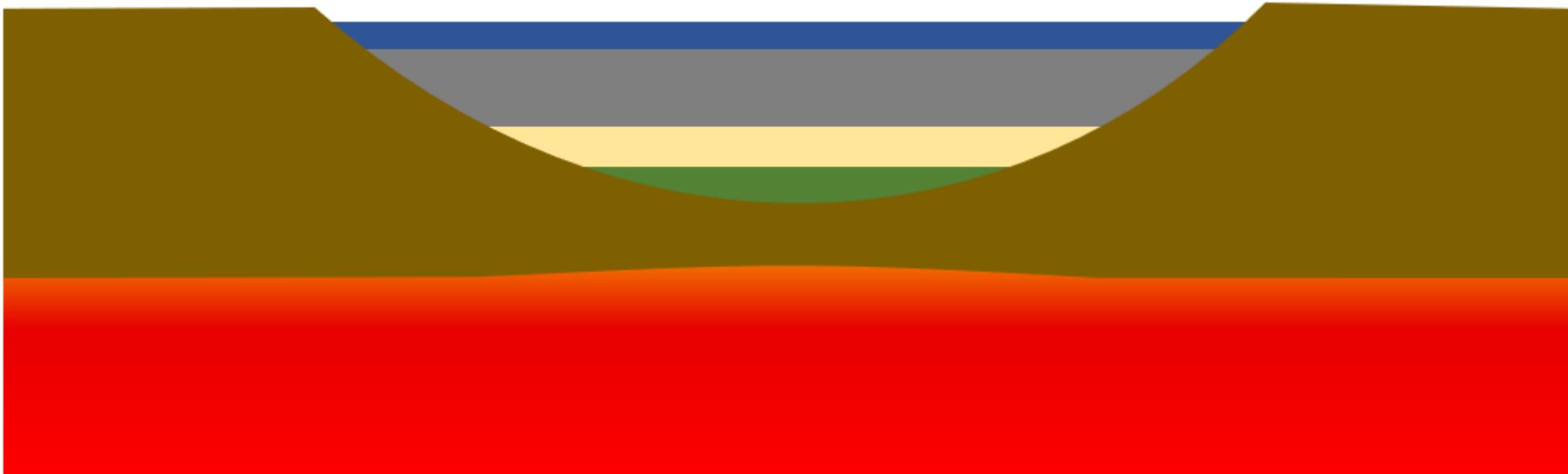


# warmth + ERT demo

Olve Maudal



a 25 minute live demo including Q&A  
FMU user forum, November 16, 2023

## ERT 8.0 documentation

Search

Welcome to the ERT Documentation!

### GETTING STARTED

Installation

**Configuration guide**

Restarting ES-MDA

Webviz-ert

### REFERENCE

Running ERT

Configuration

Workflows

Queue system

### THEORY

Ensemble based methods

### EXPERIMENTAL FEATURES

Plugin system

### DEVELOPER DOCUMENTATION

Roadmap

Development strategy

Concepts

The forward model

ERT Storage Server

Testing Qt components in pytest

# Configuration guide

This guide will introduce you to the basic concepts of ERT by demonstrating a project setup that uses a simple polynomial as the [forward model](#).

## Minimal Configuration

We first create a minimal configuration and run an experiment that doesn't execute any computations, but only generates the necessary folders and files.

### 1. Create a Folder:

Start by creating a folder named `poly_example`. This folder will contain all required files for our example.

## Create a Configuration File

Running ERT requires a dedicated configuration file, typically with the extension `.ert`.

### 1. Create the File:

Within the `poly_example` folder, create a file named `poly.ert` with the following content:

```
NUM_REALIZATIONS 5
```

`NUM_REALIZATIONS` specifies how many simulations you want to run.

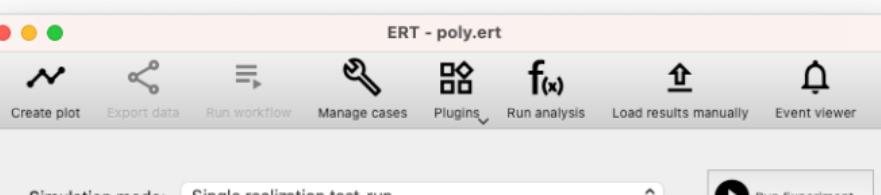
## Launch the User Interface

### 1. Run ERT:

Launch the GUI with the command:

```
ert gui poly.ert
```

The main ERT user interface window will pop up, as shown below:



### ON THIS PAGE

#### Minimal Configuration

[Create a Configuration File](#)

[Launch the User Interface](#)

[The Main Elements Relevant to This Guide Are:](#)

[Run an Empty Experiment](#)

[Inspect the results](#)

[Adding a Forward Model](#)

[Changing the Default Runpath](#)

[Creating a Script to Define the Forward Model](#)

[Marking the Script as Executable](#)

[Adding a Job to the Forward Model](#)

[Running with the New Job](#)

[Creating Parameters](#)

[Adding Prior Distributions](#)

[Configuring the parameter set](#)

[Reading Parameters in the Simulation Script](#)

[Reading Simulation Results a.k.a Responses](#)

[Increasing the number of realizations](#)

[Running with Sampled Parameters](#)

[Inspecting the Parameters and Results](#)

[Adding observations](#)

[Observations File](#)

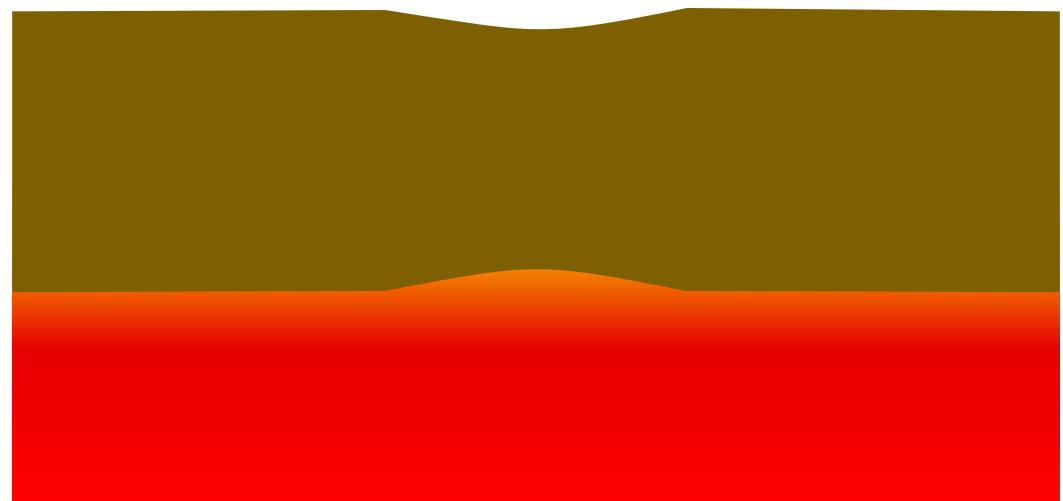
[Defining the Observation Configurations](#)

