3_orchestrating_4C_simulations_with_QUEENS

September 29, 2025

1 Simulation analytics - Orchestrating 4C simulations with QUEENS

Until now, all the simulation models we used in QUEENS were Python based. Now for a lot of us, this is not the case. From fenics to abaqus, we all have our own preferred solvers. One main advantage of QUEENS is its independence of solver type. This means, to use your QUEENS model with your favorite solver, the only thing you have to do is write an interface.

Since QUEENS and the multiphysics solver 4C share some developers, the QUEENS community already provides a ready to go interface. So let's start there.

2 The 4C model

In this example, we'll have a look at a nonlinear solid mechanics problem. The momentum equation is given by

$$\rho\ddot{u} = \nabla \cdot \sigma + b \text{ in } \Omega$$

where u are the displacements, σ the Cauchy stress and b volumetric body forces. For the examples, $b = 0$, and we prescribe Dirichlet boundary conditions

$$u = 0 \text{ on } \Gamma_w = \{z = -2.5\}$$

$$u = [0, 0, 0.55] \text{ on } \Gamma_e = \{z = 2.5\}$$

and a Neumann condition:

$$\sigma \cdot n = 0 \text{ on } \Gamma \setminus \{\Gamma_w \cup \Gamma_e\}$$

We assume a static problem, so $\ddot{u} = 0$, and use a Saint Venant–Kirchhoff model material

$$S = \lambda \mathrm{tr}(E) I + 2\mu E$$

where S is the the second Piola-Kirchhoff stress tensor and E the Green-Lagrange strain tensor with the Lamé parameters

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$$

$$\mu = \frac{E}{2(1+\nu)}$$

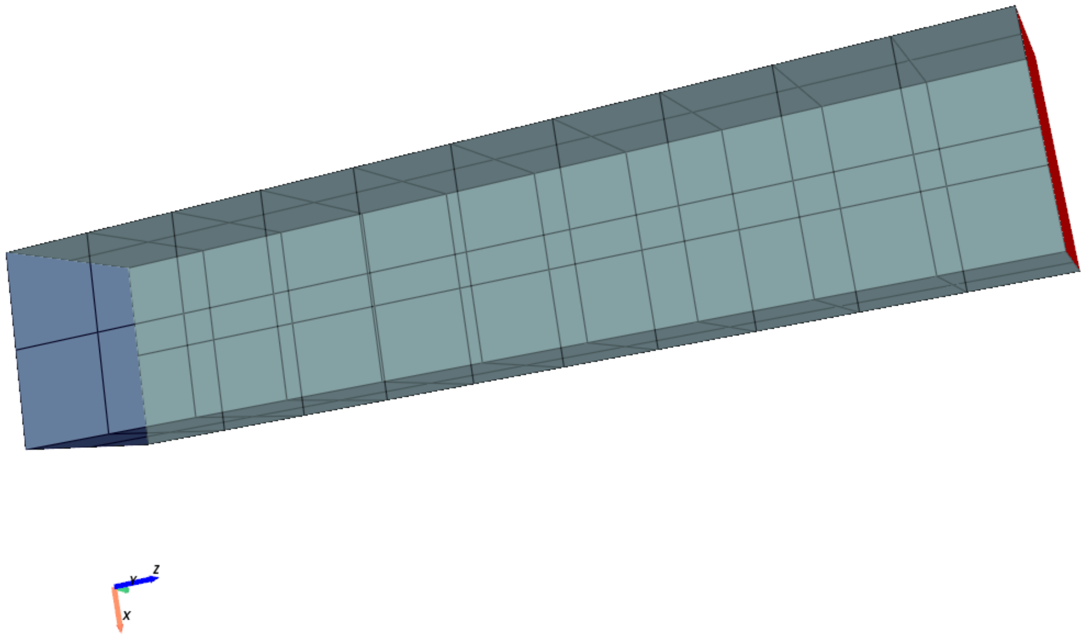
where E is the Young's moduli and ν the Poisson ratios.

The problem is discretized in space using 40 linear finite elements.

> Note: This tutorial is based on the 4C tutorial Preconditioners for Iterative Solvers

[45]:
```python
# Let's look at the mesh!

import pyvista as pv
pv.set_jupyter_backend("static")

fe_mesh = pv.read("beam_coarse.exo")

pl = pv.Plotter("Finite Element mesh")
pl.add_mesh(fe_mesh, show_edges=True, opacity=0.8)
pl.add_mesh(pv.Plane([0, 0, -2.50001], [0, 0, 1]), color="blue")  # Gamma_w
pl.add_mesh(pv.Plane([0, 0, 2.50001], [0, 0, 1]), color="red")  # Gamma_e
pl.add_axes(line_width=5)
pl.camera_position = [
    (1.1321899035097993, -6.851600196807601, 2.7649096132703574),
    (0.0, 0.0, 0.2749999999999999),
    (-0.97637930372977, -0.08995062285804697, 0.19644933366041056),
]
pl.show()
```

```
/data/a11brebu/miniforge3/envs/queens-ukacm-gacm/lib/python3.11/site-
packages/pyvista/core/utilities/misc.py:289: PyVistaDeprecationWarning:
../../../../../data/a11brebu/miniforge3/envs/queens-ukacm-
gacm/lib/python3.11/site-packages/pyvista/core/utilities/misc.py:289: Argument
'off_screen' must be passed as a keyword argument to function
'Plotter.__init__'.
From version 0.50, passing this as a positional argument will result in a
TypeError.
  obj = super().__call__(*args, **kwargs)  # type: ignore[misc]
/tmp/ipykernel_1340701/1148927816.py:10: PyVistaDeprecationWarning:
../../../../../tmp/ipykernel_1340701/1148927816.py:10: Arguments 'center',
'direction' must be passed as keyword arguments to function 'Plane'.
From version 0.50, passing these as positional arguments will result in a
TypeError.
  pl.add_mesh(pv.Plane([0, 0, -2.50001], [0, 0, 1]), color="blue")  # Gamma_w
/tmp/ipykernel_1340701/1148927816.py:11: PyVistaDeprecationWarning:
../../../../../tmp/ipykernel_1340701/1148927816.py:11: Arguments 'center',
'direction' must be passed as keyword arguments to function 'Plane'.
From version 0.50, passing these as positional arguments will result in a
TypeError.
  pl.add_mesh(pv.Plane([0, 0, 2.50001], [0, 0, 1]), color="red")  # Gamma_e
```



The red plane is Γ_e, the blue one is Γ_w.

3 Simulation Analytics

We now want to have a look at the Cauchy stresses σ for various Young's modulus E and Poisson ratio ν. For the purpose of demonstration, let's an assume a uniform distribution for each material parameters.

Of course we use QUEENS, so let's get started!

[46]:
```python
from queens.distributions import Uniform
from queens.parameters import Parameters

# Let's define the distributions
young = Uniform(10, 10000)
poisson_ratio = Uniform(0, 0.5)

# Let's define the parameters
parameters = Parameters(young=young, poisson_ratio=poisson_ratio)
```

Now that we have the parameters ready, we need a `Driver`. In QUEENS terms, drivers call the underlying simulation models, provide the simulation input and handle logging and error management. They are the interfaces to arbitrary simulation software and essentially do everything that is needed to start a **single** simulation.

As mentioned before, for 4C these are already provided by the QUEENS community.

[47]:
```python
from queens_interfaces.fourc.driver import Fourc
import pathlib

home = pathlib.Path("/home/a11brebu")
fourc_executable = "/data/a11brebu/4C_build/4C"
input_template = "template_herakles.4C.yaml"

fourc_driver = Fourc(
    parameters=parameters,  # parameters to run
    input_templates=input_template,  # input template
    executable=fourc_executable,  # 4C executable
)
```

4C uses input yaml files which needs to be created for every simulation. For this we provide a jinja2 input **template** to the driver. Jinja2 is a fast, expressive, extensible templating engine, that from the template:

```yaml
MATERIALS:
  - MAT: 1
    MAT_Struct_StVenantKirchhoff:
      YOUNG: {{ young }}
      NUE: {{ poisson_ratio }}
      DENS: 1
```

inserts the desired values (example `young = 1000` and `poisson_ratio = 0.3`) into the template creating:

```yaml
MATERIALS:
  - MAT: 1
    MAT_Struct_StVenantKirchhoff:
      YOUNG: 1000
      NUE: 0.3
      DENS: 1
```

For 4C, this allows us to create inputs for simulation runs without the need for a 4C API! Let's run this example!

[48]:
```python
import pathlib
import numpy as np

driver_output = fourc_driver.run(
    sample=np.array([1000, 0.3]),
    job_id="fourc_young_1000_poisson_ratio_03",
    num_procs=1,
    experiment_dir=pathlib.Path("."),
    experiment_name="fourc_run",
)
```

Some information: - `sample`: is the inputs for which we want to evaluate the model. The order of the data is given by the order of which we define the parameters. - `job_id`: is the identifier for the 4C run and also creates a folder with the same name where the data is stored. - `num_procs`: 4C is called using MPI. This is the number of processes that 4C will be called with. - `experiment_dir`: is the directory where we find all necessary data for 4C (input template or any other required files). - `experiment_name`: is an identifier of a QUEENS run. This will come in handy when we evaluate 4C multiple times.

