A python-based processing chain facilitating automated COSMO simulations and post-processing

Michael Jähn, David Ochsner, Gerrit Kuhlmann,
Jean-Matthieu Haussaire, Pavle Arsenovic, Qing Mu, Dominik Brunner

Empa, Atmospheric Modelling and Remote Sensing

COSMO User Workshop, ETH Zurich

21 January 2019

Empa
Materials Science and Technology

Empa
Materials Science and Technology

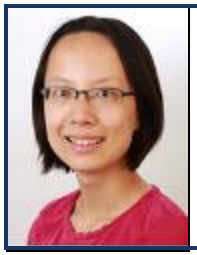
- COSMO users of our group:

GERRIT KUHLMANN
COSMO-GHG

JEAN-MATTHIEU HAUSSAIRE
COSMO-GHG

PAVLE ARSENOVIC
COSMO-GHG

MICHAEL JÄHN
COSMO-GHG



QING MU
COSMO-ART

DAVID OCHSNER
Maintenance, Testing, Documentation

Different users, COSMO versions, purposes, projects etc.

Goal: Create a tool that unifies and automates data pre-processing, int2lm, COSMO runs, post-processing

COSMO Processing Chain

Empa
Materials Science and Technology

- Originally developed using Shell scripts
  - Not very flexible
  - Cumbersome maintenance
- Translation to Python 3
- Version control with git and own GitLab server
  - Master and own working version
  - Standard git workflow (forks, branches, issues, merge requests …)
- Anaconda3 environment
  - Installation with EasyBuild on Piz Daint (CSCS)
  - Additional Python modules + own «amrs» package

Basic Features

- Easy creation of own COSMO cases
  - Subfolder as case name
  - One configuration file
  - Flexible namelist files
  - Tracer information from .csv files
- Automatic restart runs
- Nested runs possible
- Thorough documentation
- Clear output folder structure
- Compressed netCDF output files with pre-defined tolerances

Configuration File

Empa
Materials Science and Technology

- cosmo_processing_chain/cases/example/config.py

```
import os

# ORGANIZATIONAL ================================================================ #
compute_host = 'daint'
compute_queue = 'debug' # 'normal'
compute_account = 's862'

# case name = pathname in cases/
path = os.path.realpath(__file__)
casename = os.path.basename(os.path.dirname(path))

# Restart
restart_step = 24

# INPUT ========================================================================= #
# METEO ---------------------------------------------------------------------- #
meteo_dir = os.path.join(input_root, 'meteo')
meteo_prefix = "laf"
meteo_inc = 1

# EMISSIONS ------------------------------------------------------------------ #
# anthropogenic emissions pre-processed for mother and nested domain
emissions_dir = os.path.join(input_root, 'emissions_coarse')
emis_gridname = "CO2_CO_NOX_Berlin-coarse_"
```

Namelist Files

Empa
Materials Science and Technology

- cosmo_processing_chain/cases/example/cosmo_INPUT_ORG.cfg

```
&LMGRID
! Berlin full domain
  ie_tot        = 70,
  je_tot        = 60,
  ke_tot        = 60,
  pollon        = -170,  ! origin lon: 10.0
  pollat        =  43,   ! origin lat: 47.0
  dlon          = 0.1,
  dlat          = 0.1,
  startlat_tot  =  2.5,
  startlon_tot  = -1.4,
/
&RUNCTL
  dt            = 10.0,
  hstart        = {cfg.hstart},
  hstop         = {cfg.hstop},
  ydate_ini     = '{cfg.inidate_yyyymmddhh}',
  nprocx        = {cfg.cosmo_np_x},
  nprocy        = {cfg.cosmo_np_y},
  nprocio       = 0,
  hincmxt       = 1.0,
```

Empa
Materials Science and Technology

COSMO processing chain

Command line help:

```
$ python run_chain.py -h
```

Run examples:

```
$ python run_chain.py example 2015-01-01 0 24
$ python run_chain.py example_cosmoart_mother example_cosmoart_nested 2015-06-26 0 12
$ python run_chain.py example_oae 2015-01-01 0 24 -j meteo icbc oae modis biofluxes int2lm post_int2lm cosmo pos
```

