netherlands eScience center

The Oceanographic Multipurpose Software Environment: Introduction and Applications

Inti Pelupessy[1], Gijs van den Oord[1], Ben van Werkhoven[1], Maria Chertova[1], Merijn Verstraaten[1], Henk Dijkstra[2], Fredrik Jansson[3], Arjen van Elteren[4], Simon Portegies Zwart[4] (and many others!)

[1]Netherlands eScience Center, [2]IMAU, Universiteit Utrecht, [3]CWI, [4]Leiden Observatory

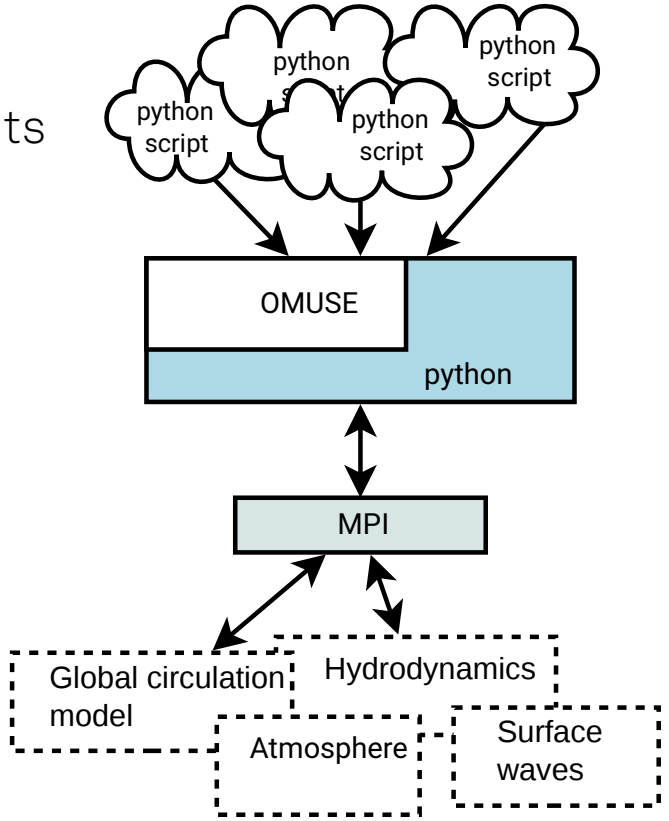
5th Workshop on Coupling Technologies for Earth System models,
21-24 september 2020

What is OMUSE?

→ Oceanographic Multi-PUrpose Software Environment
→ OMUSE is a Python environment for numerical experiments in oceanography and other climate sciences
→ based on *AMUSE*

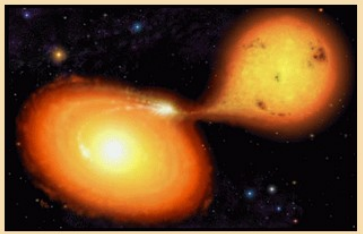
Goals:

- provide a homogeneous enviroment to run *community* codes
- enable new code couplings and interactions between components
- facilitate multi-physics and multi-scale simulations



a short history of AMUSE & OMUSE

- AMUSE has its origins in the astrophysical MODEST community



MODEST

Modeling and Observing DEnse STellar systems

- developed, with funding from NOVA, NWO and the NLeSC, at Leiden Observatory

- actively being used by 15+ groups worldwide, 60+ publications, 8+ theses

- 2014 – 2016: NLeSC project @IMAU (Pelupessy, van Werkhoven) to generalize this approach → OMUSE

- 2017- present: OMUSE being used in research (see part II), extension with new codes and general improvements in useability and installation procedures.

'Hello Ocean'

“imports”

```
from omuse.units import units
from omuse.community.qgmodel.interface import QGmodel
from amuse.io import read_set_from_file
```

“initial conditions”

```
input=read_set_from_file('initial_condition')
```

“instantiate code”

```
code=QGmodel()
```

“initialize model”

```
code.parameters.dt=0.5 | units.hour
code.grid.psi=input.psi
```

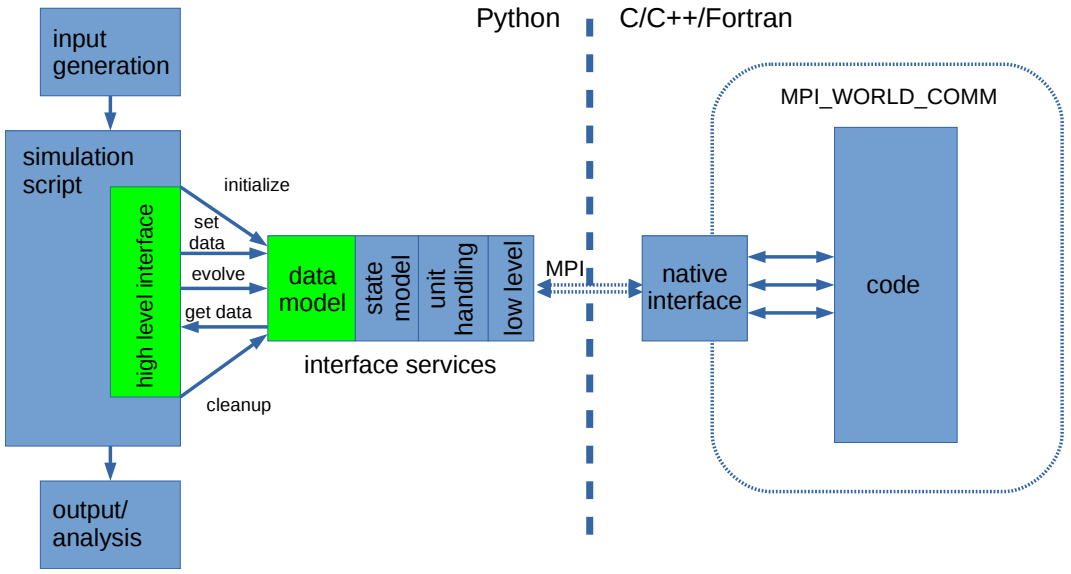
“evolve”

```
code.evolve_model(1.| units.day)
```

“analysis”

```
print code.grid.psi.max().in_(units.Sv/units.km)
```

The OMUSE interface: a bird's eye view



– OMUSE is based on remote code interfaces

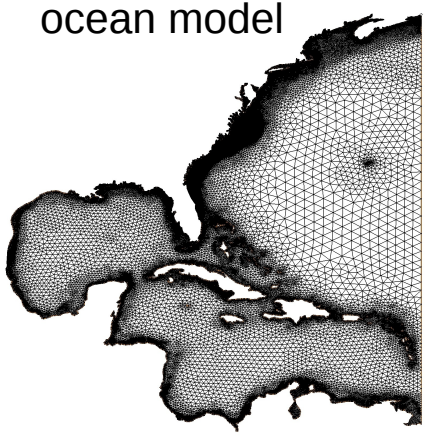
– codes run in parallel & calls can be asynchronous

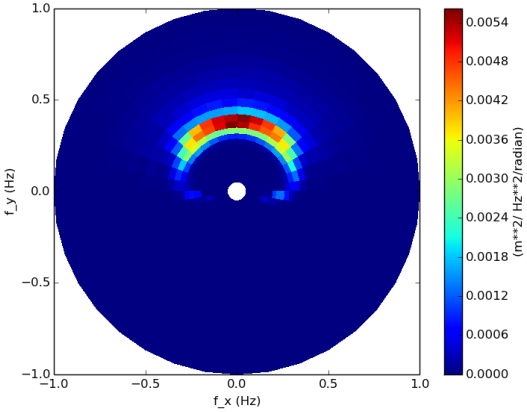
– OMUSE provides a number of *interface services:* unit handling, data structures, code state handling etc..