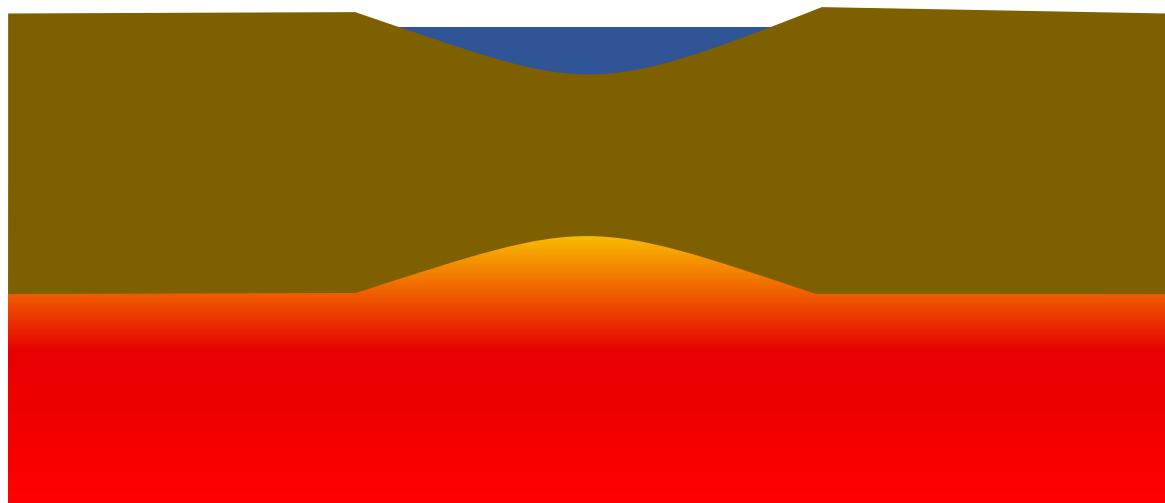
[Simulation and Analysis](#)

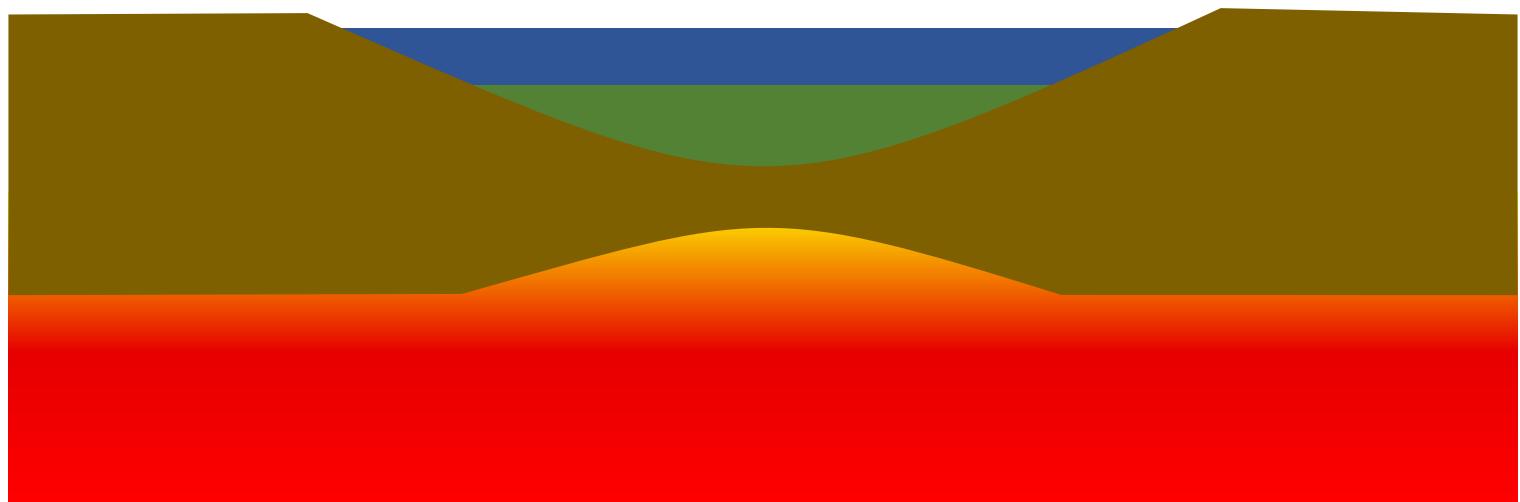
[Evaluating the Updated Parameters](#)

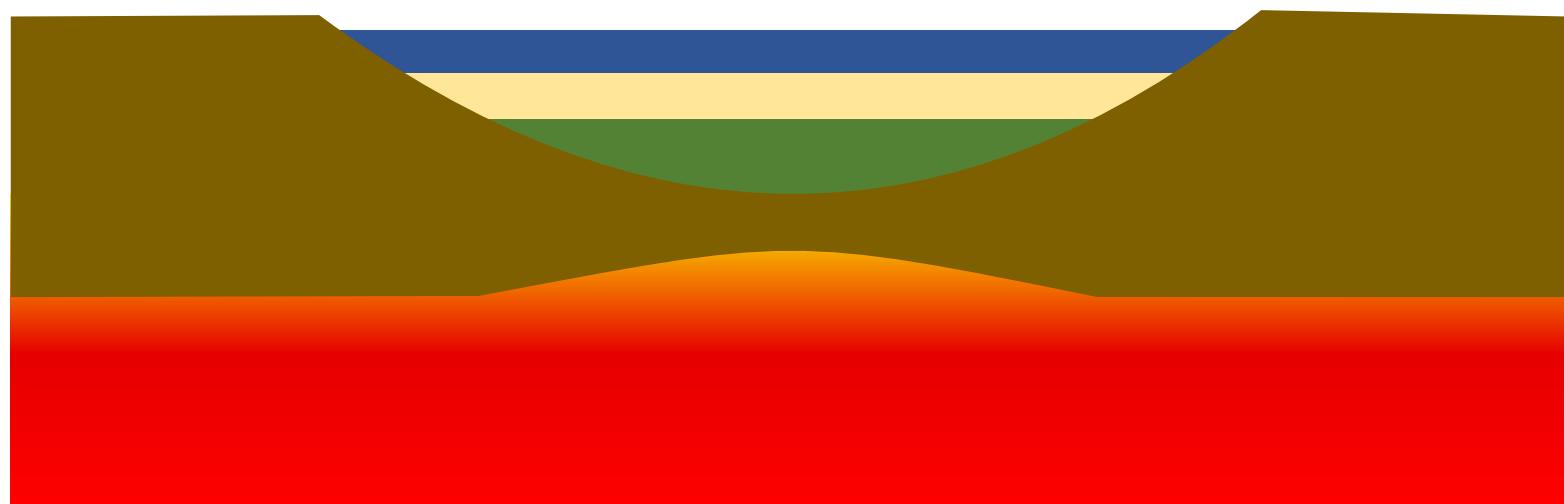
[Conclusion](#)