Let's look at the output folder structure of the QUEENS run:

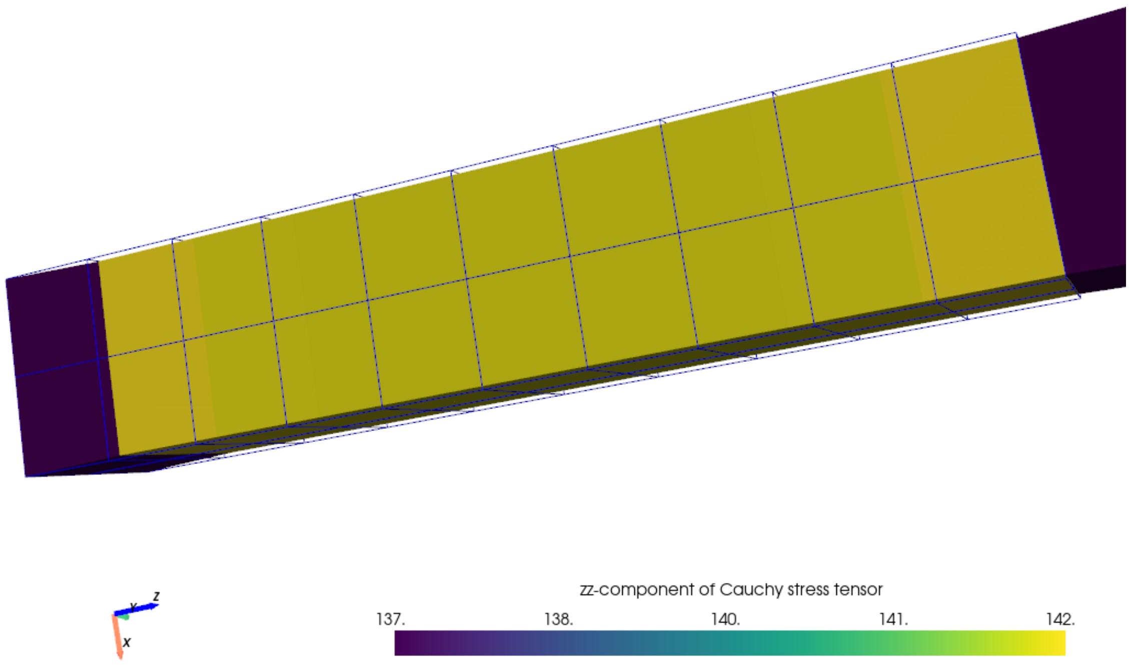
```
fourc_young_1000_poisson_ratio_03/
  input_file.yaml
  jobscript.sh
  metadata.yaml
  output
      output.control
      output.log
      output.mesh.structure.s0
      output-structure.pvd
      output-vtk-files
          structure-00000-0.vtu
          structure-00000.pvtu
          structure-00001-0.vtu
          structure-00001.pvtu
```

What are those files? - `input_file.yaml`: is the 4C input file with $E = 1000$ and $\nu = 0.3$ - `jobscript.sh`: contains the command with which 4C was called - `metadata.yaml`: contains metadata of the 4C (timings, inputs, outputs, …) - `output`-folder: all the files in here are output files of 4C! The sole exception is `output.log` where the logging data was redirected to this file by QUEENS

Go have a look at the `yaml`, `sh` and log files.

Now let us look at the Cauchy stresses σ

[49]:
```
# This is just needed for some nice plots

from utils import plot_results

plotter = pv.Plotter()
plot_results(
    "fourc_young_1000_poisson_ratio_03/output/output-vtk-files/structure-00001.
  ↪pvtu",
    plotter,
)
plotter.show()
```



We plotted the deformed mesh. The original mesh is displayed as the blue 'wireframe'. Let us look into the driver output in Python.

[50]:
```python
print("Driver output:", driver_output)
```

```
Driver output: (None, None)
```

As we can see it's a tuple of two `None`. The first one is supposed to be the output of the model, and the second one the model gradient.

> Note: The gradient value is only available for certain models, this particular 4C model can't do so.

But why is it `None`?

Well, we told the driver how to run a 4C simulation, but never told it which information we want to extract. For this we need a data processor, i.e., an object that is able to extract data from the simulation results.

[51]:
```python
from queens.data_processors import PvdFile


# Create custom data processor from the available PVD file reader
```

```python
class CustomDataProcessor(PvdFile):
    def __init__(
        self,
        field_name,
        failure_value,
        file_name_identifier=None,
        files_to_be_deleted_regex_lst=None,
    ):
        super().__init__(
            field_name,
            file_name_identifier,
            file_options_dict={},
            files_to_be_deleted_regex_lst=files_to_be_deleted_regex_lst,
            time_steps=[-1],
            block=0,
            point_data=False,  # We use cell data
        )
        self.failure_value = failure_value

    def get_data_from_file(self, base_dir_file):

        # Get the data
        data = super().get_data_from_file(base_dir_file)

        # Got some data
        if data is not None:
            data = data.flatten()
            # Simulation started but only the first step was exported
            if np.abs(data).sum() < 1e-8:
                return self.failure_value

            # Return result
            return data

        # Simulation could not even start
        else:
            return self.failure_value


custom_data_processor = CustomDataProcessor(
    field_name="element_cauchy_stresses_xyz",  # Name of the field in the pvtu␣
 ↪file
    file_name_identifier="output-structure.pvd",  # Name of the file in the␣
 ↪output folder
    failure_value=np.nan * np.ones(40 * 6),  # NaN for all elements
)
```

We defined our data processor, so let's continue.

[52]:
```python
# Create a new driver
fourc_driver = Fourc(
    parameters=parameters,
    input_templates=input_template,
    executable=fourc_executable,
    data_processor=custom_data_processor,  # tell the driver to extract the data
)

# Run the driver again
driver_output, _ = fourc_driver.run(
    sample=np.array([1000, 0.3]),
    job_id="fourc_young_1000_poisson_ratio_03",
    num_procs=1,
    experiment_dir=pathlib.Path("."),
    experiment_name="fourc_run",
)

print("Driver output", driver_output)
print(
    "Number of entries",
    len(driver_output),
    "i.e., 40 elements x 6 Cauchy stress values",
)
```

```
Driver output [ 2.33105756e+01  2.33105756e+01  1.36871583e+02  1.76081898e-01
 -8.12065251e+00 -8.12065251e+00 -8.70380431e-01 -8.70380431e-01
  1.42473762e+02  7.09472978e-02  1.60444657e+00  1.60444657e+00
 -3.19473608e-01 -3.19473608e-01  1.42331028e+02  1.17058224e-01
  2.25417129e-01  2.25417129e-01 -1.84238993e-02 -1.84238993e-02
  1.42260550e+02 -1.06082947e-02 -3.10846146e-03 -3.10846146e-03
  3.76574842e-03  3.76574842e-03  1.42255374e+02 -2.80804488e-04
 -2.89206800e-03 -2.89206800e-03  3.76574842e-03  3.76574842e-03
  1.42255374e+02 -2.80804488e-04  2.89206800e-03  2.89206800e-03
 -1.84238993e-02 -1.84238993e-02  1.42260550e+02 -1.06082947e-02
  3.10846146e-03  3.10846146e-03 -3.19473608e-01 -3.19473608e-01
  1.42331028e+02  1.17058224e-01 -2.25417129e-01 -2.25417129e-01
 -8.70380431e-01 -8.70380431e-01  1.42473762e+02  7.09472978e-02
 -1.60444657e+00 -1.60444657e+00  2.33105756e+01  2.33105756e+01
  1.36871583e+02  1.76081898e-01  8.12065251e+00  8.12065251e+00
  2.33105756e+01  2.33105756e+01  1.36871583e+02 -1.76081898e-01
  8.12065251e+00 -8.12065251e+00 -8.70380431e-01 -8.70380431e-01
  1.42473762e+02 -7.09472978e-02 -1.60444657e+00  1.60444657e+00
 -3.19473608e-01 -3.19473608e-01  1.42331028e+02 -1.17058224e-01
 -2.25417129e-01  2.25417129e-01 -1.84238993e-02 -1.84238993e-02
  1.42260550e+02  1.06082947e-02  3.10846146e-03 -3.10846146e-03
  3.76574842e-03  3.76574842e-03  1.42255374e+02  2.80804488e-04
  2.89206800e-03 -2.89206800e-03  3.76574842e-03  3.76574842e-03
  1.42255374e+02  2.80804488e-04 -2.89206800e-03  2.89206800e-03
 -1.84238993e-02 -1.84238993e-02  1.42260550e+02  1.06082947e-02
 -3.10846146e-03  3.10846146e-03 -3.19473608e-01 -3.19473608e-01
  1.42331028e+02 -1.17058224e-01  2.25417129e-01 -2.25417129e-01
 -8.70380431e-01 -8.70380431e-01  1.42473762e+02 -7.09472978e-02
  1.60444657e+00 -1.60444657e+00  2.33105756e+01  2.33105756e+01
  1.36871583e+02 -1.76081898e-01 -8.12065251e+00  8.12065251e+00
  2.33105756e+01  2.33105756e+01  1.36871583e+02 -1.76081898e-01
 -8.12065251e+00  8.12065251e+00 -8.70380431e-01 -8.70380431e-01
  1.42473762e+02 -7.09472978e-02  1.60444657e+00 -1.60444657e+00
 -3.19473608e-01 -3.19473608e-01  1.42331028e+02 -1.17058224e-01
  2.25417129e-01 -2.25417129e-01 -1.84238993e-02 -1.84238993e-02
  1.42260550e+02  1.06082947e-02 -3.10846146e-03  3.10846146e-03
  3.76574842e-03  3.76574842e-03  1.42255374e+02  2.80804488e-04
 -2.89206800e-03  2.89206800e-03  3.76574842e-03  3.76574842e-03
  1.42255374e+02  2.80804488e-04  2.89206800e-03 -2.89206800e-03
 -1.84238993e-02 -1.84238993e-02  1.42260550e+02  1.06082947e-02
  3.10846146e-03 -3.10846146e-03 -3.19473608e-01 -3.19473608e-01
  1.42331028e+02 -1.17058224e-01 -2.25417129e-01  2.25417129e-01
 -8.70380431e-01 -8.70380431e-01  1.42473762e+02 -7.09472978e-02
 -1.60444657e+00  1.60444657e+00  2.33105756e+01  2.33105756e+01
  1.36871583e+02 -1.76081898e-01  8.12065251e+00 -8.12065251e+00
  2.33105756e+01  2.33105756e+01  1.36871583e+02  1.76081898e-01
  8.12065251e+00  8.12065251e+00 -8.70380431e-01 -8.70380431e-01
  1.42473762e+02  7.09472978e-02 -1.60444657e+00 -1.60444657e+00
 -3.19473608e-01 -3.19473608e-01  1.42331028e+02  1.17058224e-01
 -2.25417129e-01 -2.25417129e-01 -1.84238993e-02 -1.84238993e-02
  1.42260550e+02 -1.06082947e-02  3.10846146e-03  3.10846146e-03
  3.76574842e-03  3.76574842e-03  1.42255374e+02 -2.80804488e-04
  2.89206800e-03  2.89206800e-03  3.76574842e-03  3.76574842e-03
  1.42255374e+02 -2.80804488e-04 -2.89206800e-03 -2.89206800e-03
 -1.84238993e-02 -1.84238993e-02  1.42260550e+02 -1.06082947e-02
 -3.10846146e-03 -3.10846146e-03 -3.19473608e-01 -3.19473608e-01
  1.42331028e+02  1.17058224e-01  2.25417129e-01  2.25417129e-01
 -8.70380431e-01 -8.70380431e-01  1.42473762e+02  7.09472978e-02
  1.60444657e+00  1.60444657e+00  2.33105756e+01  2.33105756e+01
  1.36871583e+02  1.76081898e-01 -8.12065251e+00 -8.12065251e+00]
Number of entries 240 i.e., 40 elements x 6 Cauchy stress values
```