– the user is presented with uniform high level interfaces, which integrate into a scientific python workflow

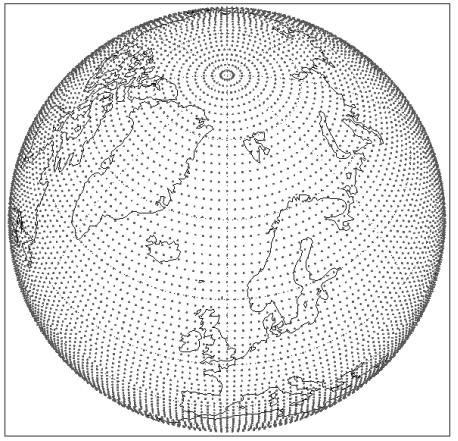
current codes in OMUSE

ADCIRC regional ocean model

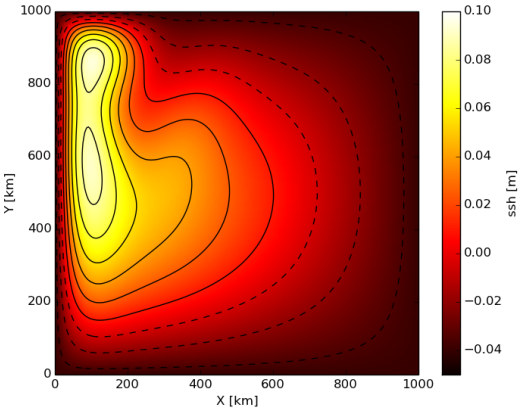




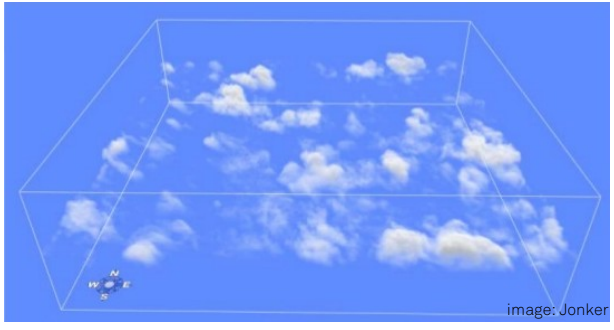
SWAN wave transport



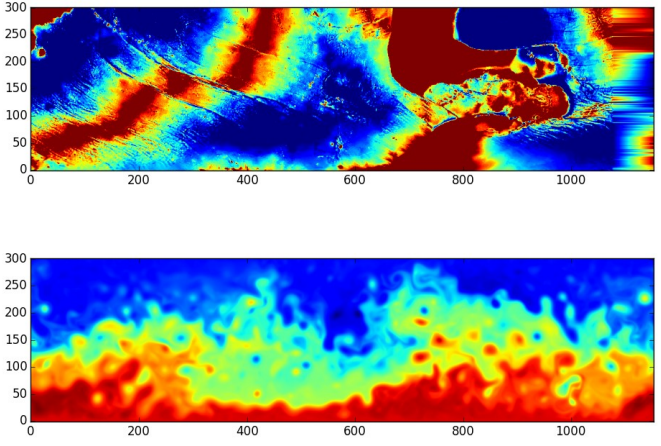
openIFS global atmosphere



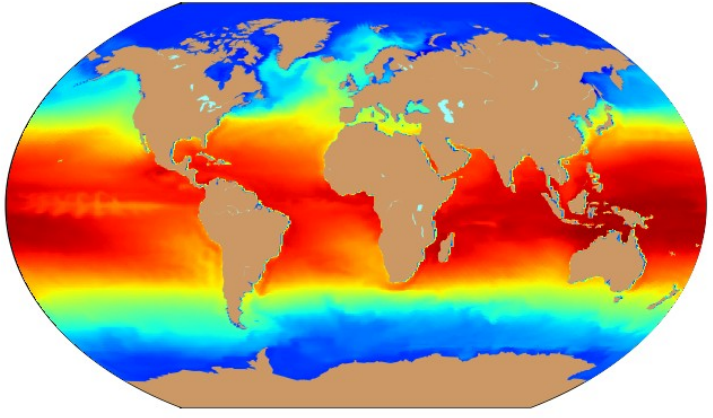
QGmodel



DALES cloud resolving model

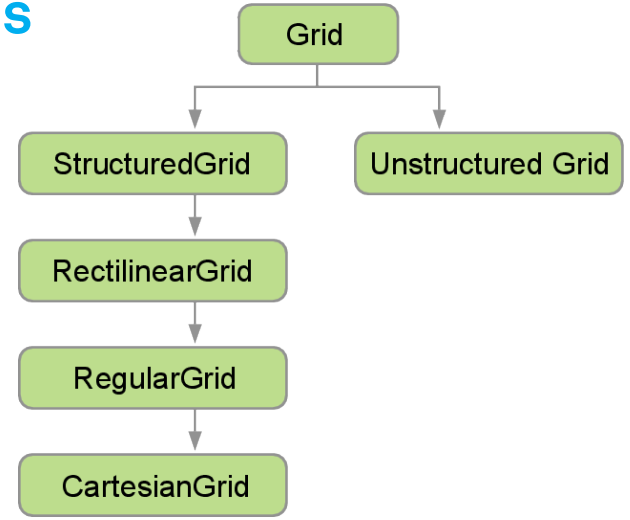


QGCM



POP global ocean model

OMUSE data structures

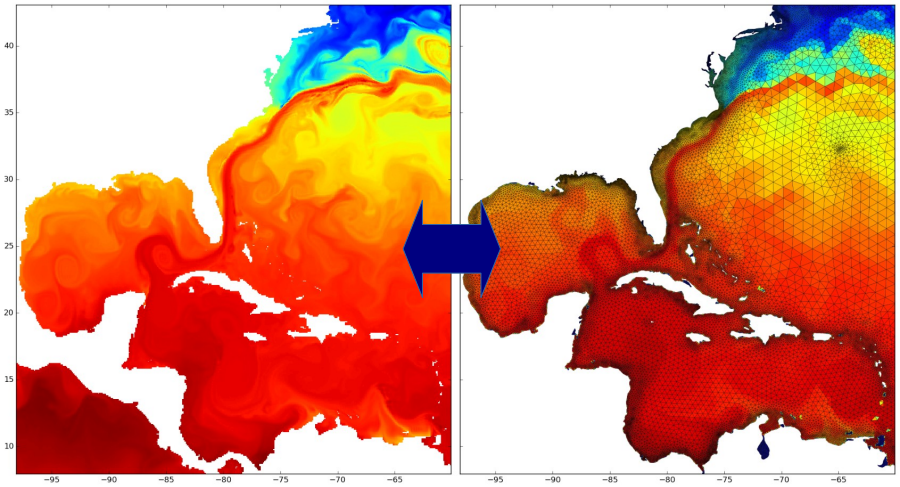


– high level interface uses grid data structures

– grids have domain specific attributes

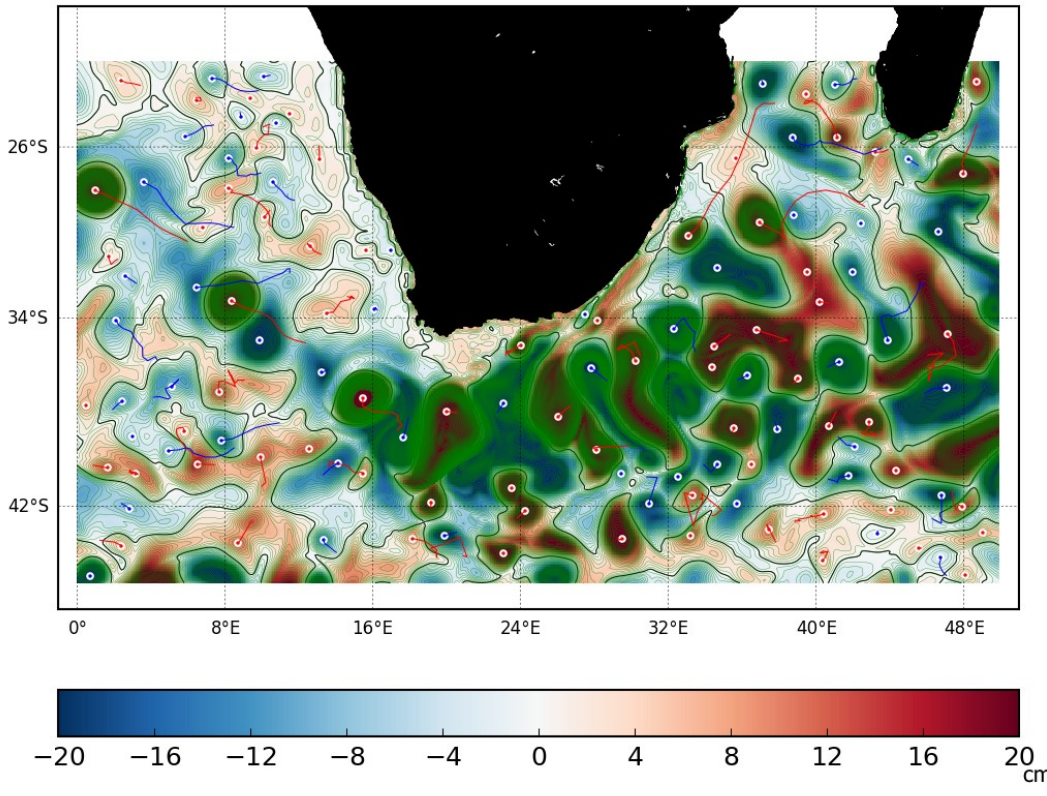
– OMUSE improved grid support:

  – different grid types
  – add grid remapping channels
  – add grid (functional) transform
  – (some) grid generation routines
  – improved support for large parameter sets and files

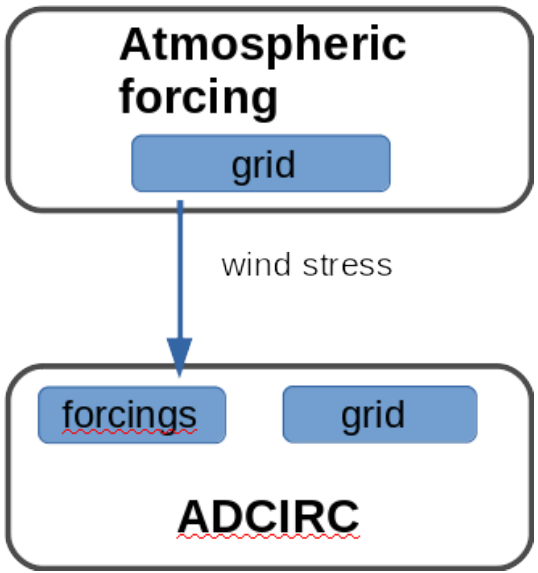


What can you do with OMUSE?

- simplify model setup and runs,
- scripting:
  - event detection,
  - stopping conditions
- 'online' data analysis
- ensemble simulations:
  - parameters searches
  - optimizations (e.g. MCMC)
  - data assimilation
- model comparison: running problems with different codes and methods
- coupling different codes to construct new solvers

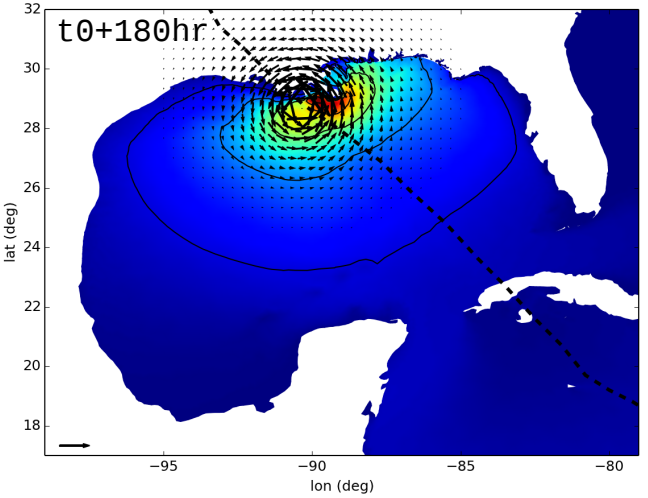


Code couplings with AMUSE