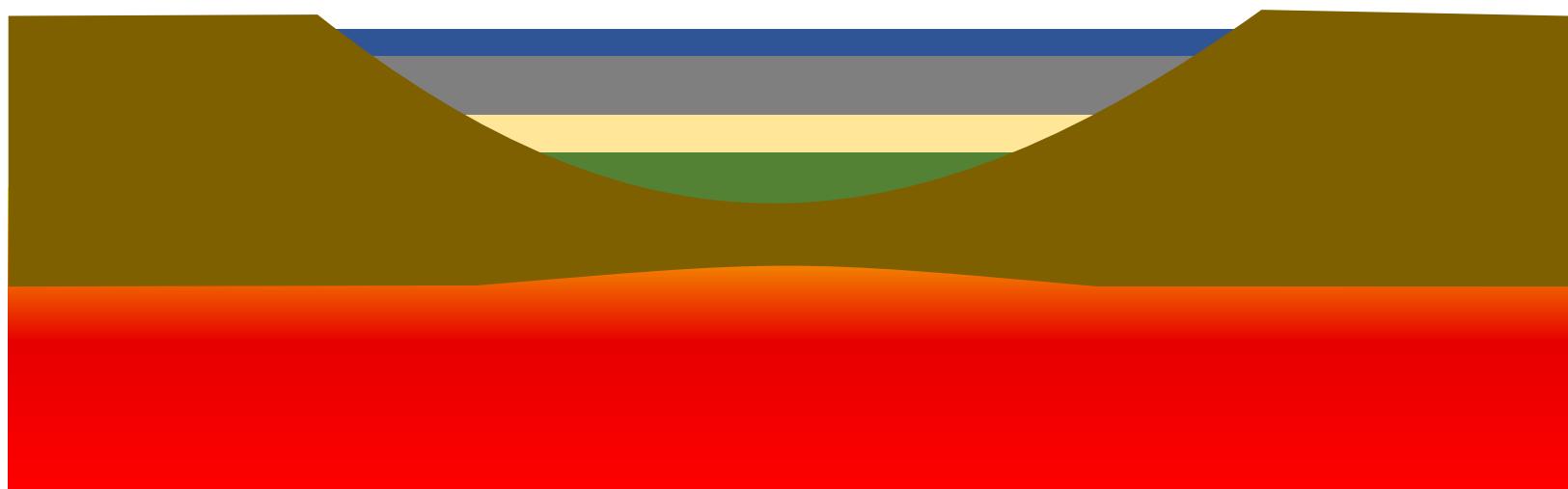


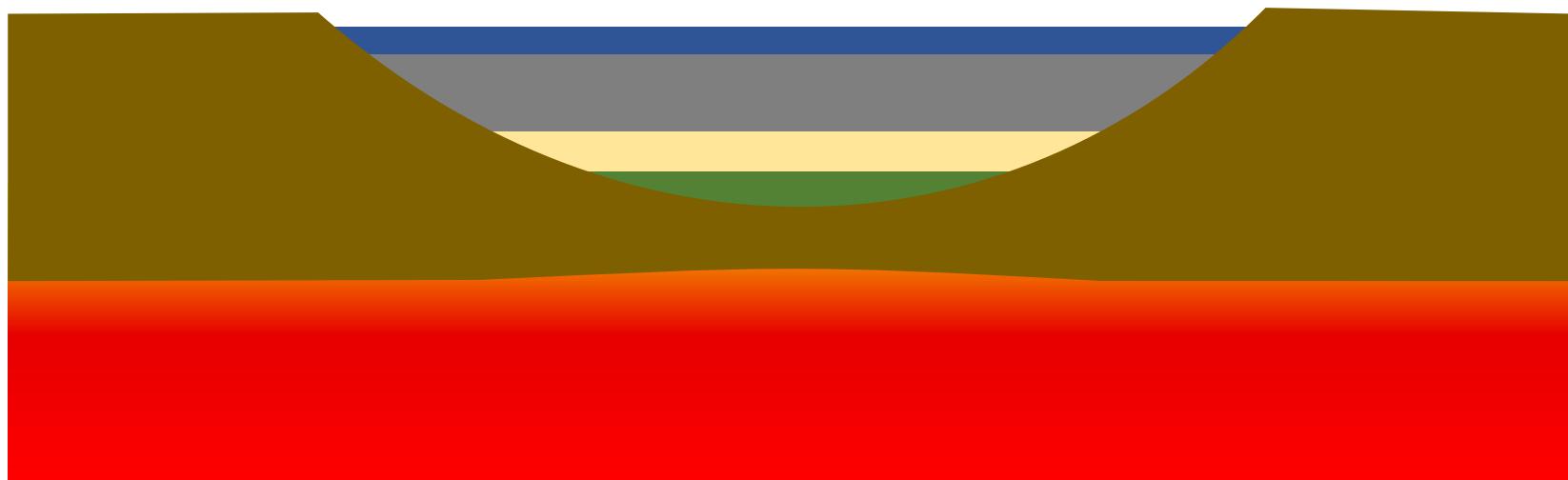


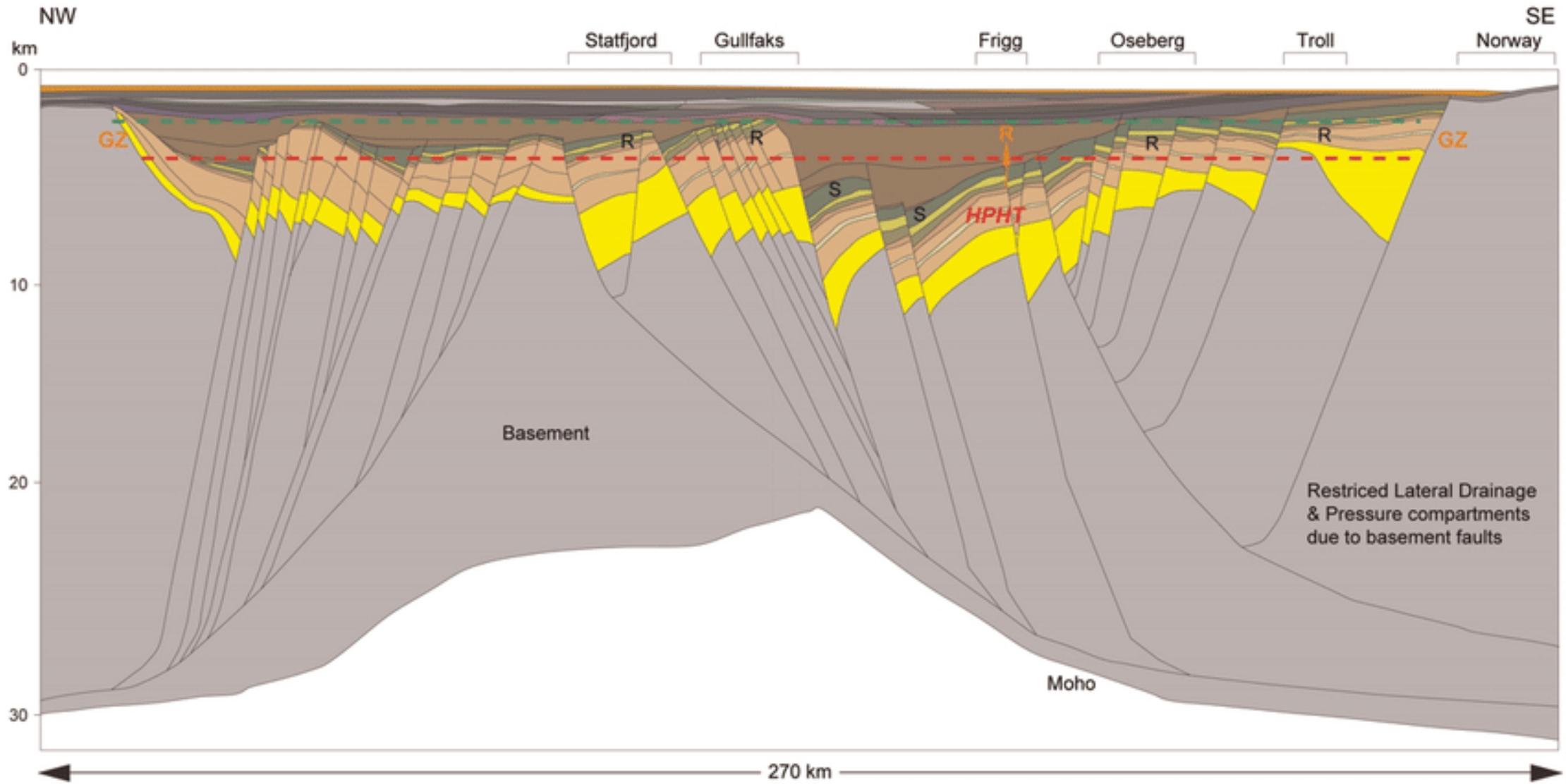












Modified after Fjeldskaar et al., 2004, Fig. 12

60°C isotherm  
120°C isotherm

R = Tertiary Reservoirs  
R = Jurassic Reservoirs

S = Jurassic Source Rocks

☰ README.md

# Warmth

## Forward modeling of thermal evolution through geological time

 Tests failing  docs passing

[Documentation](#)

warmth is a python package used for modeling thermal evolution based on McKenzie's type basin extension. It can be use for:

- Finding beta factor
- Calculating subsidence and thermal history
- Basement heat flow through time

## Features

- Multi-1D simulation
- Full 3D simulation with dolfinx
- Build model from either:
  - Python objects
  - [XTGeo](#) supported surface formats
- Multi-rift phase support
- Ensemble models with ERT <https://github.com/equinor/ert>



Adam Cheng

Prin Geol Geology - SUB CC SCM



warmth\_ertdemo

Public

Watch 0

Fork 1

Star 0

forked from [equinor/warmth](#)

main ▾ 1 branch 0 tags

Go to file

Add file ▾

Code ▾

## About



A demonstration of ERT by using Warmth

This branch is 9 commits ahead of equinor:main.

Contribute ▾ Sync fork ▾

olvemaudal	Add README file with demo script	148059b 1 hour ago	57 commits
.devcontainer	TST: pipeline to test 3D code	last week	
.github/workflows	TST: 3d ci mount	3 days ago	
demo	Add README file with demo script	1 hour ago	
docs	Merge remote-tracking branch 'origin/main' into pipelines	last week	
step0	Inline model in step 0	3 hours ago	
step1	Capture generated data in step1	3 hours ago	
step2	Add step2	2 hours ago	
step3	Add step3	2 hours ago	
subsheat3D	Fix crust properties per-node	last week	
tests	Merge remote-tracking branch 'origin/main' into pipelines	last week	
warmth	Compile for RGS, komodo and python 3.8	4 hours ago	

