Numerical Modelling Course

@Marine Benthic Ecology 2019

Modulation of global change impact by biotic interactions, bioinvasion and parasitism

Anja Eggert

Contents

How to use the Code Generation Tool	2
How to edit/plot the simulated data	3
Some technical aspects of the model and how to run it	4

How to use the Code Generation Tool

The "Code Generation Tool" is used to create source code using the host model specific "code templates" and fills in all the tracers and processes from the formal description. Creating the source code is then very easy and straightforward.

You will need to download from https://ergom.net/ the "Code GenerationTool" executables (cgt bin linux64 1.4.2.zip (4.2 MiB) or cgt bin windows 1.4.2.zip (2.0 MiB).

Finally, you need to have three folders in your working folder:

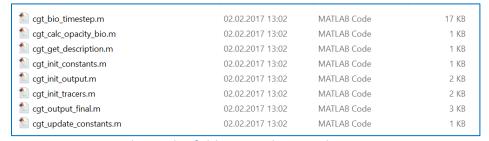
- 1. textfiles
- 2. code templates
- 3. finished_code

The folder >>textfiles<< is empty in the beginning, but fills up with txt-files once a model has been developed with the editor of the Code Generation Tool, CGT-edit.



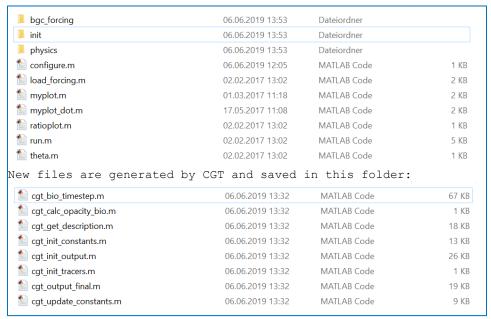
Files in the folder >>textfiles<<

The folder >>code_templates<< contains a set of Matlab files as we will run our 0D boxmodel in Matlab. Please download/copy this folder with all these files.



Files in the folder >>code_templates<<

The folder >>finished_code<< already contains folders and files. The folders contain forcing data and initial conditions of the tracers. Using the Code Generation Tool, more files are generated. In fact, these are edited files from the folder >>code_templates<<.



Folders and files in the folder >>finished_code<<

How to edit/plot the simulated data

The model runs in Matlab, i.e. simulated data are variables in the Matlab workspace. You can either use Excel or R to edit/plot the simulated data.

Solution for R:

Save all variables containing string "output_" in .mat file and read this file in R using function readMat() of package R.matlab.

```
MatLab
workspaceVars = who;
findVars = strfind(workspaceVars, 'output_');
indexVars = find(not(cellfun('isempty', findVars)));
% Now you can use workspaceVars and indexVars along with save:
save('../results/output.mat', workspaceVars{indexVars})
```

```
R
setwd("C:/.../cgt_seminar_boxmodel/results")
library(R.matlab)
data <- readMat("output.mat")</pre>
```

Solution for Excel:

You would need to export the data, e.g. using the Matlab functions writematrix() or writetable():

MatLab

```
writematrix(output temperature, '../results/output.xlsx','Sheet',1);
```

Some technical aspects of the model and how to run it

MatLab: configure.m

Here you define some basic set-ups of the model.

MatLab: cgt_init_constants.m

Here you see that the 'time-dependent constants' need to be placed in path 'bgc forcing'.

```
MatLab
forcing_din_ext = load('bgc_forcing/din_ext.txt');
% DIN concentration in Kiel fjord water pumped through KOB [mol m-3]

forcing_gam_ext = load('bgc_forcing/gam_ext.txt');
% prescribed Gammarus abundance [individuals]
```

MatLab: run.m

Here you see that physical forcing time series need to be placed in path 'physics'.

```
MatLab
%load physical forcing
forcing_temperature = load('physics/temperature.txt');
% temperature [deg_C]
```

MatLab: cgt init tracers.m

In our model, start values are not loaded from file in path 'init', but are initialized as constants (not auxiliaries).