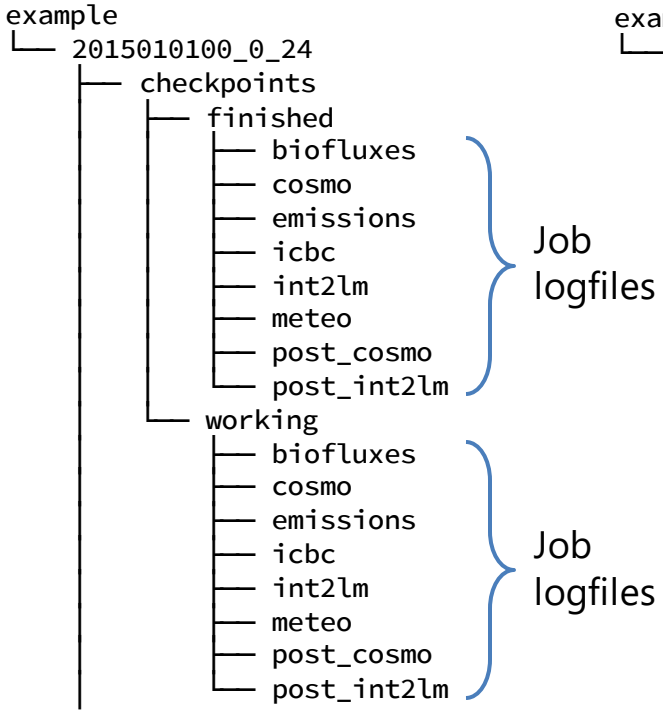
Requirements: amrs

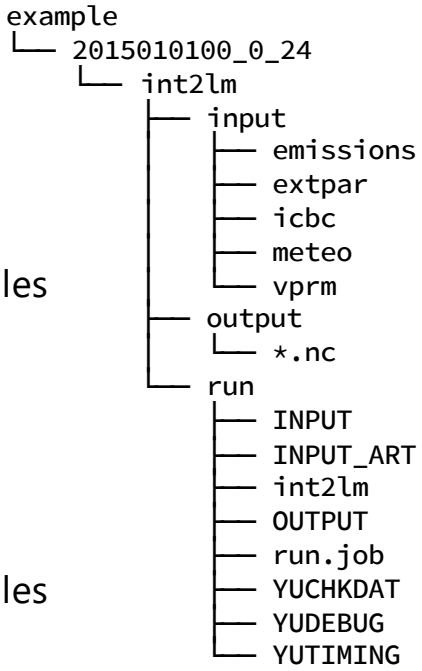
Build documentation locally (make sure you have sphinx installed):

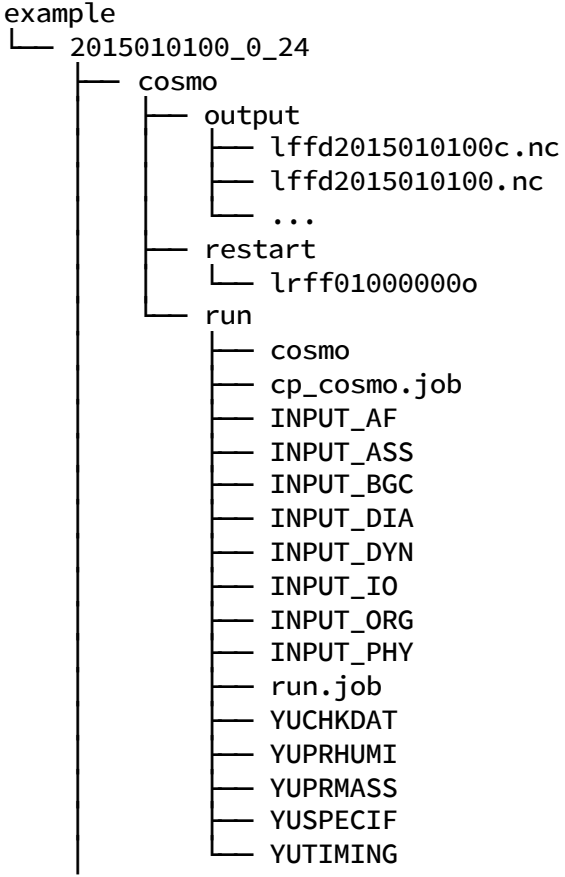
```
$ cd doc/
$ make html
$ make text
```

If all else fails: read documentation source files at docs/source/*.rst

Output Structure







Lossy Compression of netCDF4 Output Files

Empa
Materials Science and Technology

- Converting 32-bit floating point variables to 16-bit signed integer
- Using netCDF4 attributes `add_offset` and `scale_factor`
- Compression ratio 2-3
- Definition of own tolerances to conserve required precision of output fields:

```
# name,     rtol,   atol
  U,        0.0,    0.001
  V,        0.0,    0.001
  W,        0.0,    0.001
  T,        0.0,    0.01
  P,        0.0,    0.1
  CO2_BG,   0.0,    1e-8
  …
```

Sharing our Code

- `cosmo_processing_chain` and `amrs` maintained on Empa's GitLab server
- Question: Which platform to share code?
- Details to be discussed

Thank you!