[Readme](#)[LGPL-3.0 license](#)[Security policy](#)[Activity](#)[0 stars](#)[0 watching](#)[1 fork](#)[Report repository](#)

## Releases

No releases published

[Create a new release](#)

## Packages

No packages published

[Publish your first package](#)

# Demo time

```
#  
# prepare  
  
git clone git@github.com:equinor/warmth_ertdemo.git  
cd warmth_ertdemo  
  
source /prog/res/komodo/stable/enable  
komodoenv _env  
source _env/enable  
pip install .
```

File Edit View Search Terminal Help

```
olvm@be-linrgsn110 ~]$ bash
olvm@be-linrgsn110 ~]$ git clone git@github.com:equinor/warmth_ertdemo.git
Cloning into 'warmth_ertdemo'...
remote: Enumerating objects: 443, done.
remote: Counting objects: 100% (96/96), done.
remote: Compressing objects: 100% (55/55), done.
remote: Total 443 (delta 51), reused 62 (delta 38), pack-reused 347
Receiving objects: 100% (443/443), 1.92 MiB | 601.00 KiB/s, done.
Resolving deltas: 100% (201/201), done.
'.png is not a valid attribute name: .gitattributes:1
olvm@be-linrgsn110 ~]$ cd warmth_ertdemo/
olvm@be-linrgsn110 warmth_ertdemo]$ source /prog/res/komodo/stable/enable
[2023.10.03-py38] [olvm@be-linrgsn110 warmth_ertdemo]$ komodoenv _env
Looking for /prog/res/komodo/2023.10.03-py38
Komodoenv is still in beta. Be aware that issues might occur and recreating environments once in a while is necessary.
```

For progress on stabilising komodoenv, see: <https://github.com/equinor/komodoenv/milestone/1>

Tracking a stable release of komodo.

```
venv      using /prog/res/komodo/2023.10.03-py38-rhel7/root/bin/python
create    komodoenv.conf
create    /private/olvm/warmth_ertdemo/_env/root/lib/python3.8/site-packages/_komodo.pth
create    root/bin/komodoenv-update
run       root/bin/komodoenv-update
install   setuptools
install   wheel
install   pip
remove    root/shims/komodoenv
```