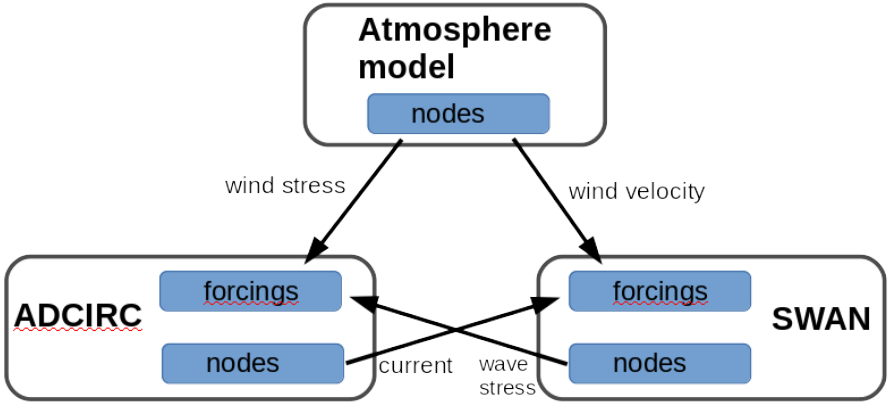



```
from omuse.community.adcirc.interface import Adcirc
from omuse.community.era5.interface import ERA5

adcirc = Adcirc()
meteo  = ERA5(variables=[ "10m_u_component_of_wind",
                          "10m_v_component_of_wind",
                          start_datetime=start_date )

channel = meteo.grid.new_remapping_channel_to(adcirc.forcings,
                                              interpolating_lonlat_remapper)

while adcirc.model_time<tend:
    meteo.evolve_model( adcirc.model_time + dt )
    channel.copy_attributes([ "_10m_u_component_of_wind",
                              "_10m_v_component_of_wind"],
                              target_names=["wind_vx","wind_vy"])
    adcirc.evolve_model( adcirc.model_time + dt )
```

Hurricane Gustav with a coupled hydrodynamic/ wave model



(Pelupessy+ 2017)




```
channel1=hurricane.grid.new_channel_to( swan.forcings )
channel2=hurricane.grid.new_channel_to( adcirc.forcings )
channel3=adcirc.nodes.new_channel_to( swan.forcings )
channel4=swan.nodes.new_channel_to( adcirc.forcings )
while time<tend:
    hurricane.evolve_model(time+dt/2)
    channel1.copy_attributes(["vx","vy"])
    channel2.copy_attributes(["tau_x","tau_y"])
    adcirc.evolve_model(time+dt/2)
    swan.evolve_model(time+dt/2)
    channel3.copy_attributes(["current_vx","current_vy"])
    channel4.copy_attributes(["wave_tau_x","wave_tau_y"])
    time+=dt
```

