Metos3D Documentation

Release 0.0.1-3

Jaroslaw Piwonski

Feb 21, 2018

Contents:

CHAPTER 1

Metos3D documentation

go back

addition documentation

- cheat sheet
- tutorial
- reference

CHAPTER 2

metos3d cheat sheet

blubi blubb

`metos3d info ~/.metos3drc`

`version data model petsc`

```
4.
$> metos3d
usage: metos3d [command] ...
metos3d
    --help, -h
    --verbose, -v
    info            # print version, config file, docs, etc ...
    simulate        # compile sources, perform a simulation, ...

    update          # not needed, handled externally, pip, conda
    optimize        # save for later

$>
metos3d info
metos3d update
metos3d simulate HAMMOC-mitgcm-128x64x15-3h.yaml

metos3d simulate N-DOP-mitgcm-360x160x23-
metos3d simulate HAMMOC-mpiom-default-mitgcm-128x64x15-3h.yaml
metos3d simulate HAMMOC-ballasting-0.3-mitgcm-128x64x15-3h.yaml
mpirun -n 128 metos3d simulate HAMMOC-mitgcm-128x64x15-3h.yaml

3.
metos3d
    init
    data
    model
    simulation      (sim)
    optimization    (opt)
    info
```

```
    update

2.
metos3d                 (m3d)
    init
    data
    model
    experiment     (exp)

1.
metos3d
    info
    update

    data
    petsc

    run
    model
    compile
    bgc
    tmm
    matrix
```

2.1 update

2.2 info

2.3 usage

```
$> python metos3d.py
```

will print the short usage description

2.4 help

print help texts for each argument

```
$> python metos3d.py -h
```

CHAPTER 3

Metos3D BGC API

Application programming interface (API) for biogeochemical (BGC) models

Definition, realization in *Fortran*

```
metos3dbgcinit(...)
metos3dbgcbegin(...)
metos3dbgc(...)
metos3dbgcend(...)
metos3dbgcfinal(...)
```

3.1 Description

The interface decouples biogeochemical models and driver routines (ocean circulation, forcing, geometry) programmatically.

It gives the modeler the possibility to provide a free number of tracers, parameters, boundary and domain conditions. It suits well an optimization as well as an Automatic Differentiation (AD) context.

The interface changed (more or less) since it was introduced for the first time. The initial version can be found at \citep[][]{PiwSla16}.

3.1.1 What is it?

Ther Metos3D BGC API is a convention how biogeochemical models can be coupled to ocean circulation.

3.1.2 What for?

modelers that want to their marine ecosystem or biogeochemical models, coupled to a global ocean circulation,

3.1.3 Why?

climate research, ocean simulation, bgc models, uncertainties, parameters, processes, require assessment, at global-basin scale, sophisticated general circulation model,

3.1.4 How?

C model template

Fortran model template

Example 1

in the simplest case, assuming you have a model written in Fortran,

File: `simple_bgc_model.f90`

```
subroutine metos3dbgc(ny, nx, nu, nb, nd, ndg, dt, q, t, y, u, b, d, dg, ctx)

    integer :: ny           ! tracer count
    integer :: nx           ! layer count
    integer :: nu           ! parameter count
    integer :: nb           ! boundary condition count
    integer :: nd           ! domain condition count
    integer :: ndg          ! diagnostic variable count
    real(8) :: dt           ! ocean time step
    real(8) :: q(nx, ny)    ! bgc model output
    real(8) :: t            ! point in time
    real(8) :: y(nx, ny)    ! bgc model input
    real(8) :: u(nu)        ! parameters
    real(8) :: b(nb)        ! boundary conditions
    real(8) :: d(nx, nd)    ! domain conditions
    real(8) :: dg(nx, ndg)  ! diagnostic variables
    integer :: ctx          ! unsed variable, place holder for bgc context

    ! your code here ...

end subroutine
```

Example 2

File: `bgctype.h90`

```
type bgcctx
    sequence
    real(8) :: a
    real(8) :: b
    logical(4) :: yesOrNo
    real(8), pointer :: ptr(:)
end type
```

File: `more_sophisticated_bgc_model.f90`

```
subroutine metos3dbgc(ny, nx, nu, nb, nd, ndg, dt, q, t, y, u, b, d, dg, ctx)

#include "bgctype.h90"

    integer         :: ny           ! tracer count
    integer         :: nx           ! layer count
    integer         :: nu           ! parameter count
    integer         :: nb           ! boundary condition count
    integer         :: nd           ! domain condition count
    integer         :: ndg          ! diagnostic variable count
    real(8)         :: dt           ! ocean time step
    real(8)         :: q(nx, ny)    ! bgc model output
    real(8)         :: t            ! point in time
    real(8)         :: y(nx, ny)    ! bgc model input
    real(8)         :: u(nu)        ! parameters
    real(8)         :: b(nb)        ! boundary conditions
    real(8)         :: d(nx, nd)    ! domain conditions
    real(8)         :: dg(nx, ndg)  ! diagnostic variables
    type(bgcctx)    :: ctx          ! own bgc context

    ! your code here ...

end subroutine
```

Example 3

Good practice,

- store constants in a *module*,
- store pointers for own memory allocation, (usually not needed at all), in data type
  - use `metos3dbgcinit` to allocate memory
  - use `metos3dbgcfinal` to free memory

was what wie how warum why wofuer what for

CHAPTER 4

Metos3D tutorial

This tutorial describes how to implement, couple and run a reaction, biogeochemical, marine ecosystem, source minus sink, model, using Metos3D,

4.1 Basics

mathematical description,

CHAPTER 5

Indices and tables

- genindex
- modindex
- search