabm
mkdir gotm_fabm
```

Compile

```
cd ~/build/fabm
cmake ~/gotm/code/extern/fabm -DFABM_HOST=gotm
make install
```

```
cd ~/build/gotm_fabm
FABM=true && FABMDIR=~ /gotm/code/extern/fabm && FABM_PREFIX=~ /local/fabm/gotm &&
FORTRAN_COMPILER=GFORTRAN && NETCDF_VERSION$
cmake ~/gotm/code -DGOTM_USE_STIM=ON -DGOTM_USE_FABM=ON
make install
```

Run

```
cd ~/gotm/cases/resolute
~/build/gotm_fabm/gotm
```