What we did:

- We defined a driver using QUEENS to run 4C simulation as a Python function call
- We provided a custom way of extracting simulations results
- Although we selected a single process, setting `num_procs>1` allows us to run a simulation in parallel without having to change anything in the interface!

The complexity of working with a sophisticated simulation code (without an API) is hidden from the user!

Instead, it is reduced to `fourc_driver.run(<arguments>)`

Now in the day-to-day of computational mechanics research, we often want to study convergence of a model depending on certain parameters. In this example, we'll have a look to nonconverging parameter combinations of $E \in [10, 10000]$ and $\nu \in [0, 0.5]$. Let's evaluate the 4C model on a grid of 4×21 points.

[53]:
```python
# Make it less verbose
import logging

logging.getLogger("distributed").setLevel(logging.WARN)

from queens.schedulers import Local
from queens.iterators import Grid
from queens.models import Simulation
from queens.global_settings import GlobalSettings
from queens.main import run_iterator
from queens.utils.io import load_result

n_grid_points_young = 4
n_grid_points_poisson_ratio = 21

grid_output = None
if __name__ == "__main__":
    pathlib.Path("grid").mkdir(exist_ok=True)
    with GlobalSettings(
        ␣
 ↪experiment_name=f"grid_{n_grid_points_young}x{n_grid_points_poisson_ratio}",
        output_dir="grid",
    ) as gs:
        scheduler = Local(
            gs.experiment_name,
            num_jobs=4,  # 4 simulations in parallel
            num_procs=1,  # each simulation uses 1 process
        )
        model = Simulation(scheduler, fourc_driver)

        iterator = Grid(
            model=model,
            parameters=parameters,
            global_settings=gs,
            result_description={"write_results": True},
            grid_design={
                "young": {
                    "num_grid_points": n_grid_points_young,
                    "axis_type": "log10",
                    "data_type": "FLOAT",
                },
                "poisson_ratio": {
                    "num_grid_points": n_grid_points_poisson_ratio,
                    "axis_type": "lin",
                    "data_type": "FLOAT",
                },
            },
        )

        run_iterator(iterator, gs)

        grid_output = load_result(gs.result_file("pickle"))
```

```
                                  *
                                * | *
                               * \|/ *
                          * * * \|O|/ * * *
                           \o\o\o|O|o/o/o/
                           (<><><>O<><><>)
                            '==========='

          :'#######::'##::::'##:'########:'########:'##::: ##::'######::
          '##… ##: ##:::: ##: ##…:: ##…:: ###:: ##:'##… ##:
           ##:::: ##: ##:::: ##: ##::::::: ##::::::: ####: ##: ##:::..::
           ##:::: ##: ##:::: ##: ######::: ######::: ## ## ##:. ######::
           ##:'## ##: ##:::: ##: ##…:::: ##…:::: ##. ####::… ##:
           ##:.. ##:: ##:::: ##: ##::::::: ##::::::: ##:. ###:'##::: ##:
          : ##### ##:. #######:: ########: ########: ##::. ##:. ######::
          :…:..:::…:::…::…::..::::..:::…:::


           A general purpose framework for Uncertainty Quantification,
            Physics-Informed Machine Learning, Bayesian Optimization,
                    Inverse Problems and Simulation Analytics


+--------------------------------------------------------------------------------
+
|                                     Local
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| self            : <queens.schedulers.local.Local object at 0x7284d43a3bd0>
|
| experiment_name : 'grid_4x21'
|
| num_jobs        : 4
|
| num_procs       : 1
|
| restart_workers : False
|
| verbose         : True
|
+------------------------------------------------------------------------------
+

To view the Dask dashboard open this link in your browser:
http://127.0.0.1:8787/status

+------------------------------------------------------------------------------
+
|                                  Simulation
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| self      : <queens.models.simulation.Simulation object at 0x7284d41cf010>
|
| scheduler : <queens.schedulers.local.Local object at 0x7284d43a3bd0>
|
| driver    : <queens_interfaces.fourc.driver.Fourc object at 0x7284cc5e5d10>
|
+------------------------------------------------------------------------------
+


2025-09-29 16:00:25,097 - distributed.worker - INFO -       Start worker at:
tcp://127.0.0.1:37897
2025-09-29 16:00:25,097 - distributed.worker - INFO -          Listening to:
tcp://127.0.0.1:37897
2025-09-29 16:00:25,097 - distributed.worker - INFO -           Worker name:
1
2025-09-29 16:00:25,097 - distributed.worker - INFO -          dashboard at:
127.0.0.1:39907
2025-09-29 16:00:25,097 - distributed.worker - INFO - Waiting to connect to:
tcp://127.0.0.1:35933
2025-09-29 16:00:25,097 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:25,097 - distributed.worker - INFO -               Threads:
1
2025-09-29 16:00:25,097 - distributed.worker - INFO -                Memory:
125.85 GiB
2025-09-29 16:00:25,097 - distributed.worker - INFO -       Local Directory:
/tmp/dask-scratch-space/worker-vo0_jt4y
2025-09-29 16:00:25,097 - distributed.worker - INFO -
-------------------------------------------------
```

```
2025-09-29 16:00:25,107 - distributed.worker - INFO - Starting Worker plugin
shuffle
2025-09-29 16:00:25,108 - distributed.worker - INFO -         Registered to:
tcp://127.0.0.1:35933
2025-09-29 16:00:25,108 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:25,108 - distributed.core - INFO - Starting established
connection to tcp://127.0.0.1:35933
2025-09-29 16:00:25,146 - distributed.worker - INFO -       Start worker at:
tcp://127.0.0.1:38495
2025-09-29 16:00:25,146 - distributed.worker - INFO -          Listening to:
tcp://127.0.0.1:38495
2025-09-29 16:00:25,146 - distributed.worker - INFO -           Worker name:
0
2025-09-29 16:00:25,146 - distributed.worker - INFO -          dashboard at:
127.0.0.1:34113
2025-09-29 16:00:25,146 - distributed.worker - INFO - Waiting to connect to:
tcp://127.0.0.1:35933
2025-09-29 16:00:25,146 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:25,146 - distributed.worker - INFO -               Threads:
1
2025-09-29 16:00:25,146 - distributed.worker - INFO -                Memory:
125.85 GiB
2025-09-29 16:00:25,146 - distributed.worker - INFO -       Local Directory:
/tmp/dask-scratch-space/worker-x3z8a6v7
2025-09-29 16:00:25,146 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:25,156 - distributed.worker - INFO - Starting Worker plugin
shuffle
2025-09-29 16:00:25,156 - distributed.worker - INFO -         Registered to:
tcp://127.0.0.1:35933
2025-09-29 16:00:25,156 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:25,157 - distributed.core - INFO - Starting established
connection to tcp://127.0.0.1:35933
2025-09-29 16:00:25,222 - distributed.worker - INFO -       Start worker at:
tcp://127.0.0.1:34547
2025-09-29 16:00:25,222 - distributed.worker - INFO -          Listening to:
tcp://127.0.0.1:34547
2025-09-29 16:00:25,222 - distributed.worker - INFO -           Worker name:
2
2025-09-29 16:00:25,222 - distributed.worker - INFO -          dashboard at:
127.0.0.1:34105
2025-09-29 16:00:25,222 - distributed.worker - INFO - Waiting to connect to:
tcp://127.0.0.1:35933
2025-09-29 16:00:25,222 - distributed.worker - INFO -
-------------------------------------------------
```

```
2025-09-29 16:00:25,222 - distributed.worker - INFO -               Threads:
1
2025-09-29 16:00:25,222 - distributed.worker - INFO -                Memory:
125.85 GiB
2025-09-29 16:00:25,222 - distributed.worker - INFO -       Local Directory:
/tmp/dask-scratch-space/worker-jwxfgxcn
2025-09-29 16:00:25,222 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:25,223 - distributed.worker - INFO -       Start worker at:
tcp://127.0.0.1:45413
2025-09-29 16:00:25,223 - distributed.worker - INFO -          Listening to:
tcp://127.0.0.1:45413
2025-09-29 16:00:25,223 - distributed.worker - INFO -           Worker name:
3
2025-09-29 16:00:25,223 - distributed.worker - INFO -          dashboard at:
127.0.0.1:39193
2025-09-29 16:00:25,223 - distributed.worker - INFO - Waiting to connect to:
tcp://127.0.0.1:35933
2025-09-29 16:00:25,223 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:25,223 - distributed.worker - INFO -               Threads:
1
2025-09-29 16:00:25,223 - distributed.worker - INFO -                Memory:
125.85 GiB
2025-09-29 16:00:25,223 - distributed.worker - INFO -       Local Directory:
/tmp/dask-scratch-space/worker-92r48tqu
2025-09-29 16:00:25,224 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:25,231 - distributed.worker - INFO - Starting Worker plugin
shuffle
2025-09-29 16:00:25,231 - distributed.worker - INFO -         Registered to:
tcp://127.0.0.1:35933
2025-09-29 16:00:25,231 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:25,231 - distributed.core - INFO - Starting established
connection to tcp://127.0.0.1:35933
2025-09-29 16:00:25,237 - distributed.worker - INFO - Starting Worker plugin
shuffle
2025-09-29 16:00:25,237 - distributed.worker - INFO -         Registered to:
tcp://127.0.0.1:35933
2025-09-29 16:00:25,237 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:25,238 - distributed.core - INFO - Starting established
connection to tcp://127.0.0.1:35933


+--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
```