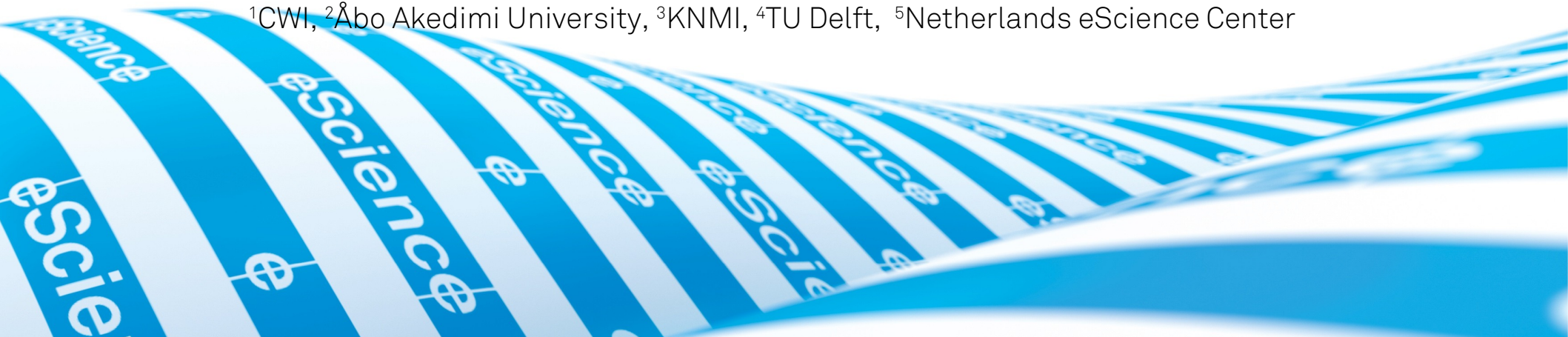
part II: Applications

netherlands eScience center

Regional superparameterization in a Global Circulation Model using Large Eddy Simulations

Fredrik Jansson[1], Daan Crommelin[1], Joanna Grönqvist[2], Pier Siebesma[3,4], Maria Chertova[5], Gijs van den Oord[5], Inti Pelupessy[5]

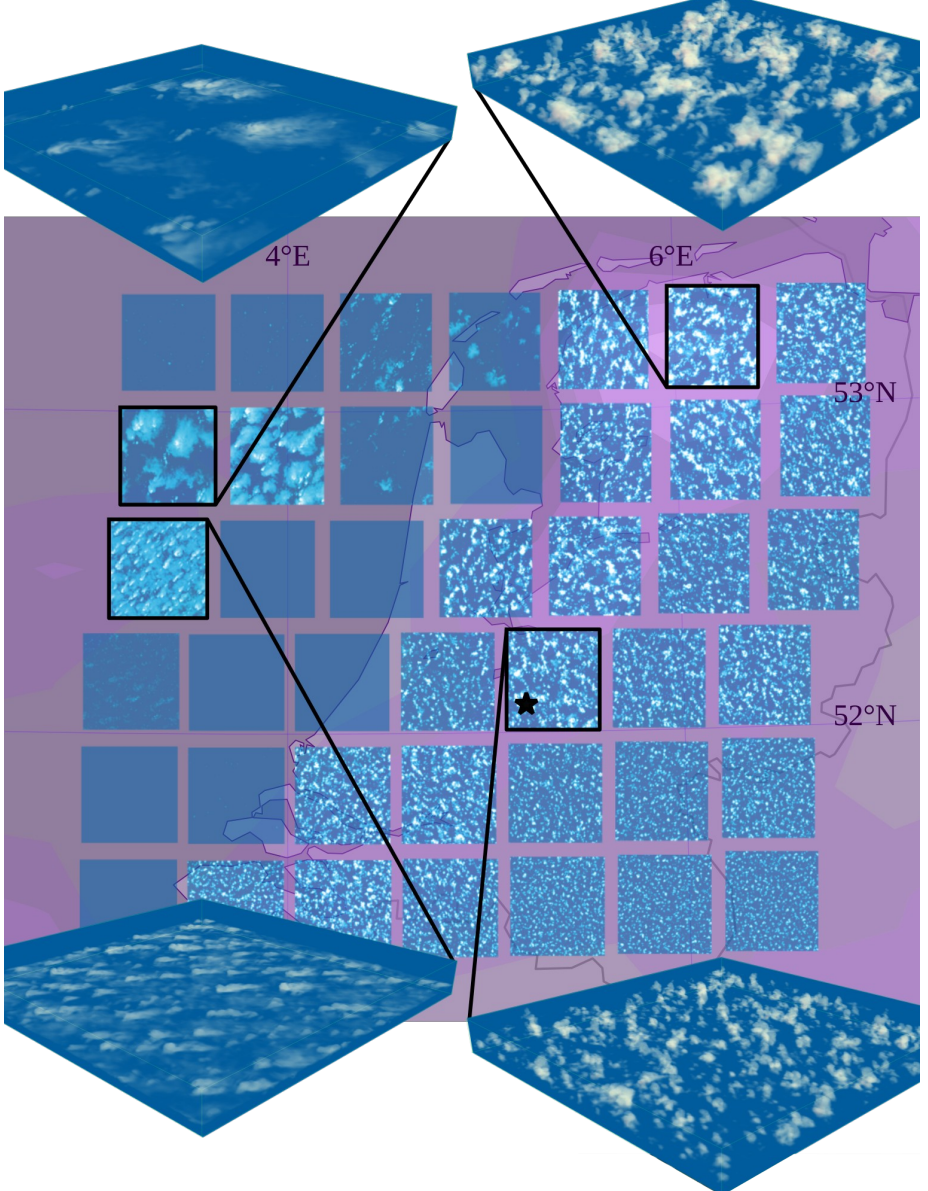
[1]CWI, [2]Åbo Akedimi University, [3]KNMI, [4]TU Delft, [5]Netherlands eScience Center



DALES – OIFS coupling

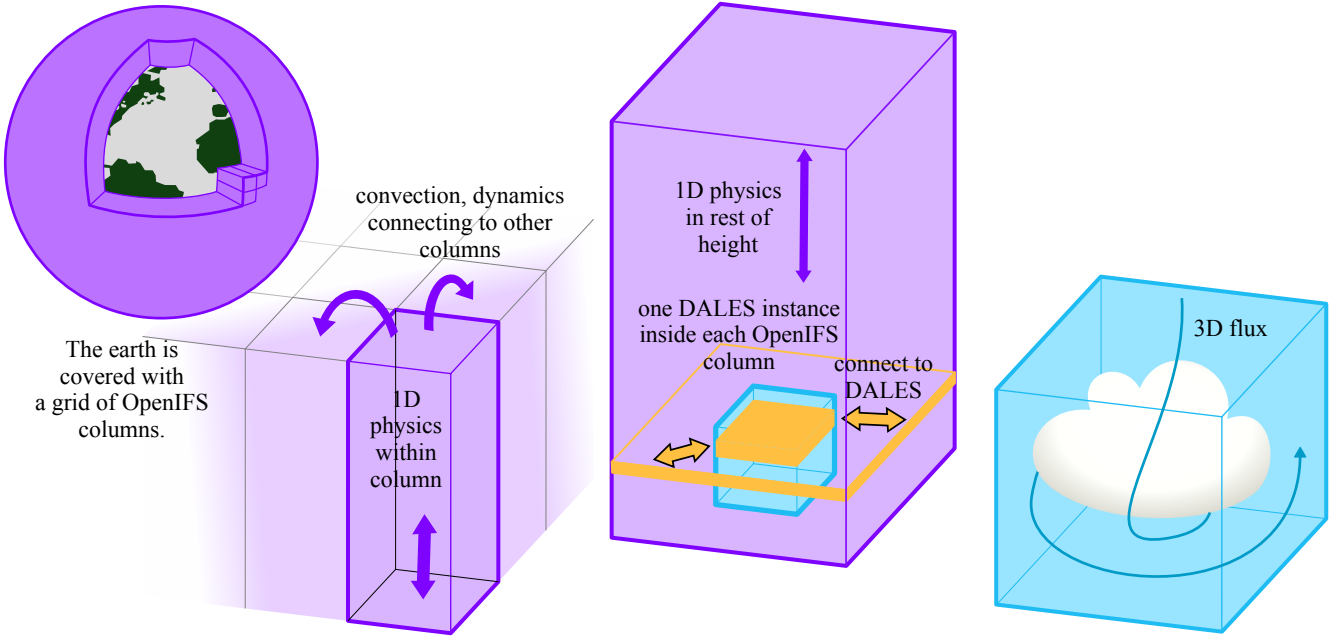
DALES: Dutch Atmospheric Large Eddy Simulation
OpenIFS: community version of ECMWF operational IFS model