Komodoenv has successfully been generated. You can now pip-install software.

```
$ source /private/olvm/warmth_ertdemo/_env/enable

[2023.10.03-py38] [olvm@be-linrgsn110 warmth_ertdemo]$ source _env/enable
_env + 2023.10.03-py38-rhel7) [olvm@be-linrgsn110 warmth_ertdemo]$ pip install .
Processing /private/olvm/warmth_ertdemo
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: pandas<2,>=1 in /prog/res/komodo/2023.10.03-py38-rhel7/root/lib64/python3.8/site-packages (from warmth==0.0.1) (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in /prog/res/komodo/2023.10.03-py38-rhel7/root/lib/python3.8/site-packages (from pandas<2,>=1->warmth==0.0.1) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /prog/res/komodo/2023.10.03-py38-rhel7/root/lib/python3.8/site-packages (from pandas<2,>=1->warmth==0.0.1) (2023.3)
Requirement already satisfied: numpy>=1.20.3 in /prog/res/komodo/2023.10.03-py38-rhel7/root/lib64/python3.8/site-packages (from pandas<2,>=1->warmth==0.0.1) (1.24.4)
Requirement already satisfied: six>=1.5 in /prog/res/komodo/2023.10.03-py38-rhel7/root/lib/python3.8/site-packages (from python-dateutil>=2.8.1->pandas<2,>=1->warmth==0.0.1) (1.16.0)
Building wheels for collected packages: warmth
  Building wheel for warmth (pyproject.toml) ... done
  Created wheel for warmth: filename=warmth-0.0.1-py3-none-any.whl size=87536 sha256=43a1bdaf44a8b5415c649ae5c7fd90c441ceeb0cd8b3e49179d68f0f85a744a8
  Stored in directory: /tmp/pip-ephem-wheel-cache-2jx2qvuw/wheels/3c/e2/ee/bf40fb2472e3c6bb8fd5b9a3125304eda7fb35c387abd5325c
Successfully built warmth
Installing collected packages: warmth
Successfully installed warmth-0.0.1
_env + 2023.10.03-py38-rhel7) [olvm@be-linrgsn110 warmth_ertdemo]$ S
```

```
#  
# step0 - demonstrate a warmth simulation  
#  
  
cd step0  
ls -al  
view mywell.py  
python mywell.py  
cd ..
```

File Edit View Search Terminal Help

```

import numpy as np
import pandas as pd
import warmth
import io
import random

#
# Load model
#

model_str = """
top          topage   k_cond    rhp        phi      decay    solidus    liquidus
360.000000    0.0  2.564000  1.498000e-06  0.584000  0.622000  2708.000000  2708.000000
523.190000    5.0  2.277900  1.585208e-06  0.606586  0.662183  2707.024001  2707.024001
845.880000   23.0  1.856198  1.608295e-06  0.637742  0.700175  2707.734916  2707.734916
1113.070000   34.0  0.769366  9.523252e-07  0.328388  0.389374  1266.639495  1266.639495
1364.940000   56.0  2.685191  1.428223e-06  0.568786  0.594719  2709.049274  2709.049274
1739.870000   66.0  1.959962  1.634753e-06  0.617657  0.637082  2709.410640  2709.410640
1991.920000  145.0  2.389656  1.524629e-06  0.592339  0.623368  2708.640057  2708.640057
2142.170000  164.0  2.139735  1.593953e-06  0.604479  0.662501  2638.327714  2638.327714
2572.900000  174.0  3.129789  1.008131e-06  0.493581  0.449826  2716.135856  2716.135856
3094.860000  201.0  3.481859  8.653857e-07  0.454647  0.385675  2708.997194  2708.997194
5416.485239  250.0  2.564000  1.498000e-06  0.584000  0.622000  2708.000000  2708.000000
"""

model_df = pd.read_fwf(io.StringIO(model_str.strip()))

#
# Set parameters
# hc           initial crust thickness (eg, 30000 meters)
# qbase        heat flow at base of the crust/moho, (eg, 0.030 W/m^2)
# time_start   start simulation (eg, 0 250 Ma)
# time_end     end of simulation (eg, 0 Ma)
# rift_start   time of rift starting (eg, 250 Ma)
# rift_end     time of rift ending (eg, 240 Ma)
# 

params = {
    'hc':             random.uniform(25000, 45000),
    'qbase':          random.uniform(0.020, 0.060),
    'time_start':     250,
    'time_end':       0,
    'rift_start':    250,
    'rift_end':      240
}

print(f"{params['hc']=}")
print(f"{params['qbase']=}")

#

```

```

    }

    print(f"{params['hc']=}")
    print(f"{params['qbase']=}")

    #

    # Run simulation
    #

    node = warmth.single_node()
    node.sediments_inputs = model_df
    model = warmth.Model()
    model.parameters.time_end = params['time_end']
    model.parameters.time_start = params['time_start']
    node.hc = params['hc']
    node.qbase = params['qbase']
    node.rift = np.array([[params['rift_start'], params['rift_end']]])
    model.builder.nodes = [[node]]
    model.builder.set_eustatic_sea_level(warmth.data.haq87)
    model.simulator.run(parallel=False)

    #

    # Capture results
    #

    node = model.builder.nodes[0][0]
    depths = node.result.temperature(0)[['depth']][1:]
    temps = node.result.temperature(0)[['values']][1:]
    depths_temps = [(d, t) for d, t in zip(depths, temps)][:-50]

    #

    # Output results
    #

    print("Depth Temp")
    for d, t in depths_temps[:5]:
        print(f"{int(d):>5} {int(t):>4}")
    print("...")
    for d, t in depths_temps[-5:]:
        print(f"{int(d):>5} {int(t):>4}")

```

```

import numpy as np
import pandas as pd
import warmth
import io
import random

#
# Load model
#

model_str = """
top      topage   k_cond    rhp      phi      decay
360.000000     0.0  2.564000  1.498000e-06  0.584000  0.622000
523.190000     5.0  2.277900  1.585208e-06  0.606586  0.662183
845.880000    23.0  1.856198  1.608295e-06  0.637742  0.700175
1113.070000   34.0  0.769366  9.523252e-07  0.328388  0.389374
1364.940000   56.0  2.685191  1.428223e-06  0.568786  0.594719
1739.870000   66.0  1.959962  1.634753e-06  0.617657  0.637082
1991.920000  145.0  2.389656  1.524629e-06  0.592339  0.623368
2142.170000  164.0  2.139735  1.593953e-06  0.604479  0.662501
2572.900000  174.0  3.129789  1.008131e-06  0.493581  0.449826
3094.860000  201.0  3.481859  8.653857e-07  0.454647  0.385675
5416.485239  250.0  2.564000  1.498000e-06  0.584000  0.622000
"""
model_df = pd.read_fwf(io.StringIO(model_str.strip()))

#
# Set parameters
# hc           initial crust thickness (eg, 30000 meters)
# qbase        heat flow at base of the crust/moho, (eg, 0.030
# time_start   start simulation (eg, 0 250 Ma)
# time_end     end of simulation (eg, 0 Ma)
# rift_start   time of rift starting (eg, 250 Ma)
# rift_end     time of rift ending (eg, 240 Ma)
#
params = {
    'hc':         random.uniform(25000, 45000),
    'qbase':      random.uniform(0.020, 0.060),
    'time_start': 250,
}

