Table of Contents

SOCOL v3

EULER cluster

Before you start running SOCOL, make yourself familiar with Euler, ETHZ's Linux custer:
http://brutuswiki.ethz.ch/brutus/Getting_started_with_Euler

We belong to a shareholder group, the Euler Climate group. Find some more information on this collaboration here.

Documentation

Now that you know the basic stuff about Euler, we come to the second step: RTFM!!!

Slides with a lot of tips & tricks around SOCOL from a SOCOL hands-on session in October 2019:
SOCOL_workshop_Oct2019.pdf

ECHAM5 GCM

- Getting started with ECHAM5 (how to implement new tracers, additional output, etc.)
- ECHAM5 namelists
- ECHAM5 output variables
- The atmospheric general circulation model ECHAM 5. PART I: Model description
- The atmospheric general circulation model ECHAM5 Part II: Sensitivity of simulated climate to horizontal and vertical resolution
- collection of technical manuals, partly only drafts
- hand written call tree of the physics routine

MEZON Chemistry

Unfortunately, we haven´t found a technical documentation of the MEZON chemistry scheme so far. If you have one, please put it here.

- Call tree chemistry module MEZON
- List of chemical reactions
- List of tracers
- SOCOL specific namelist

Mixed-Layer Ocean

SOCOL can be run with a mixed-layer ocean (MLO), which considers a water column with a fixed depth of 50m and a thermodynamic sea ice module. This model configuration does not consider active ocean dynamics, but captures the shorter term response of the ocean s surface temperature to the atmosphere. The heat budget of the oceanic mixed-layer is described by

```
Cm*dTm/dt = H - Q
```

where Tm denotes the temperature of the mixed-layer, which substitutes the sea surface temperature in the model version without ocean. Cm and H describe the effective heat capacity of the mixed-layer and the net heat flux at the atmosphere ocean surface, respectively. The oceanic heat transport Q (also called flux correction term) is intended to compensate for the energy transport by ocean currents, which is not simulated by the MLO model itself (Steppuhn et al., 2006).

The respective source code (just in case your model version does not come with the correct MLO source files) as well as some instructions on how to run the MLO can be found here:
https://wiki.iac.ethz.ch/pub/Chemie/SOCOLv3/mlo.tar

Get the model code

to be updated

Compile

Set stacksize to unlimited in case of intel compiler

⚠ **IMPORTANT**: The model will crash with segmentation fault, if the stacksize is too small. Set it to unlimited when using intel compiler.

- For bash shell

```
ulimit -s unlimited
```

- For tcsh shell

```
limit stacksize unlimited
```

It's recommended to set the stacksize also in the run script (see further down).

On ETH cluster Euler

⚠ **IMPORTANT**: Do not use intel/11.0.081. The model will crash with segmentation fault.

- Load compiler module

```
module load intel
```

- Load openMPI module

```
module load open_mpi
```

- Load netcdf module

```
module load netcdf
```

- **IMPORTANT** You have to load first the compiler and afterwards the modules for openmpi and netcdf. Otherwise you don't get the openmpi and netcdf versions which correspond to your compiler !! If you like you can load a modules at once with

```
module load intel open_mpi netcdf
```

- Run make

```
make
```

Run the model

On ETH cluster Euler

- Run script, for example **/home/beyerleu/echam5/run_00040.run-for-brutus**
- Load the right modules !

```
module load intel netcdf open_mpi
```

- Submit the job to the queuing system with

```
bsub < run_00040.run-for-brutus
```

- What's important in **run_00040.run-for-brutus**
  - Define parameters for the queuing system

```
#BSUB -J run_00040
#BSUB -n 4
#BSUB -W 0:59
```

  - **IMPORTANT**: The requested cores, number after **-n** has to be equal the the value of **NCPUS** further down.
  - Set the wallclock time **-W** to something higher than your model will run. There are 1 hour and 7 hour and 24 hours(?) queues.
  - Load the correct modules and set stacksize to unlimited

```
# load modules
source /etc/profile.d/modules.sh
module purge
module load intel/10.1.018 netcdf open_mpi/1.3.3
module list

# set stacksize to unlimted
ulimit -s unlimited
```

  - Remark: The stacksize on brutus has also to be increased on the nodes (`/etc/security/limits.conf`). This has been already be done be the brutus administrators.
  - Set NCPUS, NPROCA, NPROCB, for example

```
NCPUS=2
NPROCA=2
NPROCB=1
```

  - Set correct WORKDIR, for example

```
WORKDIR=/cluster/work/uwis/$USER/echam5
```

  - Call model

```
ompirun $MODEL
```

Chemical Preprocessor for SOCOLv3

The chemical preprocessor sets up the chemical equation system and creates the respective model code.

Members of the SOCOL Git group can download the preprocessor package (Fortran code and run scripts) from the IAC GitLab server:

```
git clone https://git.iac.ethz.ch/stenkea/socolv3_chemical_preprocessor.git
```

How to run Chemical Code Generator (CEDR)

- go to the **work** directory
- edit the reaction table (Retab.d, Retab_tropo.d)
- edit the script **cedr.csh**: choose between standard SOCOL chemistry (**STRATO**) or more comprehensive tropospheric chemistry (**TROPO**):

```
echo "Chemical code generator"
ifort -o chegen.x CHEGEN.f
chegen.x TROPO   # <- replace TROPO with STRATO if you want the standard SOCOL chemi
```

- run cedr.csh

- that's it, the output files are in the **output** directory
- copy the new source files to the respective source code directories **/src** and **/modules**
- compile the model

Notes

- new chemical species (= tracers) have to be included manually into mo_socol_tracers.f90
- same for photolysis rates (mo_socol_sun.f90)

Preparation of photolysis rates (LUT) for SOCOLv3

SOCOLv3 reads pre-calculated photolysis rates as a function of O2 and O3 columns (modules/mo_socol_sun.f90). The actual photolysis rates are calculated online by bi-linear interpolation of the tabulated data onto actual values of total oxygen (XO2) and ozone (XO3) above the respective grid point (src/socol_dis.f90).

Members of the SOCOL Git group can download the LUT package (Fortran code and run scripts) from the IAC GitLab server:

```
git clone https://git.iac.ethz.ch/stenkea/socolv3_lut_code.git
```

Step-by-step introduction on how to use git can be found here.

Instructions on the LUT code are given here: How to lut

List of model simulations

Google spreadsheet

- Google spreadsheet SOCOL simulations

SOCOL vs3

The model output from the CCMI model simulations is available at BADC (you might have to request an account there).

EXPNO	RES	YEARS	DATA	NOTE
00050	T31L39	1975-2005	/net/hydro/chemie/stenkea/afterburner/run00050 and /net/hydro/chemie/stenkea/brutus/raw/run00050	reference run SOCOL v3 (Stenke et al., 2013)
00060	T42L39	1975-2005	/net/hydro/chemie/stenkea/afterburner/run00060 and /net/hydro/chemie/stenkea/brutus/raw/run00060	reference run SOCOL v3 (Stenke et al., 2013)
00360	T42L90	1975-2000	/net/hydro/chemie/stenkea/brutus/raw/run00360	reference run SOCOL v3

Edit | Attach | Print version | History: %REVISIONS% | Backlinks | Raw View | More topic actions
Topic revision: r6 - 12 Apr 2022 - 17:55:14 - AndreaStenke

- ☐ Chemie

No permission to view Chemie.WebLeftBar

-
-
-

Copyright © by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding Wiki? Send feedback