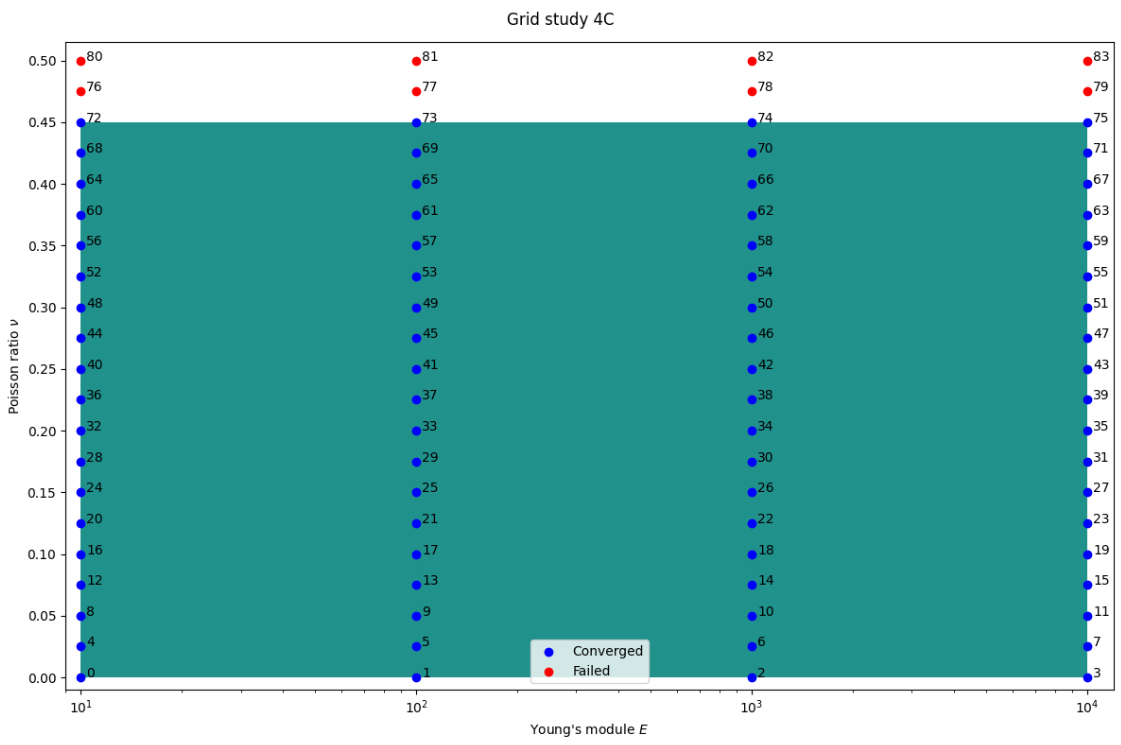
```
-------------------------+
|
Grid
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - |
| self              : <queens.iterators.grid.Grid object at 0x7284d4199b90>
|
| model             : <queens.models.simulation.Simulation object at
0x7284d41cf010>
|
| parameters        : <queens.parameters.parameters.Parameters object at
0x7284d4296090>
|
| global_settings   : <queens.global_settings.GlobalSettings object at
0x7284d4137e10>
|
| result_description : {'write_results': True}
|
| grid_design       : {'young': {'num_grid_points': 4, 'axis_type': 'log10',
'data_type': 'FLOAT'}, 'poisson_ratio': {'num_grid_points': 21, 'axis_type':
'lin', 'data_type': 'FLOAT'}} |
+--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-------------------------+


+--------------------------------------------------------------------------------
+
|                                 git information
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| commit hash       : 8b58f1998eecf6465541aeba0b8dd7871f1b10f8
|
| branch            : main
|
| clean working tree : False
|
+--------------------------------------------------------------------------------
+

Grid for experiment: grid_4x21

Starting Analysis…
```

```
 92%|     | 77/84 [00:16<00:01,  3.54it/s]The file '/home/a11brebu/queens-
experiments/grid_4x21/81/output/output-structure.pvd' does not exist!
The file '/home/a11brebu/queens-experiments/grid_4x21/80/output/output-
structure.pvd' does not exist!
 96%|     | 81/84 [00:18<00:00,  3.11it/s]The file '/home/a11brebu/queens-
experiments/grid_4x21/82/output/output-structure.pvd' does not exist!
The file '/home/a11brebu/queens-experiments/grid_4x21/83/output/output-
structure.pvd' does not exist!
100%|     | 84/84 [00:18<00:00,  3.99it/s]


+--------------------------------------------------------------------------------
+
|                          Batch summary for jobs 0 - 83
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 84
|
| number of parallel jobs       : 4
|
| number of procs               : 1
|
| total elapsed time            : 1.857e+01s
|
| average time per parallel job : 8.844e-01s
|
+--------------------------------------------------------------------------------
+


100%|     | 84/84 [00:18<00:00,  4.52it/s]


Time for CALCULATION: 18.61548399925232 s



2025-09-29 16:00:44,082 - distributed.worker - INFO - Stopping worker at
tcp://127.0.0.1:38495. Reason: nanny-close
2025-09-29 16:00:44,082 - distributed.worker - INFO - Removing Worker plugin
shuffle
2025-09-29 16:00:44,082 - distributed.worker - INFO - Stopping worker at
tcp://127.0.0.1:37897. Reason: nanny-close
2025-09-29 16:00:44,082 - distributed.worker - INFO - Stopping worker at
tcp://127.0.0.1:34547. Reason: nanny-close
2025-09-29 16:00:44,083 - distributed.worker - INFO - Removing Worker plugin
shuffle
2025-09-29 16:00:44,083 - distributed.worker - INFO - Stopping worker at
tcp://127.0.0.1:45413. Reason: nanny-close
```

```
2025-09-29 16:00:44,083 - distributed.worker - INFO - Removing Worker plugin
shuffle
2025-09-29 16:00:44,083 - distributed.worker - INFO - Removing Worker plugin
shuffle
2025-09-29 16:00:44,084 - distributed.core - INFO - Connection to
tcp://127.0.0.1:35933 has been closed.
2025-09-29 16:00:44,085 - distributed.core - INFO - Connection to
tcp://127.0.0.1:35933 has been closed.
2025-09-29 16:00:44,086 - distributed.core - INFO - Connection to
tcp://127.0.0.1:35933 has been closed.
2025-09-29 16:00:44,086 - distributed.core - INFO - Connection to
tcp://127.0.0.1:35933 has been closed.
2025-09-29 16:00:44,086 - distributed.nanny - INFO - Worker closed
2025-09-29 16:00:44,089 - distributed.nanny - INFO - Worker closed
2025-09-29 16:00:44,089 - distributed.nanny - INFO - Worker closed
2025-09-29 16:00:44,089 - distributed.nanny - INFO - Worker closed
```