– superparametrization
– embedding of cloud resolving in global atmosphere model
– two way coupling
– N models in single global instance

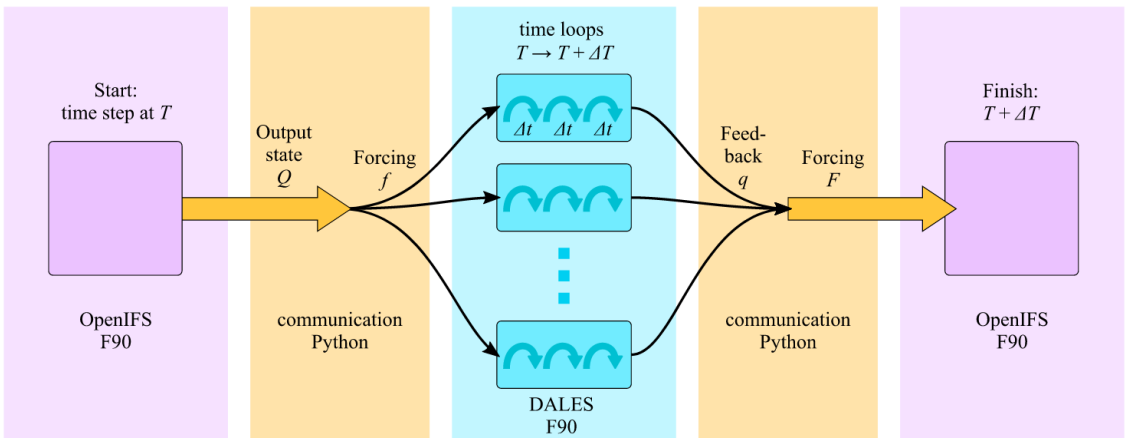


Jansson et al. 2019, JAMES 11, 2958
van den Oord et al. 2020, SoftwareX, submitted

Superparametrization: model overview

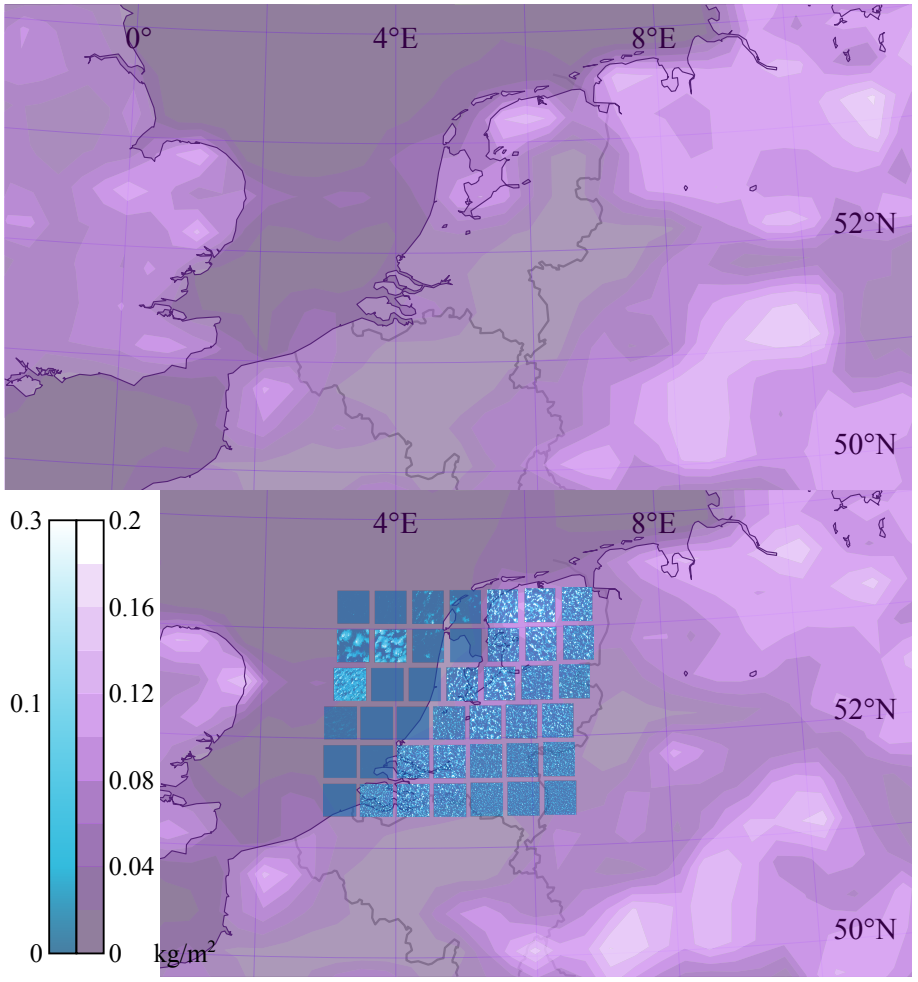


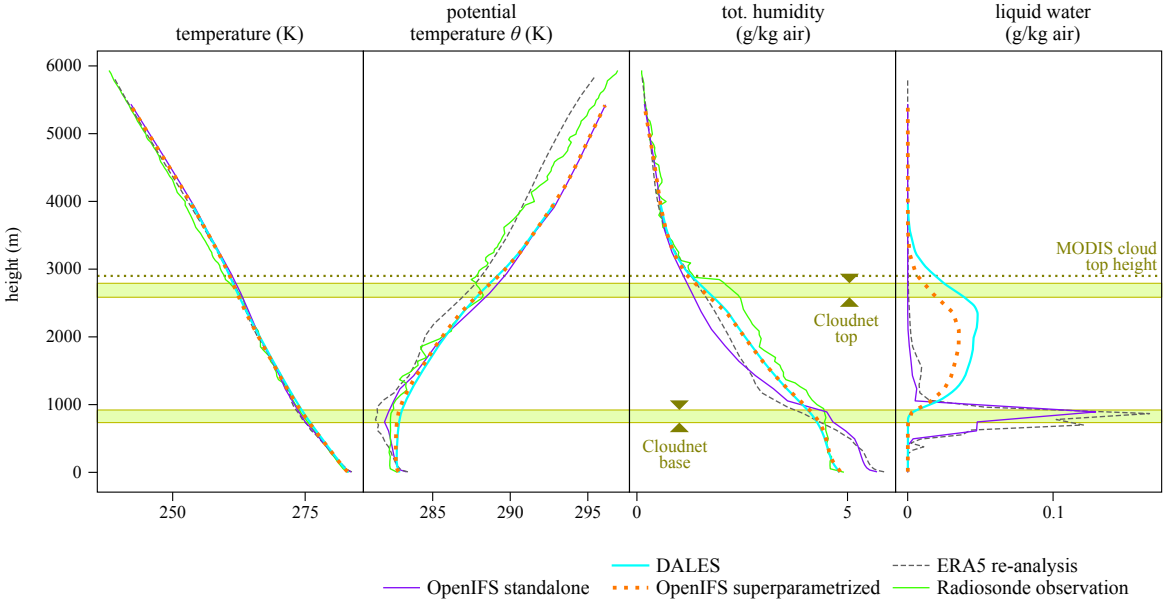
timestepping



purple: OpenIFS
yellow: model coupling
cyan: DALES

Superparametrization results: summary





- superparametrization improves predictions for cloud cover, water content and cloud layer heights
- probably due to improved convective mixing
- experiments have shown problems with advection of clouds, which may remedied (paper in preparation)