```

-:--- mywell.py Top (1,0) Git-main (Python)  
Loading vc-git...done

solidus liquidus

olvm@be-linrgsn110:~/sb/warmth\_ertdemo/step0

Depth	Temp	solidus	liquidus
428	5		
462	7		
527	10		
591	14		
672	19		
...			
4324	165		
4420	167		
4517	169		
4614	172		
4711	174		
...			
406	5		
440	7		
505	12		
569	17		
650	24		
...			
4302	223		
4398	226		
4495	230		
4592	233		
4689	237		

(\_env + 2023.10.03-py38-rhel7) [olvm@be-linrgsn110 step0]\$ python mywell.py  
params['hc']=38536.01002920719  
params['qbase']=0.03710564379802116  
Depth Temp

(\_env + 2023.10.03-py38-rhel7) [olvm@be-linrgsn110 step0]\$ python mywell.py  
params['hc']=36502.71377690384  
params['qbase']=0.058204522641749926  
Depth Temp

(\_env + 2023.10.03-py38-rhel7) [olvm@be-linrgsn110 step0]\$

```
#  
# Step 1 - using ERT just to run 5 realizations  
  
#  
  
cd step1  
ls -al  
# show mywell_eval.py // focus on output result  
chmod +x mywell_eval.py  
./mywell_eval.py  
ls -al  
cat mywell_temp.out  
# show mywell.ert  
# show MYWELL_WVAL  
ert gui mywell.ert  
# run ensemble experiment, approx 20 seconds  
tree mywell_out  
ls -al mywell_out/re*/it*/mywell_temp.out  
head -5 mywell_out/re*/it*/mywell_temp.out  
rm -rf storage mywell_out logs .ert_runpath_list  
cd ..
```

File Edit View Search Terminal Help

NUM\_REALIZATIONS 5

RUNPATH mywell\_out/realization-<IENS>/iter-<ITER>

GEN\_DATA MYWELL\_TEMP RESULT\_FILE:mywell\_temp.out

INSTALL\_JOB mywell\_eval MYWELL\_EVAL

FORWARD\_MODEL mywell\_eval

-UU-----F1 mywell.ert All (1,0) Git-main (Fundamental) -----  
EXECUTABLE mywell\_eval.py

```

        }

        #
        # Run simulation
        #

        node = warmth.single_node()
        node.sediments_inputs = model_df
        model = warmth.Model()
        model.parameters.time_end = params['time_end']
        model.parameters.time_start = params['time_start']
        node.hc = params['hc']
        node.qbase = params['qbase']
        node.rift = np.array([[params['rift_start'], params['rift_end']]])
        model.builder.nodes = [[node]]
        model.builder.set_eustatic_sea_level(warmth.data.haq87)
        model.simulator.run(parallel=False)

        #
        # Capture results
        #

        node = model.builder.nodes[0][0]
        depths = node.result.temperature(0)[['depth']][1:]
        temps = node.result.temperature(0)[['values']][1:]
        depths_temps = [(d, t) for d, t in zip(depths, temps)][:50]

        #
        # Output results
        #

        with open("mywell_temp.out", "w", encoding="utf-8") as f:
            for d, t in depths_temps:
                print(t, file=f)

```

-UUU-----F1 MYWELL\_EVAL All (1,0) Git-main (Fundamental) -----

-UU-----F1 mywell\_eval.py Bot (80,0) Git-main (Python) -----

View Help

Simulation mode: Ensemble experiment

Experiment

Current case:

Runpath: /private/olvms/sb/warmth\_ertdemo/step1/mywell\_out/realization-&lt;1EN

Number of realizations: 5

Active realizations 0-4

Iteration 0

Configuration summary

Experiment - mywell.

Total progress 100% — Simulations completed.

Progress for iteration 0

Finished (5/5)

Failed (0/5)

Running (0/5)

Pending (0/5)

Waiting (0/5)

Unknown (0/5)