[54]:
```
# Let's plot the data
grid_points = grid_output["input_data"]
cauchy_stresses = grid_output["raw_output_data"]["result"]

import matplotlib.pyplot as plt

converged_samples = []
failed_samples = []
failed = []
for input_sample, cauchy_output in zip(grid_points, cauchy_stresses,␣
 ↪strict=True):
    if np.isnan(cauchy_output).any():
        failed_samples.append(input_sample)
        failed.append(1)
    else:
        converged_samples.append(input_sample)
        failed.append(0)

converged_samples = np.array(converged_samples)
failed_samples = np.array(failed_samples)
failed = np.array(failed)

fig, ax = plt.subplots(figsize=(12, 8))
ax.contourf(
    grid_points[:, 0].reshape(21, 4),
    grid_points[:, 1].reshape(21, 4),
    failed.reshape(21, 4),
    levels=[0, 0.000001],
)
```

```
for i, gp in enumerate(grid_points):
    plt.text(gp[0] * 1.04, gp[1], i)

ax.scatter(converged_samples[:, 0], converged_samples[:, 1], c="b",␣
 ↪label="Converged")
ax.scatter(failed_samples[:, 0], failed_samples[:, 1], c="r", label="Failed")

ax.set_xlim(10 * 0.9, 10000 * 1.2)
ax.set_yticks(np.linspace(0, 0.5, 11))
ax.set_ylim(-0.01, 0.515)
ax.set_xlabel("Young's module $E$")
ax.set_xscale("log")
ax.set_ylabel("Poisson ratio $\\nu$")
fig.suptitle("Grid study 4C")
ax.legend()
fig.tight_layout()
plt.show()
```



Each point represents one simulation, i.e., one 4C run with different parameters. The number next to it is the job id, so the unique identifier for this simulation for QUEENS. The color of the points: - blue: The simulation ran through and returned a value. - red: The simulation failed and NaN was returned.

A colored background indicates the region of the parameter space leads to converging results.

Previously, we called 4C using the driver directly. However, when using QUEENS iterators, this won't work anymore. Instead, we need a `Model`. The difference is that a driver handle a single simulation, whereas a model accepts a list of relevant input samples / configuration.

When working with drivers, the `Simulation` model is used. Besides a driver, this requires a `Scheduler`. As the name suggests, schedulers schedule evaluations of the driver. The scheduler also handles the evaluation of multiple drivers calls at the same time. Keep in mind that each driver call can itself be parallelized with MPI, such that QUEENS natively supports nested parallelism!

Since we are doing our computations locally, we use a `Local` scheduler. However, there is also a `Cluster` scheduler which is able to submit jobs on high performance clusters using `SLURM`, `PBS` etc.

The scheduler also creates the required folder structure for an experiment. The default value is set to `~/queens-experiments/<experiment_name>`. For the grid iterator, it looks like this:

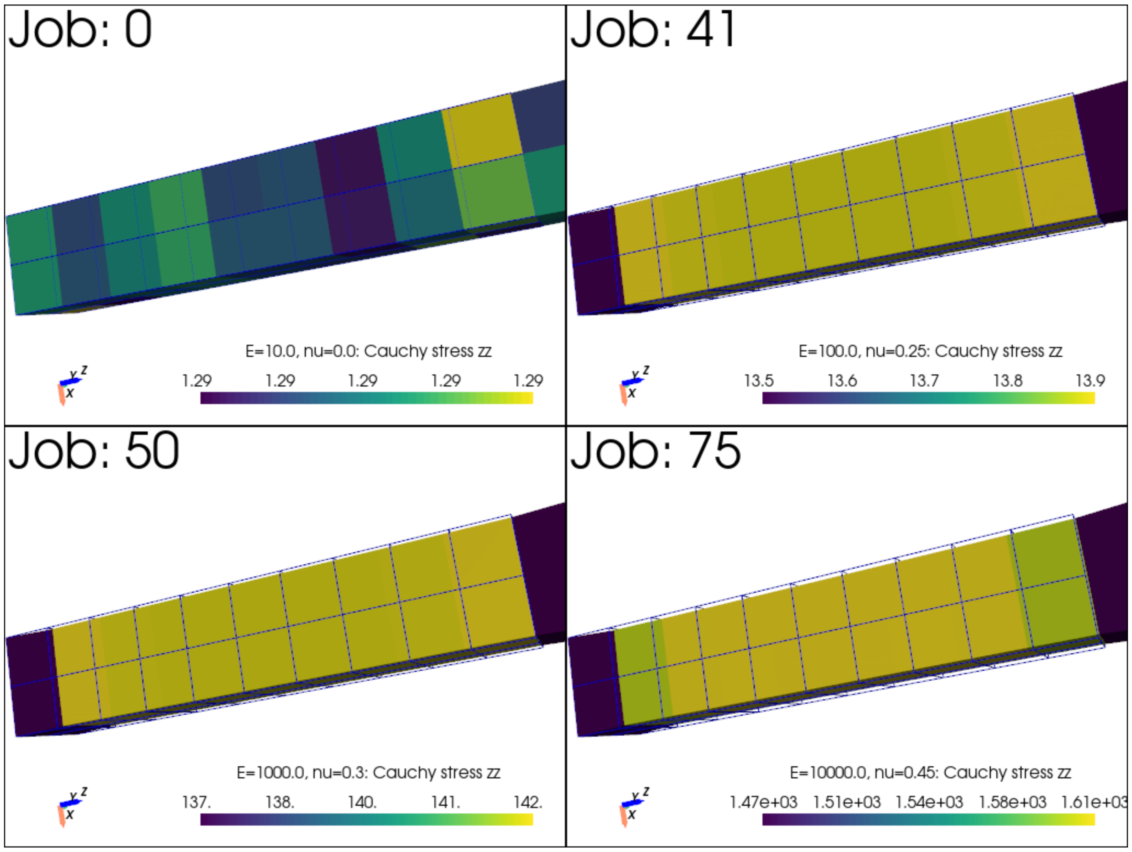
```
grid_4x21/
  0
  1
...
  80
  81
  82
  83
  9
  template.4C.yaml
```

Each number is a job with its output:

```
10
  input_file.yaml
  jobscript.sh
  metadata.yaml
  output
      output.control
      output.log
      output.mesh.structure.s0
      output-structure.pvd
      output-vtk-files
          structure-00000-0.vtu
          structure-00000.pvtu
          structure-00001-0.vtu
          structure-00001.pvtu
```

Once more, have a look at the files. In particular look at the log files of the failed simulations.

[55]:
```python
# Let's look at some simulations outputs!
job_ids = [0, 41, 50, 75]

plotter = pv.Plotter(shape=(2, 2))

for i in range(2):
    for j in range(2):
        plotter.subplot(i, j)
        job_id = job_ids[i * 2 + j]
        file_path = (
            home
            / f"queens-experiments/grid_4x21/{job_id}/output/output-vtk-files/
 ↪structure-00001.pvtu"
        )
        plotter.add_text(f"Job: {job_id}")
        plot_results(
            file_path,
            plotter,
            f"E={grid_points[job_id][0]}, nu={np.
 ↪round(grid_points[job_id][1],decimals=4)}: Cauchy stress zz\n",
        )
plotter.show()
```


Job: 0
E=10.0, nu=0.0: Cauchy stress zz
1.29 1.29 1.29 1.29 1.29
Job: 41
E=100.0, nu=0.25: Cauchy stress zz
13.5 13.6 13.7 13.8 13.9
Job: 50
E=1000.0, nu=0.3: Cauchy stress zz
137. 138. 140. 141. 142.
Job: 75
E=10000.0, nu=0.45: Cauchy stress zz
1.47e+03 1.51e+03 1.54e+03 1.58e+03 1.61e+03
y z x

In summary, we employed QUEENS to do some simulation analytics of a 4C model. We identified failed simulations, looked at the outputs and extracted data.

4 Calibration

Now, let's imagine we have some experimental measurements of the Cauchy stresses at the element centers σ_{ex} of a real life cantilever. We can now use this, to calibrate our parameters E and ν from experiment.

> Note: We assume that the 4C model is able to predict the real-word accurately, i.e, we have the right boundary conditions, constitutive law etc.

To do so, we'll use deterministic optimization to minimize the difference between simulated and measured stresses:

$$(E, \nu) \in \underset{E,\nu}{\operatorname{argmin}} \frac{1}{n_{\text{ele}}} \sum_{e=1}^{n_{\text{ele}}} (\sigma_e(E,\nu) - \sigma_{\text{ex},e}) : (\sigma_e(E,\nu) - \sigma_{\text{ex},e})$$

where $\sigma_e(E,\nu)$ are the elementwise stresses obtained from the 4C model evaluated at E, ν, where the supscript e is the element id.

Before we start the optimization, we need to change our parameter space. As we previously identified, the simulation model does not converge for $\nu > 0.45$. So in order to make meaningful predictions, let's change the parameter space.