netherlands eScience center

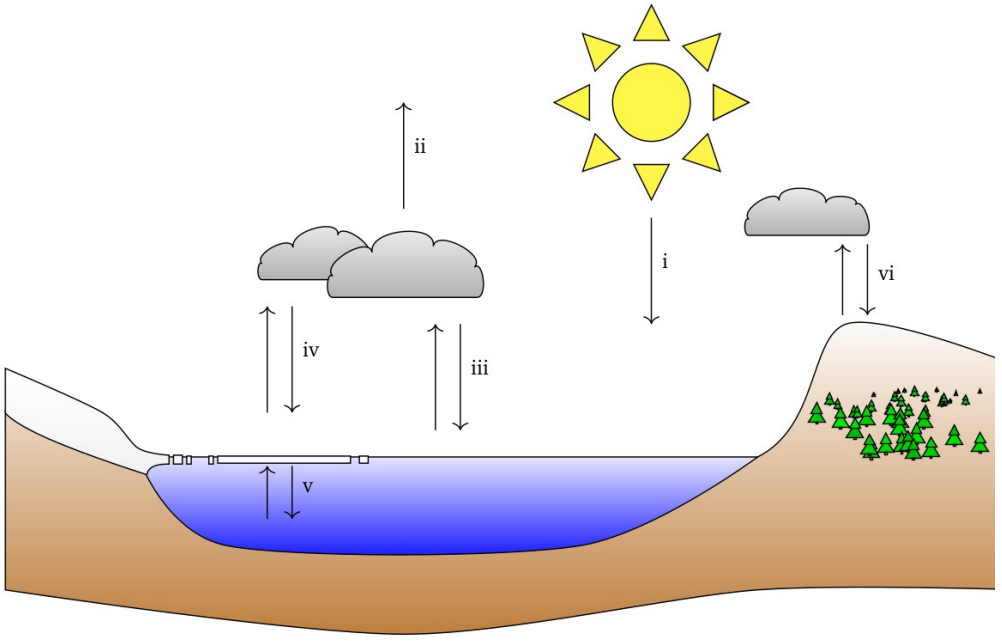
SMCM: Stochastic Multi-scale Climate Models

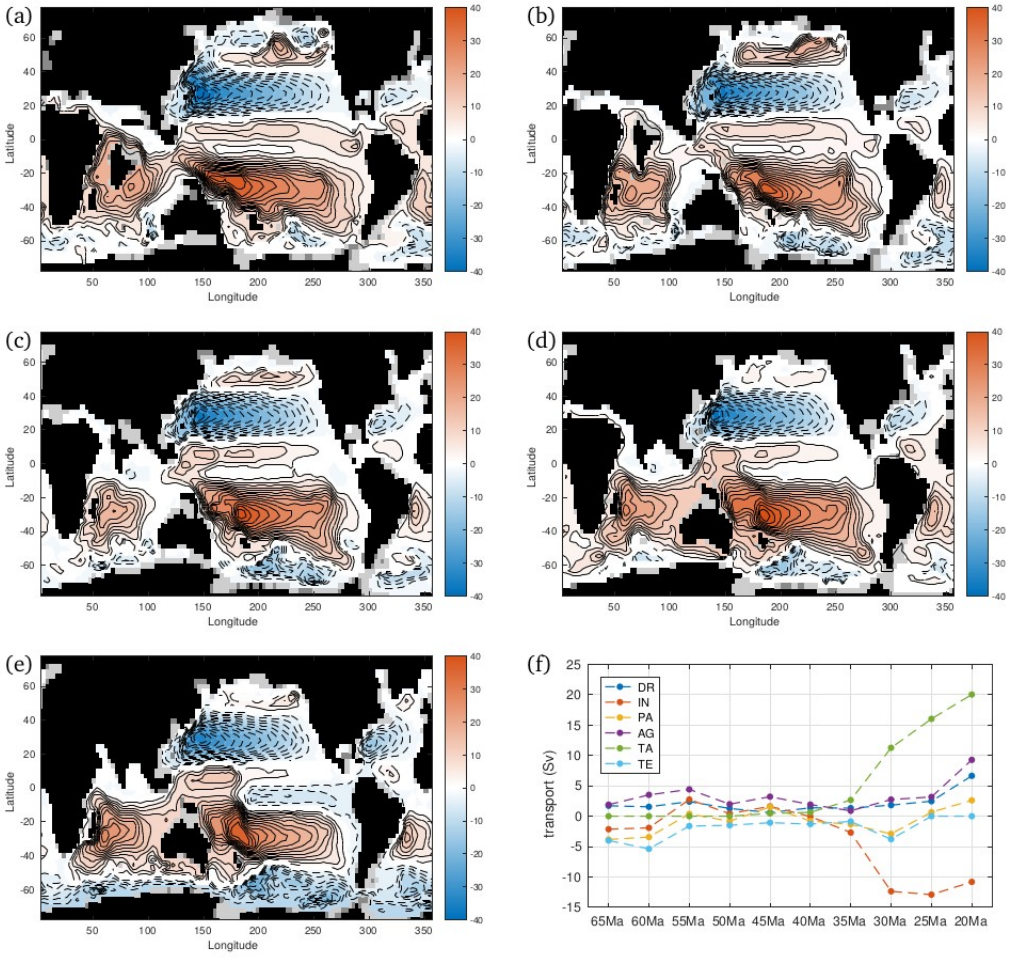
Erik Mulder[1], Sven Baars[1], Fred Wubs[1], Henk Dijkstra[2],
Maria Chertova[3], Merijn Verstraaten[3], Inti Pelupessy[3]

[1]Bernoulli Instituut, RUG, [2]IMAU, Utrecht University [3]Netherlands eScience Center

I-EMIC: An Implicit Earth System Model of Intermediate Complexity

- primitive equation ocean model
- diffuse transport of atmospheric heat and moisture
- evaporation and precipitation
- sea ice formation
- albedo adjustment