Running time: 19 seconds

Data types

Search

MYWELL\_TEMP@0

Observations available

Plot case

Add case to plot

default

Ensemble

Statistics

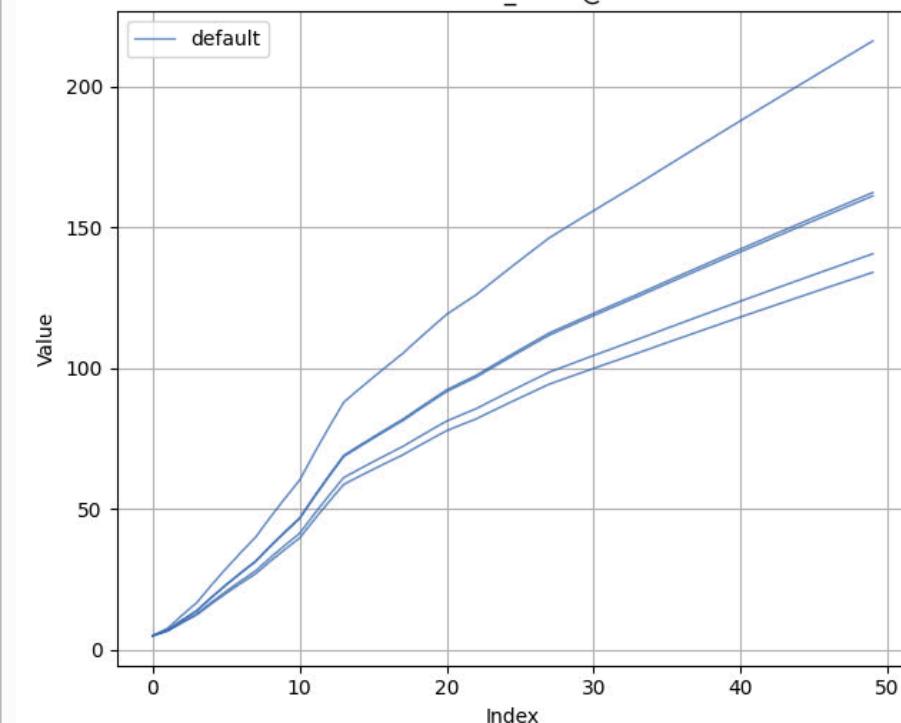
Histogram

Gaussian KDE

Distribution

Cross case statistics

MYWELL\_TEMP@0



olvm@be-linrgsn110:~/sb/warmth\_ertdemo/step1

File Edit View Search Terminal Help

```
(wd now: ~/sb/warmth_ertdemo/step1)
(_env + 2023.10.03-py38-rhel7) [olvm@be-linrgsn110 step1]$
(_env + 2023.10.03-py38-rhel7) [olvm@be-linrgsn110 step1]$ ert gui mywell.ert
2023-11-16 13:21:39,159 [WARNING] py.warnings: /prog/res/komodo/2023.10.03-py38-rhel7/root/lib/python3.8/site-packages/xtgeoapp_grd3dmmaps/hook_implementations/jobs.py:3: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
from pkg_resources import resource_filename
```

```
#  
# Step 2 - let ERT generate parameters  
  
#  
  
cd step2  
ls -al  
ert gui mywell.ert &  
# run Ensemble experiment // approx 40 seconds  
# show mywell.ert  
# show mywell_eval.py  
# show params tmpl  
# show params priors  
tree mywell_out  
head mywell_out/re*/it*/params.json  
ert gui mywell.ert  
rm -rf storage mywell_out logs .ert_runpath_list  
cd ..
```

File Edit View Search Terminal Help

```

NUM_REALIZATIONS 30

RUNPATH mywell_out/realization-<IENS>/iter-<ITER>

FORWARD_MODEL COPY_FILE(<FROM>=<CONFIG_PATH>/mymodel.txt, <TO>=<RUNPATH>/mymodel.txt)

GEN_KW PARAMS params tmpl params.json params_priors

GEN_DATA MYWELL_TEMP RESULT_FILE:mywell_temp.out

INSTALL_JOB mywell_eval MYWELL_EVAL
FORWARD_MODEL mywell_eval

# Load model
#
model_df = pd.read_fwf("mymodel.txt")

#
# Set parameters
# hc           initial crust thickness (eg, 30000 meters)
# qbase        heat flow at base of the crust/moho, (eg, 0.030 W/m^2)
# time_start   start simulation (eg, 0 250 Ma)
# time_end     end of simulation (eg, 0 Ma)
# rift_start   time of rift starting (eg, 250 Ma)
# rift_end     time of rift ending (eg, 240 Ma)
#
with open("params.json", encoding="utf-8") as f:
    params = json.load(f)

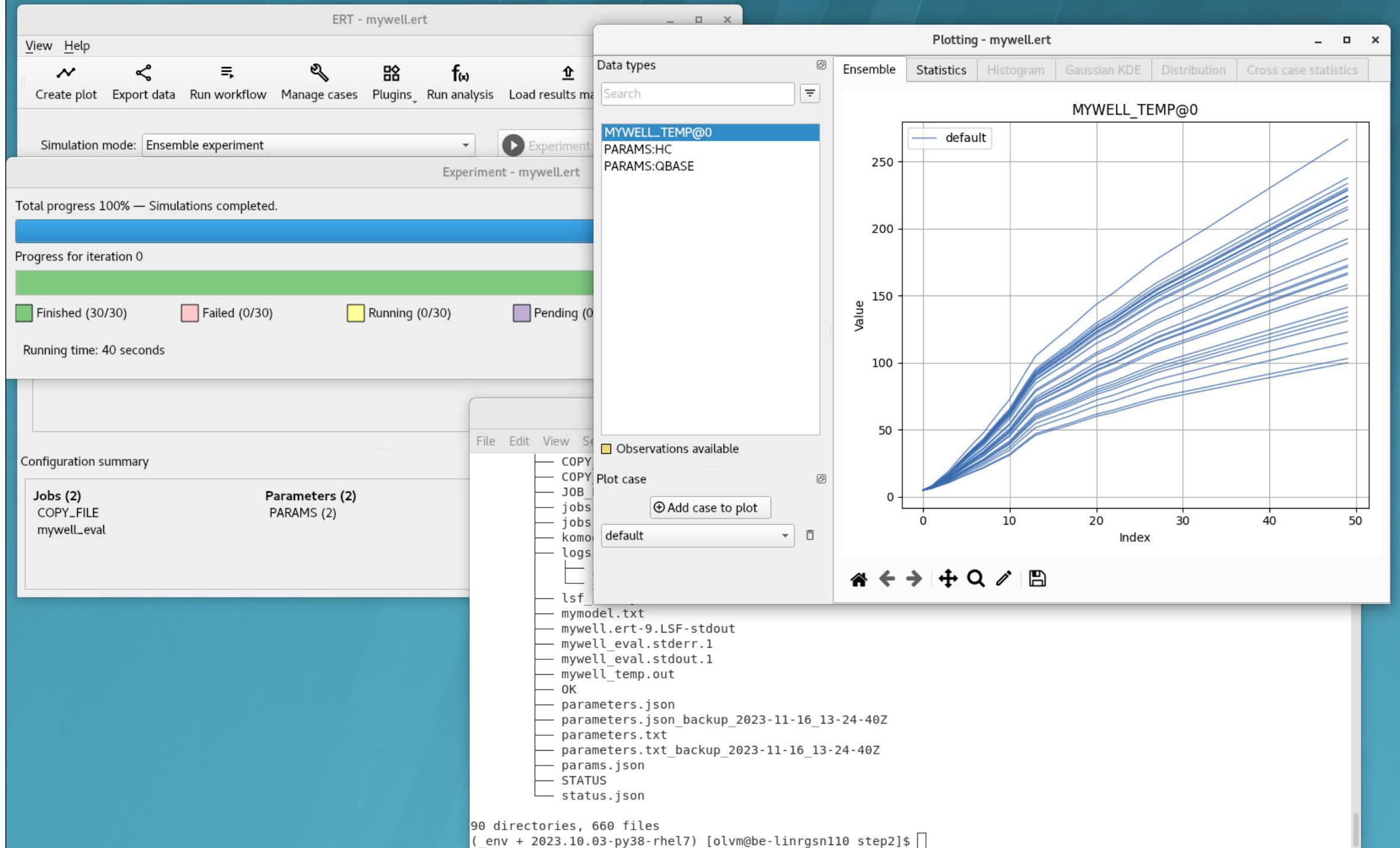
#
# Run simulation
#



-UU-----F1 mywell.ert All (1,0) Git-main (Fundamental) -----
{ "hc": <HC>, "qbase": <QBASE>, "time_start": 250, "time_end": 0, "rift_start": 250, "rift_end": 240 }
-UU-----F1 mywell_eval.py 7% (32,0) Git-main (Python) -----
top topage k_cond rhp phi decay solidus liquidus
360.000000 0.0 2.564000 1.498000e-06 0.584000 0.622000 2708.000000 2708.000000
523.190000 5.0 2.277900 1.585208e-06 0.606586 0.662183 2707.024001 2707.024001
845.880000 23.0 1.856198 1.608295e-06 0.637742 0.700175 2707.734916 2707.734916
1113.070000 34.0 0.769366 9.523252e-07 0.328388 0.389374 1266.639495 1266.639495
1364.940000 56.0 2.685191 1.428223e-06 0.568786 0.594719 2709.049274 2709.049274
1739.870000 66.0 1.959962 1.634753e-06 0.617657 0.637082 2709.410640 2709.410640
1991.920000 145.0 2.389656 1.524629e-06 0.592339 0.623368 2708.640057 2708.640057
2142.170000 164.0 2.139735 1.593953e-06 0.604479 0.662501 2638.327714 2638.327714
2572.900000 174.0 3.129789 1.008131e-06 0.493581 0.449826 2716.135856 2716.135856
3094.860000 201.0 3.481859 8.653857e-07 0.454647 0.385675 2708.997194 2708.997194
5416.485239 250.0 2.564000 1.498000e-06 0.584000 0.622000 2708.000000 2708.000000

-UU-----F1 params tmpl All (1,0) Git-main (Fundamental) -----
HC UNIFORM 25000 45000
QBASE UNIFORM 0.020 0.060

-UU-----F1 params_priors All (3,0) Git-main (Fundamental) -----
-UU-----F1 mymodel.txt All (1,0) Git-main (Text) -----
```



```
#  
# Step 3 - add observations and do history matching with ES-MDA  
  
#  
  
cd step3  
ert es_mda mywell.ert  
# show mywell_temp_data.txt  
# show observations  
# show mywell.ert  
ert gui mywell.ert  
rm -rf storage mywell_out logs .ert_runpath_list  
cd ..
```