[56]:
```python
# Reduced parameter space for the poisson ratio
poisson_ratio = Uniform(0, 0.45)

parameters = Parameters(young=young, poisson_ratio=poisson_ratio)
```

[57]:
```python
# Let's implement the loss function

from queens.models._model import Model


class LossFunction(Model):
    def __init__(self, experimental_cauchy_stresses, forward_model):
        self.experimental_cauchy_stresses = experimental_cauchy_stresses
        self.forward_model = forward_model
        super().__init__()

    def _evaluate(self, samples):
        element_cauchy_stresses = self.forward_model.evaluate(samples)["result"]

        difference = element_cauchy_stresses - np.tile(
            self.experimental_cauchy_stresses, (len(samples), 1)
        )
        squared_norm = (difference**2).sum(axis=1) + (difference[:, 3:] ** 2).
↪sum(
            axis=1
        )  # the output is only a symmetrical tensor, so we have to add the␣
↪offdiagonal terms once more
```

```
        squared_norm /= 40  # 40 elements

        return {"result": squared_norm.reshape(-1)}

    def grad(self, samples, upstream_gradient):
        pass


from experimental_data import experimental_elementwise_cauchy_stress
```

Ok, now let's set up the optimization using the L-BFGS-B algorithm.

[58]:
```
from scipy.optimize import Bounds
from queens.iterators import Optimization

optimum = None

if __name__ == "__main__":
    pathlib.Path("optimization").mkdir(exist_ok=True)
    with GlobalSettings(
        experiment_name="optimization_young_poisson_ratio",␣
 ↪output_dir="optimization"
    ) as gs:
        scheduler = Local(
            gs.experiment_name,
            num_jobs=4,  # 4 simulations in parallel
            num_procs=1,  # each simulation uses 1 process
        )
        custom_data_processor = CustomDataProcessor(
            field_name="element_cauchy_stresses_xyz",  # Name of the field in␣
 ↪the pvtu file
            file_name_identifier="output-structure.pvd",  # Name of the file in␣
 ↪the output folder
            failure_value=np.nan * np.ones(40 * 6),  # NaN for all elements
            files_to_be_deleted_regex_lst=[
                "output.control",
                "*.mesh.s0",
                "output-vtk-files/*",
            ],  # Delete these files after we extracted the data.
        )
        fourc_driver = Fourc(
            parameters=parameters,
            input_templates=input_template,
            executable=fourc_executable,
            data_processor=custom_data_processor,  # tell the driver to extract␣
 ↪the data
        )
```

```python
        model = Simulation(scheduler, fourc_driver)
        loss_function = LossFunction(experimental_elementwise_cauchy_stress, model)
        optimization = Optimization(
            loss_function,
            parameters,
            gs,
            initial_guess=[200, 0.4],
            result_description={"write_results": True},
            objective_and_jacobian=True,
            bounds=Bounds([10, 0], [10000, 0.45]),
            algorithm="L-BFGS-B",
            max_feval=1000,
        )

        run_iterator(optimization, gs)

        optimum = optimization.solution.x
```

```
                                  *
                                * | *
                               * \|/ *
                          * * * \|O|/ * * *
                           \o\o\o|O|o/o/o/
                           (<><><>O<><><>)
                            '==========='

          :'#######::'##::::'##:'########:'########:'##::: ##::'######::
          '##… ##: ##:::: ##: ##…:: ##…:: ###:: ##:'##… ##:
           ##:::: ##: ##:::: ##: ##::::::: ##::::::: ####: ##: ##:::..::
           ##:::: ##: ##:::: ##: ######::: ######::: ## ## ##:. ######::
           ##:'## ##: ##:::: ##: ##…:::: ##…:::: ##. ####::… ##:
           ##:.. ##:: ##:::: ##: ##::::::: ##::::::: ##:. ###:'##::: ##:
          : ##### ##:. #######:: ########: ########: ##::. ##:. ######::
          :…:..:::…:::…::…::..::::..:::…:::


           A general purpose framework for Uncertainty Quantification,
            Physics-Informed Machine Learning, Bayesian Optimization,
                    Inverse Problems and Simulation Analytics


+---------------------------------------------------------------------------------
+
|                                    Local
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
-|
| self            : <queens.schedulers.local.Local object at 0x72849426ca90>
|
| experiment_name : 'optimization_young_poisson_ratio'
|
| num_jobs        : 4
|
| num_procs       : 1
|
| restart_workers : False
|
| verbose         : True
|
+--------------------------------------------------------------------------------
+

To view the Dask dashboard open this link in your browser:
http://127.0.0.1:8787/status

+--------------------------------------------------------------------------------
+
|                                  Simulation
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| self      : <queens.models.simulation.Simulation object at 0x728494230f50>
|
| scheduler : <queens.schedulers.local.Local object at 0x72849426ca90>
|
| driver    : <queens_interfaces.fourc.driver.Fourc object at 0x728494246fd0>
|
+--------------------------------------------------------------------------------
+


+--------------------------------------------------------------------------------
-----------------+
|                                          Optimization
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - |
| self                  : <queens.iterators.optimization.Optimization object at
0x7284cc29b890> |
| model                 : <__main__.LossFunction object at 0x7284941baed0>
|
| parameters            : <queens.parameters.parameters.Parameters object at
0x728494222590>    |
| global_settings       : <queens.global_settings.GlobalSettings object at
```

```
0x7284cc142810>      |
| initial_guess          : [200, 0.4]
|
| result_description     : {'write_results': True}
|
| verbose_output         : False
|
| bounds                 : Bounds(array([10,  0]), array([1.0e+04, 4.5e-01]))
|
| constraints            : None
|
| max_feval              : 1000
|
| algorithm              : 'L-BFGS-B'
|
| jac_method             : '2-point'
|
| jac_rel_step           : None
|
| objective_and_jacobian : True
|
+--------------------------------------------------------------------------------
-----------------+


+--------------------------------------------------------------------------------
+
|                                 git information
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| commit hash        : 8b58f1998eecf6465541aeba0b8dd7871f1b10f8
|
| branch             : main
|
| clean working tree : False
|
+--------------------------------------------------------------------------------
+

Optimization for experiment: optimization_young_poisson_ratio

Starting Analysis…

Initialize Optimization run.
Welcome to Optimization core run.
2025-09-29 16:00:47,549 - distributed.worker - INFO -       Start worker at:
```

```
tcp://127.0.0.1:45897
2025-09-29 16:00:47,549 - distributed.worker - INFO -          Listening to:
tcp://127.0.0.1:45897
2025-09-29 16:00:47,549 - distributed.worker - INFO -           Worker name:
0
2025-09-29 16:00:47,549 - distributed.worker - INFO -          dashboard at:
127.0.0.1:34139
2025-09-29 16:00:47,549 - distributed.worker - INFO - Waiting to connect to:
tcp://127.0.0.1:45879
2025-09-29 16:00:47,549 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,549 - distributed.worker - INFO -               Threads:
1
2025-09-29 16:00:47,549 - distributed.worker - INFO -                Memory:
125.85 GiB
2025-09-29 16:00:47,549 - distributed.worker - INFO -       Local Directory:
/tmp/dask-scratch-space/worker-shk2mea_
2025-09-29 16:00:47,549 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,551 - distributed.worker - INFO -       Start worker at:
tcp://127.0.0.1:44645
2025-09-29 16:00:47,551 - distributed.worker - INFO -          Listening to:
tcp://127.0.0.1:44645
2025-09-29 16:00:47,551 - distributed.worker - INFO -           Worker name:
1
2025-09-29 16:00:47,551 - distributed.worker - INFO -          dashboard at:
127.0.0.1:37657
2025-09-29 16:00:47,551 - distributed.worker - INFO - Waiting to connect to:
tcp://127.0.0.1:45879
2025-09-29 16:00:47,551 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,551 - distributed.worker - INFO -               Threads:
1
2025-09-29 16:00:47,551 - distributed.worker - INFO -                Memory:
125.85 GiB
2025-09-29 16:00:47,551 - distributed.worker - INFO -       Local Directory:
/tmp/dask-scratch-space/worker-9j51ytqs
2025-09-29 16:00:47,551 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,553 - distributed.worker - INFO -       Start worker at:
tcp://127.0.0.1:44497
2025-09-29 16:00:47,553 - distributed.worker - INFO -          Listening to:
tcp://127.0.0.1:44497
2025-09-29 16:00:47,553 - distributed.worker - INFO -           Worker name:
2
2025-09-29 16:00:47,553 - distributed.worker - INFO -          dashboard at:
127.0.0.1:36043
2025-09-29 16:00:47,554 - distributed.worker - INFO - Waiting to connect to:
```

```
tcp://127.0.0.1:45879
2025-09-29 16:00:47,554 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,554 - distributed.worker - INFO -               Threads:
1
2025-09-29 16:00:47,554 - distributed.worker - INFO -                Memory:
125.85 GiB
2025-09-29 16:00:47,554 - distributed.worker - INFO -       Local Directory:
/tmp/dask-scratch-space/worker-o8fs2a8_
2025-09-29 16:00:47,554 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,566 - distributed.worker - INFO - Starting Worker plugin
shuffle
2025-09-29 16:00:47,566 - distributed.worker - INFO -         Registered to:
tcp://127.0.0.1:45879
2025-09-29 16:00:47,566 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,567 - distributed.worker - INFO -       Start worker at:
tcp://127.0.0.1:41229
2025-09-29 16:00:47,567 - distributed.worker - INFO -          Listening to:
tcp://127.0.0.1:41229
2025-09-29 16:00:47,567 - distributed.worker - INFO -           Worker name:
3
2025-09-29 16:00:47,567 - distributed.worker - INFO -          dashboard at:
127.0.0.1:36067
2025-09-29 16:00:47,567 - distributed.worker - INFO - Waiting to connect to:
tcp://127.0.0.1:45879
2025-09-29 16:00:47,567 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,567 - distributed.worker - INFO -               Threads:
1
2025-09-29 16:00:47,567 - distributed.worker - INFO -                Memory:
125.85 GiB
2025-09-29 16:00:47,567 - distributed.worker - INFO -       Local Directory:
/tmp/dask-scratch-space/worker-4comopge
2025-09-29 16:00:47,567 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,567 - distributed.core - INFO - Starting established
connection to tcp://127.0.0.1:45879
2025-09-29 16:00:47,568 - distributed.worker - INFO - Starting Worker plugin
shuffle
2025-09-29 16:00:47,568 - distributed.worker - INFO -         Registered to:
tcp://127.0.0.1:45879
2025-09-29 16:00:47,568 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,569 - distributed.core - INFO - Starting established
connection to tcp://127.0.0.1:45879
2025-09-29 16:00:47,570 - distributed.worker - INFO - Starting Worker plugin
```

```
shuffle
2025-09-29 16:00:47,571 - distributed.worker - INFO -         Registered to:
tcp://127.0.0.1:45879
2025-09-29 16:00:47,571 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,572 - distributed.core - INFO - Starting established
connection to tcp://127.0.0.1:45879
2025-09-29 16:00:47,584 - distributed.worker - INFO - Starting Worker plugin
shuffle
2025-09-29 16:00:47,585 - distributed.worker - INFO -         Registered to:
tcp://127.0.0.1:45879
2025-09-29 16:00:47,585 - distributed.worker - INFO -
-------------------------------------------------
2025-09-29 16:00:47,586 - distributed.core - INFO - Starting established
connection to tcp://127.0.0.1:45879
 33%|        | 1/3 [00:01<00:03,  1.96s/it]


+--------------------------------------------------------------------------------
+
|                         Batch summary for jobs 0 - 2
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 3
|
| number of parallel jobs       : 4
|
| number of procs               : 1
|
| total elapsed time            : 1.977e+00s
|
| average time per parallel job : 1.977e+00s
|
+--------------------------------------------------------------------------------
+


100%|     | 3/3 [00:01<00:00,  1.52it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [200.    0.4]


100%|     | 3/3 [00:01<00:00,  1.79it/s]


+--------------------------------------------------------------------------------
+
|                         Batch summary for jobs 3 - 5
```

```
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs              : 3
|
| number of parallel jobs     : 4
|
| number of procs             : 1
|
| total elapsed time          : 1.747e+00s
|
| average time per parallel job : 1.747e+00s
|
+--------------------------------------------------------------------------------
+


100%|     | 3/3 [00:01<00:00,  1.72it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [192.96485945   0.        ]


 33%|        | 1/3 [00:00<00:01,  1.23it/s]


+--------------------------------------------------------------------------------
+
|                         Batch summary for jobs 6 - 8
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs              : 3
|
| number of parallel jobs     : 4
|
| number of procs             : 1
|
| total elapsed time          : 8.315e-01s
|
| average time per parallel job : 8.315e-01s
|
+--------------------------------------------------------------------------------
+


100%|     | 3/3 [00:00<00:00,  3.60it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [192.63505078   0.        ]
 33%|        | 1/3 [00:00<00:01,  1.33it/s]

+-------------------------------------------------------------------------------
+
|                         Batch summary for jobs 9 - 11
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs               : 3
|
| number of parallel jobs      : 4
|
| number of procs              : 1
|
| total elapsed time           : 7.705e-01s
|
| average time per parallel job : 7.705e-01s
|
+-------------------------------------------------------------------------------
+

100%|     | 3/3 [00:00<00:00,  3.88it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [191.31581613   0.        ]

 33%|        | 1/3 [00:00<00:01,  1.31it/s]

+-------------------------------------------------------------------------------
+
|                        Batch summary for jobs 12 - 14
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs               : 3
|
| number of parallel jobs      : 4
|
| number of procs              : 1
|
| total elapsed time           : 7.895e-01s
|
| average time per parallel job : 7.895e-01s
|
+-------------------------------------------------------------------------------
```

```
+

100%|     | 3/3 [00:00<00:00,  3.79it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [186.03887753   0.        ]

 33%|        | 1/3 [00:00<00:01,  1.24it/s]

+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 15 - 17
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs               : 3
|
| number of parallel jobs      : 4
|
| number of procs              : 1
|
| total elapsed time           : 8.341e-01s
|
| average time per parallel job : 8.341e-01s
|
+--------------------------------------------------------------------------------
+

100%|     | 3/3 [00:00<00:00,  3.59it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [143.06999262   0.        ]

 33%|        | 1/3 [00:00<00:01,  1.31it/s]

+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 18 - 20
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs               : 3
|
| number of parallel jobs      : 4
|
| number of procs              : 1
```

```
|
| total elapsed time            : 8.239e-01s
|
| average time per parallel job : 8.239e-01s
|
+--------------------------------------------------------------------------------
+


100%|     | 3/3 [00:00<00:00,  3.63it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [1.42356700e+02 2.24250142e-02]


 33%|        | 1/3 [00:00<00:01,  1.36it/s]


+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 21 - 23
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 3
|
| number of parallel jobs       : 4
|
| number of procs               : 1
|
| total elapsed time            : 7.586e-01s
|
| average time per parallel job : 7.586e-01s
|
+--------------------------------------------------------------------------------
+


100%|     | 3/3 [00:00<00:00,  3.94it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [1.39503532e+02 1.12125071e-01]


 33%|        | 1/3 [00:00<00:01,  1.35it/s]


+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 24 - 26
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
-|
| number of jobs              : 3
|
| number of parallel jobs     : 4
|
| number of procs             : 1
|
| total elapsed time          : 7.589e-01s
|
| average time per parallel job : 7.589e-01s
|
+--------------------------------------------------------------------------------
+

100%|     | 3/3 [00:00<00:00,  3.94it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [128.75644565   0.45      ]

 33%|        | 1/3 [00:00<00:01,  1.30it/s]

+--------------------------------------------------------------------------------
+
|                         Batch summary for jobs 27 - 29
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs              : 3
|
| number of parallel jobs     : 4
|
| number of procs             : 1
|
| total elapsed time          : 8.183e-01s
|
| average time per parallel job : 8.183e-01s
|
+--------------------------------------------------------------------------------
+

100%|     | 3/3 [00:00<00:00,  3.66it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [138.33709776   0.14879629]

 33%|        | 1/3 [00:00<00:01,  1.04it/s]
```

```
+--------------------------------------------------------------------------------
+
|                         Batch summary for jobs 30 - 32
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs              : 3
|
| number of parallel jobs     : 4
|
| number of procs             : 1
|
| total elapsed time          : 9.717e-01s
|
| average time per parallel job : 9.717e-01s
|
+--------------------------------------------------------------------------------
+
```

```
100%|     | 3/3 [00:00<00:00,  3.08it/s]
```

```
The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [129.76606369   0.45      ]
```

```
 33%|        | 1/3 [00:00<00:01,  1.27it/s]
```

```
+--------------------------------------------------------------------------------
+
|                         Batch summary for jobs 33 - 35
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs              : 3
|
| number of parallel jobs     : 4
|
| number of procs             : 1
|
| total elapsed time          : 8.016e-01s
|
| average time per parallel job : 8.016e-01s
|
+--------------------------------------------------------------------------------
+
100%|      | 3/3 [00:00<00:00,  3.73it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [133.82709398   0.30728707]


 33%|        | 1/3 [00:00<00:01,  1.30it/s]


+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 36 - 38
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 3
|
| number of parallel jobs       : 4
|
| number of procs               : 1
|
| total elapsed time            : 7.825e-01s
|
| average time per parallel job : 7.825e-01s
|
+--------------------------------------------------------------------------------
+


100%|      | 3/3 [00:00<00:00,  3.82it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [116.36544161   0.45      ]


 33%|        | 1/3 [00:00<00:01,  1.24it/s]


+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 39 - 41
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 3
|
| number of parallel jobs       : 4
|
| number of procs               : 1
|
| total elapsed time            : 8.337e-01s
```

```
|
| average time per parallel job : 8.337e-01s
|
+------------------------------------------------------------------------------
+


100%|     | 3/3 [00:00<00:00,  3.59it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [129.56218349   0.3421439 ]


 33%|        | 1/3 [00:00<00:01,  1.29it/s]


+------------------------------------------------------------------------------
+
|                        Batch summary for jobs 42 - 44
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 3
|
| number of parallel jobs       : 4
|
| number of procs               : 1
|
| total elapsed time            : 8.021e-01s
|
| average time per parallel job : 8.021e-01s
|
+------------------------------------------------------------------------------
+


100%|     | 3/3 [00:00<00:00,  3.73it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [120.31396659   0.45      ]


 33%|        | 1/3 [00:00<00:01,  1.25it/s]


+------------------------------------------------------------------------------
+
|                        Batch summary for jobs 45 - 47
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 3
```