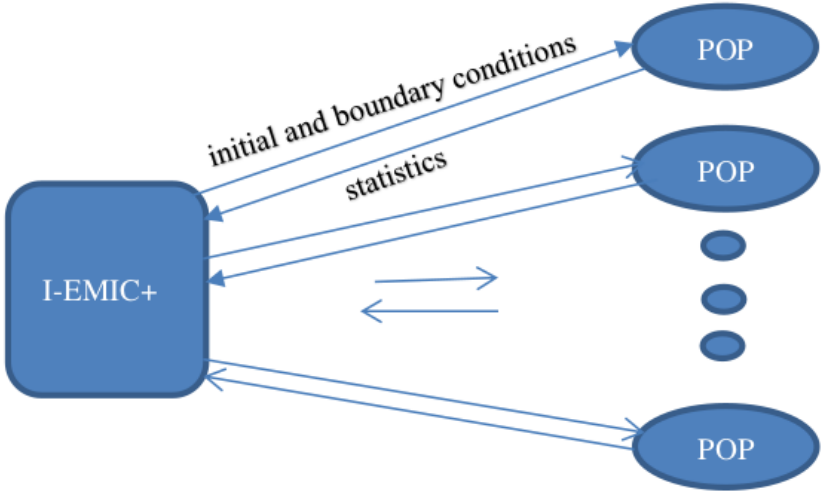


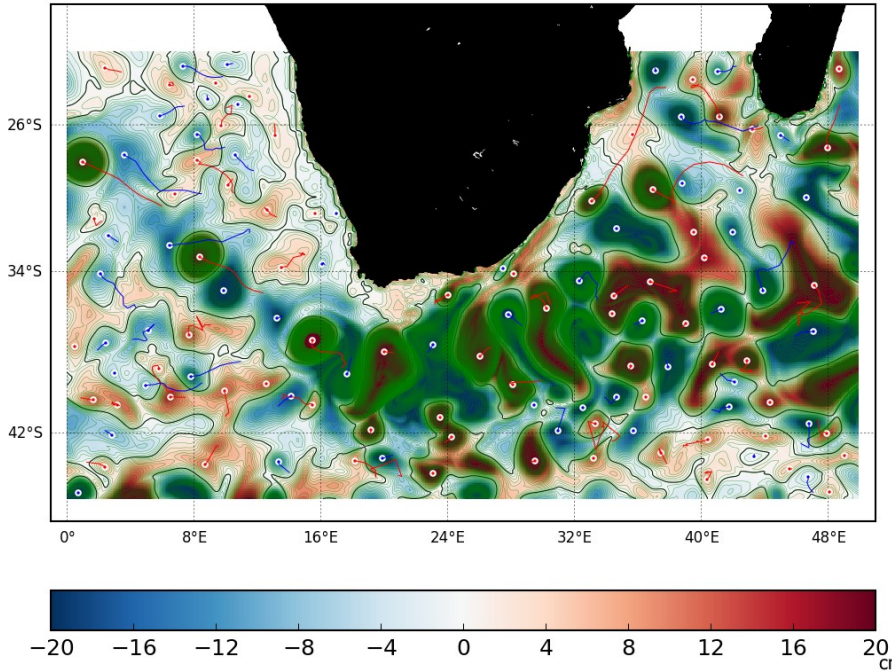
(Steady state barotropic stream function for the global ocean model from 60 to 20 Ma, Mulder 2019)

Stochastic Multiscale Climate Modelling project

Implicit earth system models like I-EMIC allow in-depth study of the climatic state transitions and long term projections for e.g. paleo-earth modelling. However, their ocean component cannot be run at eddy-permitting resolution, and do not properly account for ocean eddy transport.

Solution: Couple I-EMIC with an eddy resolving model (e.g. POP)





SMCM: development highlights

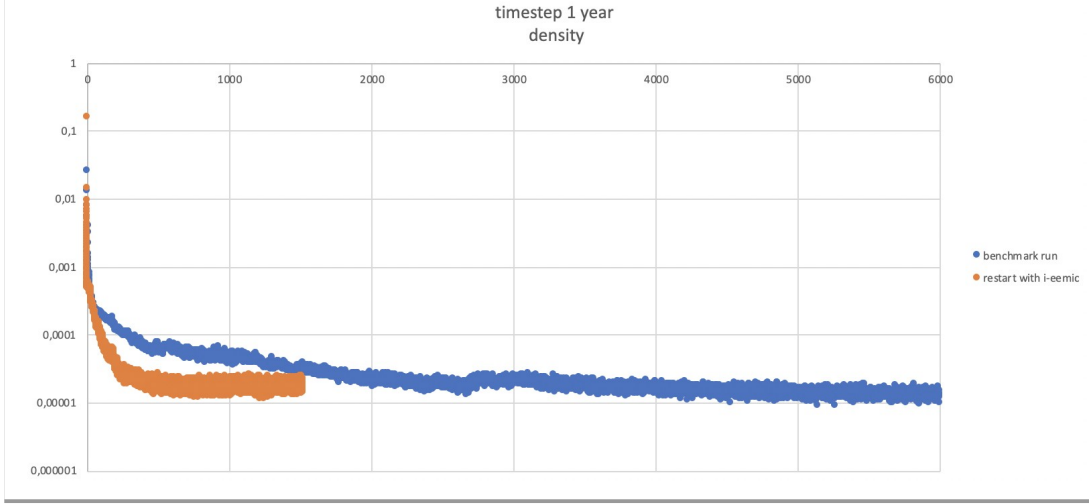
– IEMIC published on github, development on user friendliness; parameters, documentation and installation issues,

– work on performance (new HyMLS preconditioner),

– OMUSE IEMIC interface, and development of `RemoteStateVector` class (wraps mixed C++/ Fortran manipulations),

– Python implementations of continuation and timestepping

– example application:
acceleration of POP spin up
(slow due to meridional overturning circulation)



netherlands eScience center

IVM VU
Institute for Environmental Studies

NWO
Netherlands Organisation for Scientific Research

Deltares
Enabling Delta Life

MOSAIC:
MOdelling Sea level and Inundation for Cyclones

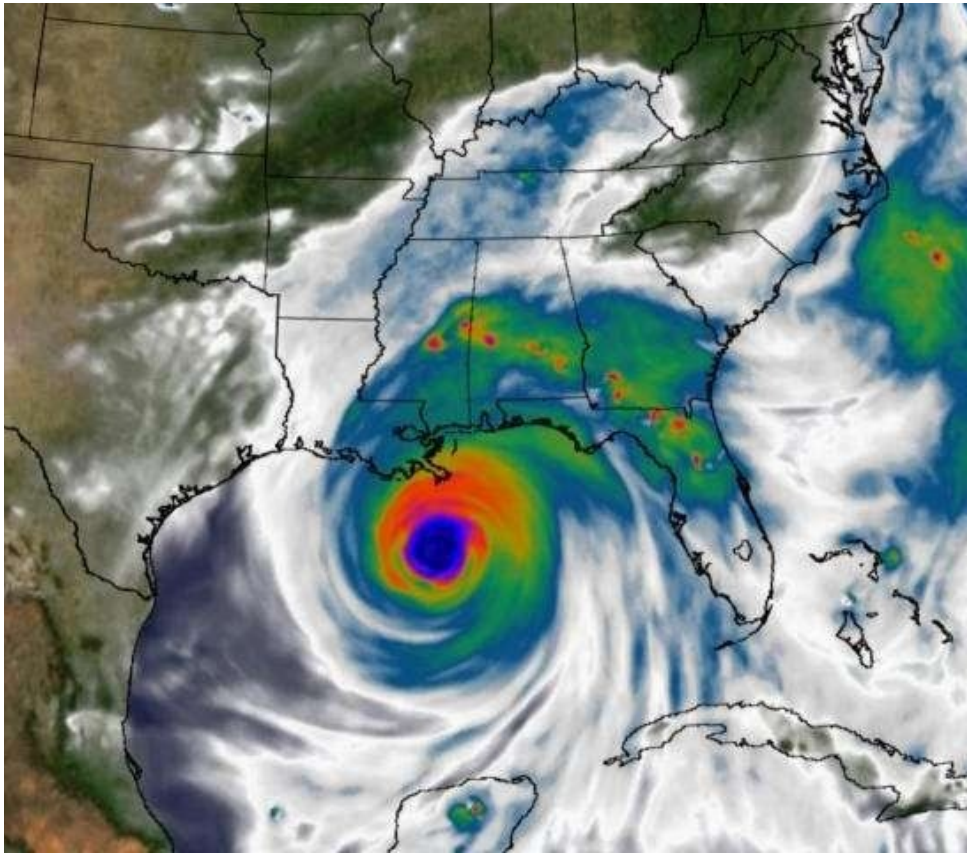
Sanne Muis[1], Job Dullaart[1], Nadia Bloemendaal[1],
Philip Ward[1], Maria Chertova[2], Inti Pelupessy[2]

[1]IVM, VU, [2]Netherlands eScience Center

MOSAIC: MOdelling Sea level and Inundation for Cyclones

Tropical Cyclones(TC):

- One of the deadliest and costliest natural hazards
- Affect coastal areas across the globe
- Storm surge, waves, wind, precipitation