View Help

Create plot Export data Run workflow Manage cases Plugins Run analysis Load results manually Event viewer

Simulation mode: Multiple Data Assimilation (ES MDA) - Recommended

Run Experiment

Runpath: /private/olvm/sb/warmth\_ertdemo/step3/mywell\_out/realization-&lt;IENS&gt;/iter-&lt;ITER&gt;

Number of realizations: 30

Target c

olvm@be-linrgsn110:~/sb/warmth\_ertdemo/step3

Relative File Edit View Search Terminal Help

Normali NUM\_REALIZATIONS 30

Analysis QUEUE\_SYSTEM LSF

Active re RUNPATH mywell\_out/realization-&lt;IENS&gt;/iter-&lt;ITER&gt;

Restart FORWARD\_MODEL COPY\_FILE(&lt;FROM&gt;=&lt;CONFIG\_PATH&gt;/mymodel.txt, &lt;TO&gt;=&lt;RUNPATH&gt;/mymodel.txt)

Restart FORWARD\_MODEL COPY\_FILE(&lt;FROM&gt;=&lt;CONFIG\_PATH&gt;/mymodel.txt, &lt;TO&gt;=&lt;RUNPATH&gt;/mymodel.txt)

Restart GEN\_KW PARAMS params tmpl params json params priors

Configuration GEN\_DATA MYWELL\_TEMP RESULT\_FILE:mywell\_temp.out

**Jobs (2)**

OBS\_CONFIG observations

COPY\_FILE INSTALL\_JOB mywell\_eval MYWELL\_EVAL

mywell\_eval FORWARD\_MODEL mywell\_eval

```
-UU-----F1 mywell.ert All (1,0) Git-main (Fundamental) -----
GENERAL_OBSERVATION MYWELL_TEMP {
  DATA      = MYWELL_TEMP;
  INDEX_LIST = 20,25,36;
  OBS_FILE   = mywell_temp_data.txt;
};
```

```
-UU-----F1 observations All (6,0) Git-main (Fundamental) -----
82.0 10.0
92.0 10.0
118.0 10.0
```

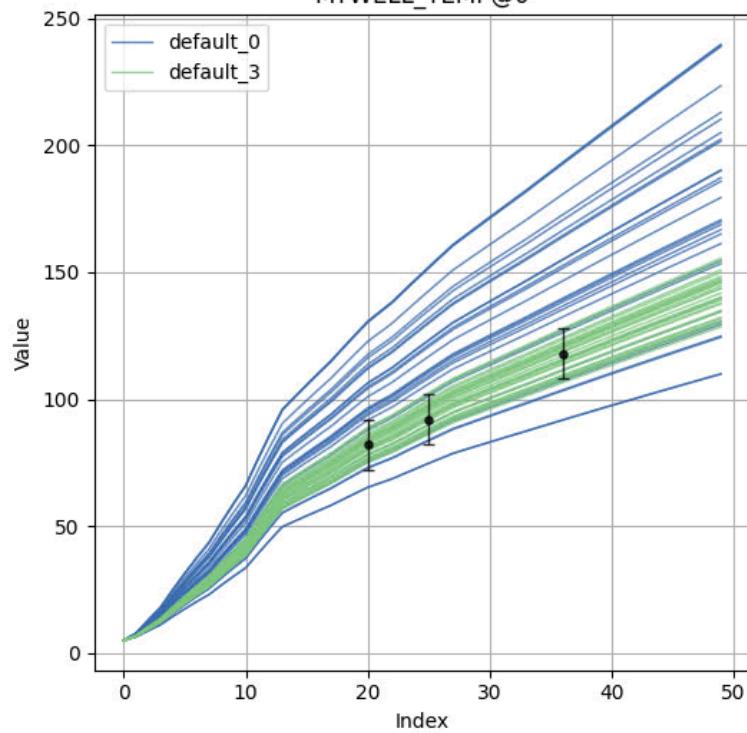
```
-UU-----F1 mywell_temp_data.txt All (1,0) Git-main (Text) -----
```



Plotting - mywell.ert

Ensemble Statistics Histogram Gaussian KDE Distribution Cross

## MYWELL\_TEMP@0



Home Back Forward Magnifying glass Pen Save

Q&A

Thanks to Adam Cheng, Heike Gröger, Tryge Kløv, Espen Hauge, Kristine Årland for inviting me into doing stuff like this at the TDI Subsurface Uncertainty Hackathon in Trondheim, October 2023 24-26

A big thank to Adam Cheng for creating the Warmth tool

A big thanks to SCOUT and H2 for creating and maintaining this fantastic ERT tool

A big thank to the HPC community in Equinor

A big thank to the whole FMU community in Equinor

