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

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**.

auxiliaries.txt	22.06.2017 13:45	Textdokument	23 KB
celements.txt	22.06.2017 13:45	Textdokument	1 KB
constants.txt	22.06.2017 13:45	Textdokument	31 KB
elements.txt	22.06.2017 13:45	Textdokument	1 KB
modelinfos.txt	06.06.2019 13:32	Textdokument	3 KB
processes.txt	22.06.2017 13:45	Textdokument	17 KB
tracers.txt	22.06.2017 13:45	Textdokument	7 KB

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.

cgt_bio_timestep.m	02.02.2017 13:02	MATLAB Code	17 KB
cgt_calc_opacity_bio.m	02.02.2017 13:02	MATLAB Code	1 KB
cgt_get_description.m	02.02.2017 13:02	MATLAB Code	1 KB
cgt_init_constants.m	02.02.2017 13:02	MATLAB Code	1 KB
cgt_init_output.m	02.02.2017 13:02	MATLAB Code	2 KB
cgt_init_tracers.m	02.02.2017 13:02	MATLAB Code	2 KB
cgt_output_final.m	02.02.2017 13:02	MATLAB Code	3 KB
cgt_update_constants.m	02.02.2017 13:02	MATLAB Code	1 KB

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<<.

📁 bgc_forcing	06.06.2019 13:53	Dateiordner	
📁 init	06.06.2019 13:53	Dateiordner	
📁 physics	06.06.2019 13:53	Dateiordner	
configure.m	06.06.2019 12:05	MATLAB Code	1 KB
load_forcing.m	02.02.2017 13:02	MATLAB Code	2 KB
myplot.m	01.03.2017 11:18	MATLAB Code	2 KB
myplot_dot.m	17.05.2017 11:08	MATLAB Code	2 KB
ratioplot.m	02.02.2017 13:02	MATLAB Code	1 KB
run.m	02.02.2017 13:02	MATLAB Code	5 KB
theta.m	02.02.2017 13:02	MATLAB Code	1 KB

```
New files are generated by CGT and saved in this folder:
```

cgt_bio_timestep.m	06.06.2019 13:32	MATLAB Code	67 KB
cgt_calc_opacity_bio.m	06.06.2019 13:32	MATLAB Code	1 KB
cgt_get_description.m	06.06.2019 13:32	MATLAB Code	18 KB
cgt_init_constants.m	06.06.2019 13:32	MATLAB Code	13 KB
cgt_init_output.m	06.06.2019 13:32	MATLAB Code	26 KB
cgt_init_tracers.m	06.06.2019 13:32	MATLAB Code	1 KB
cgt_output_final.m	06.06.2019 13:32	MATLAB Code	19 KB
cgt_update_constants.m	06.06.2019 13:32	MATLAB Code	9 KB

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")
```

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
start_date        = datenum(2019,06,01); % initial date
end_date          = datenum(2019,08,01); % final date
repeated_runs     = 1;                   % how often the same forcing period
                                           is repeated
timestep          = 3.0/24;              % timestep [days], here every 3
                                           hours
output_interval   = 24.0/24;             % output interval [days]
```

- **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).

```MatLab
% some need to be loaded from files

% others are initialized as constant

  tracer_scalar_t_din           = t_din_0;
  tracer_scalar_t_nfu           = t_nfu_0;
```