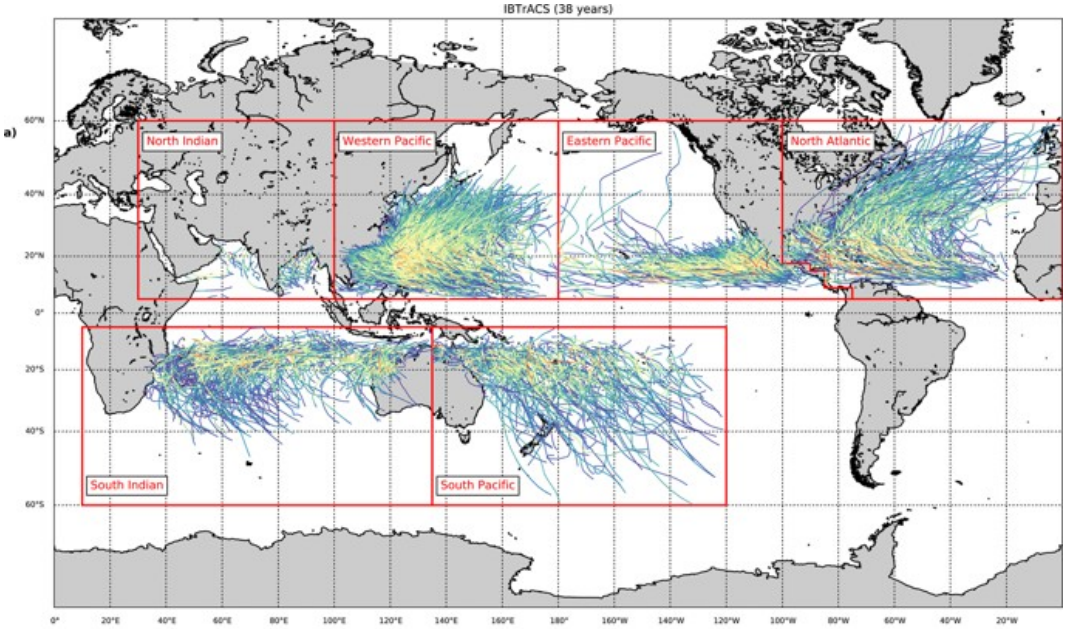


Goal:

developing and validating a computationally efficient, scalable, framework for large-scale flood risk assessment that incorporates synthetic TCs and couples global sea level prediction models to hydrodynamic inundation modelling.

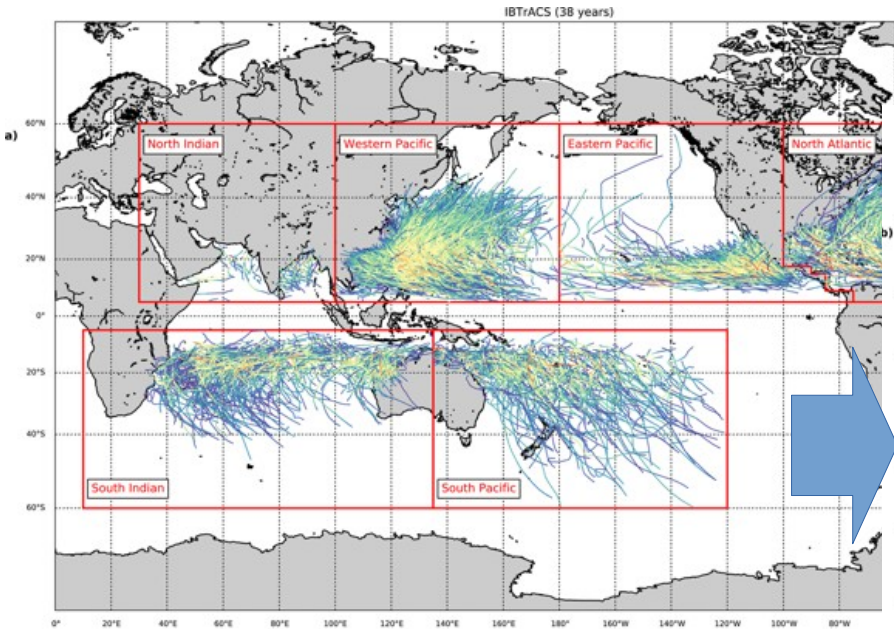
Generating a database of model cyclones

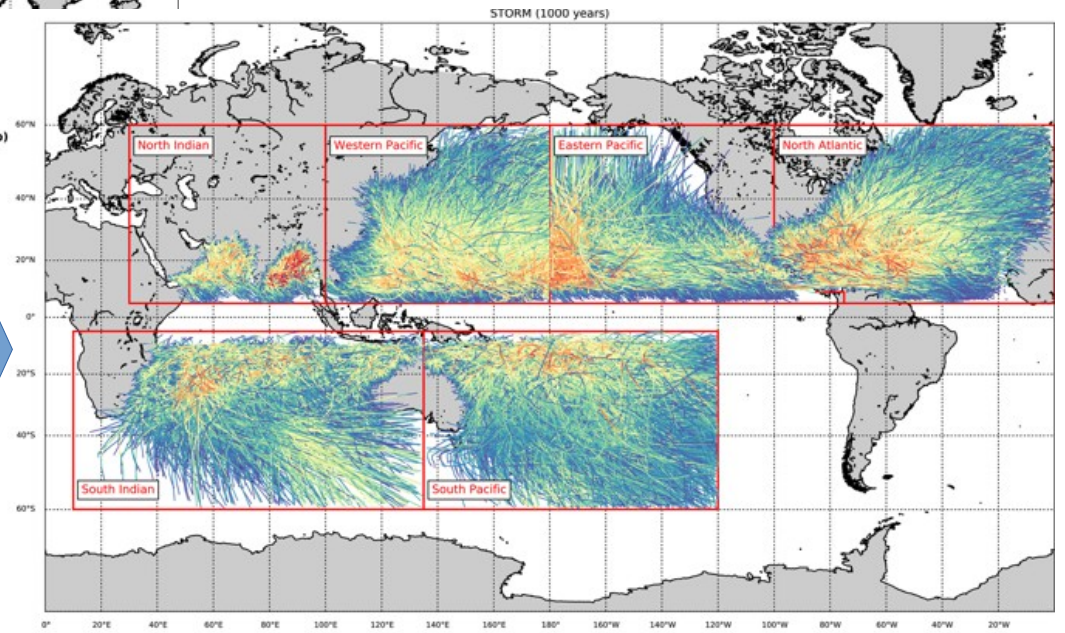
Historical events:
too few to reliably estimate extreme values



Generating a database of model cyclones

Historical events:
too few to reliably estimate extreme values





Synthetic database: statistical resampling of historical events extended to 10,000 years of events. (Bloemendaal et al. 2020)

Status/plan

- use the Global Tide and Surge Model (GTSM), based on the Dflow-FM engine of Delft3D
- OMUSE interface based on BMI interface of Dflow-FM
- use database of synthetic cyclones to base statistics on
- Optimize by combining multiple TC in one simulation (Dullaart, Chertova in prep.)
- model nesting: couple local inundation model with GTSM within OMUSE
- (semi) automatic grid extraction and refinement tools

netherlands eScience center

Conclusions

- OMUSE is a user friendly platform to conduct numerical experiments in Earth System modelling. OMUSE allows the user to express intricate couplings in readable & portable Python scripts

- OMUSE builds on the experience gained with AMUSE..
  ...it has been an interesting experience in cross-domain thinking!

- Current work is on extending the framework and make it more performant for massively parallel simulations

netherlands eScience center

Resources

OMUSE repository:

github.com/omuse-geoscience/omuse

OMUSE example scripts:

github.com/omuse-geoscience/omuse-examples

Project repositories:

github.com/CloudResolvingClimateModeling
github.com/nlesc-smcm
github.com/nlesc-mosaic

OMUSE Code paper:

Pelupessy et al. 2017, GMD 10, 3167