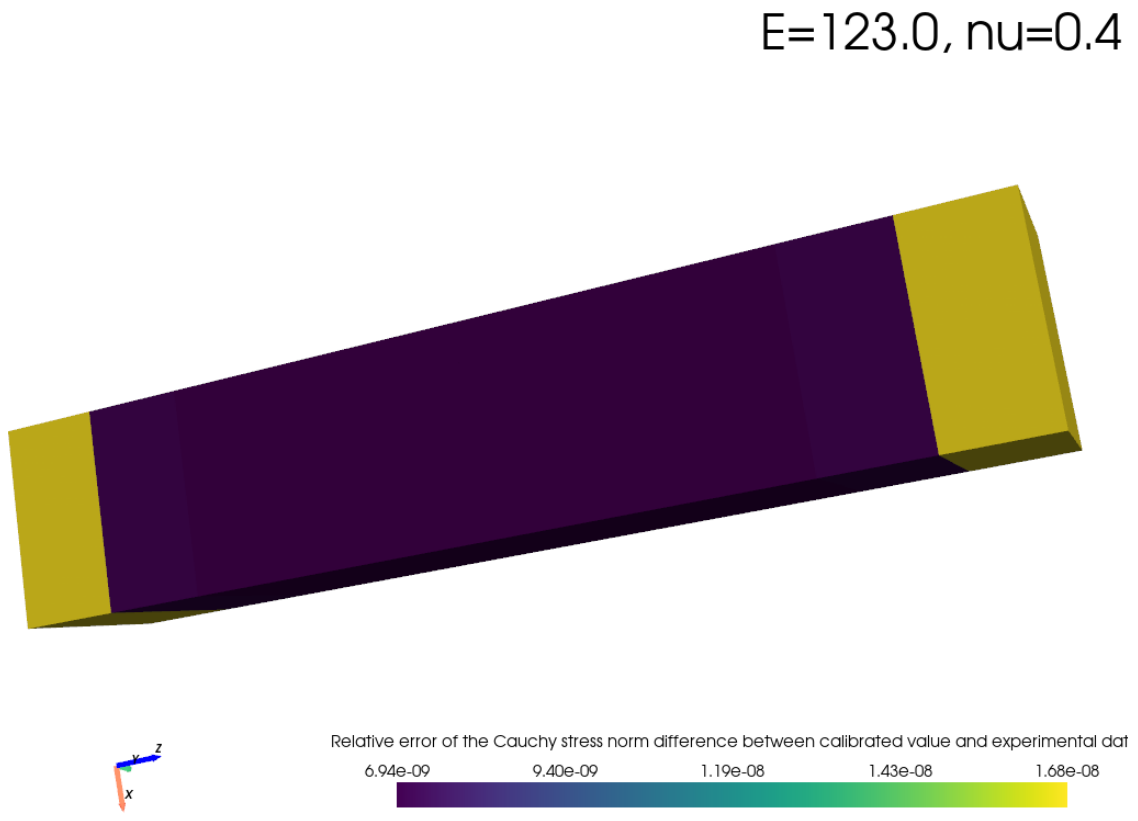
```
|
| number of parallel jobs       : 4
|
| number of procs               : 1
|
| total elapsed time            : 8.140e-01s
|
| average time per parallel job : 8.140e-01s
|
+--------------------------------------------------------------------------------
+

100%|     | 3/3 [00:00<00:00,  3.68it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [125.7515413    0.38658501]


 33%|        | 1/3 [00:00<00:01,  1.34it/s]


+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 48 - 50
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 3
|
| number of parallel jobs       : 4
|
| number of procs               : 1
|
| total elapsed time            : 7.678e-01s
|
| average time per parallel job : 7.678e-01s
|
+--------------------------------------------------------------------------------
+

100%|     | 3/3 [00:00<00:00,  3.89it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [120.31407392   0.4074305 ]


 33%|        | 1/3 [00:00<00:01,  1.25it/s]


+--------------------------------------------------------------------------------
```

```
+
|                        Batch summary for jobs 51 - 53
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 3
|
| number of parallel jobs       : 4
|
| number of procs               : 1
|
| total elapsed time            : 8.400e-01s
|
| average time per parallel job : 8.400e-01s
|
+--------------------------------------------------------------------------------
+

100%|     | 3/3 [00:00<00:00,  3.56it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [123.1225246    0.39666381]

 33%|        | 1/3 [00:00<00:01,  1.33it/s]

+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 54 - 56
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 3
|
| number of parallel jobs       : 4
|
| number of procs               : 1
|
| total elapsed time            : 7.766e-01s
|
| average time per parallel job : 7.766e-01s
|
+--------------------------------------------------------------------------------
+

100%|     | 3/3 [00:00<00:00,  3.85it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [122.77541054   0.40501085]

 33%|        | 1/3 [00:00<00:01,  1.12it/s]

+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 57 - 59
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs               : 3
|
| number of parallel jobs      : 4
|
| number of procs              : 1
|
| total elapsed time           : 9.206e-01s
|
| average time per parallel job : 9.206e-01s
|
+--------------------------------------------------------------------------------
+

100%|      | 3/3 [00:00<00:00,  3.25it/s]
The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [122.98363062   0.40000379]

 33%|        | 1/3 [00:00<00:01,  1.24it/s]

+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 60 - 62
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs               : 3
|
| number of parallel jobs      : 4
|
| number of procs              : 1
|
| total elapsed time           : 8.460e-01s
|
| average time per parallel job : 8.460e-01s
|
+--------------------------------------------------------------------------------
+

100%|     | 3/3 [00:00<00:00,  3.54it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [122.99965387   0.40000297]

 33%|        | 1/3 [00:00<00:01,  1.27it/s]

+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 63 - 65
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 3
|
| number of parallel jobs       : 4
|
| number of procs               : 1
|
| total elapsed time            : 8.061e-01s
|
| average time per parallel job : 8.061e-01s
|
+--------------------------------------------------------------------------------
+

100%|     | 3/3 [00:00<00:00,  3.71it/s]

The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [123.00002037   0.39999982]

 33%|        | 1/3 [00:01<00:02,  1.09s/it]

+--------------------------------------------------------------------------------
+
|                        Batch summary for jobs 66 - 68
|
|- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-|
| number of jobs                : 3
|
| number of parallel jobs       : 4
|
| number of procs              : 1
|
| total elapsed time           : 1.117e+00s
|
| average time per parallel job : 1.117e+00s
|
+--------------------------------------------------------------------------------
+

100%|     | 3/3 [00:01<00:00,  2.68it/s]
The intermediate, iterated parameters ['young', 'poisson_ratio'] are:
        [122.9999999    0.39999999]
Optimization took 2.209550E+01 seconds.
The optimum:
        [122.9999999    0.39999999]

Time for CALCULATION: 22.117238998413086 s



2025-09-29 16:01:09,930 - distributed.worker - INFO - Stopping worker at
tcp://127.0.0.1:45897. Reason: nanny-close
2025-09-29 16:01:09,931 - distributed.worker - INFO - Removing Worker plugin
shuffle
2025-09-29 16:01:09,931 - distributed.worker - INFO - Stopping worker at
tcp://127.0.0.1:44497. Reason: nanny-close
2025-09-29 16:01:09,932 - distributed.worker - INFO - Removing Worker plugin
shuffle
2025-09-29 16:01:09,932 - distributed.worker - INFO - Stopping worker at
tcp://127.0.0.1:41229. Reason: nanny-close
2025-09-29 16:01:09,933 - distributed.worker - INFO - Stopping worker at
tcp://127.0.0.1:44645. Reason: nanny-close
2025-09-29 16:01:09,933 - distributed.worker - INFO - Removing Worker plugin
shuffle
2025-09-29 16:01:09,933 - distributed.worker - INFO - Removing Worker plugin
shuffle
2025-09-29 16:01:09,934 - distributed.core - INFO - Connection to
tcp://127.0.0.1:45879 has been closed.
2025-09-29 16:01:09,934 - distributed.core - INFO - Connection to
tcp://127.0.0.1:45879 has been closed.
2025-09-29 16:01:09,935 - distributed.core - INFO - Connection to
tcp://127.0.0.1:45879 has been closed.
2025-09-29 16:01:09,935 - distributed.core - INFO - Connection to
tcp://127.0.0.1:45879 has been closed.
2025-09-29 16:01:09,937 - distributed.nanny - INFO - Worker closed
2025-09-29 16:01:09,937 - distributed.nanny - INFO - Worker closed
2025-09-29 16:01:09,939 - distributed.nanny - INFO - Worker closed
```

```
2025-09-29 16:01:09,939 - distributed.nanny - INFO - Worker closed
```

[59]:
```
# Let's run the optimum:

fourc_driver.data_processor = None

fourc_driver.run(
    optimum,
    experiment_dir=pathlib.Path("."),
    job_id="calibrated_value",
    num_procs=1,
    experiment_name=None,
)


def tensor_norm(tensor_data):
    data = np.hstack(
        (tensor_data.tolist(), tensor_data[:, 3:].tolist())
    )  # duplicate offdiagonal terms
    return np.sqrt((data**2).sum(axis=1)).reshape(-1, 1)


def plot_tensor_norm(output_mesh, field_name, data, plotter, color_bar_title):
    output_mesh[field_name] = data

    plotter.add_mesh(
        output_mesh,
        scalars=field_name,
        scalar_bar_args={
            "title": color_bar_title,
            "title_font_size": 15,
            "label_font_size": 15,
        },
    )
    plotter.add_axes(line_width=5)
    plotter.camera_position = [
        (1.1321899035097993, -6.851600196807601, 2.7649096132703574),
        (0.0, 0.0, 0.27499999999999999),
        (-0.97637930372977, -0.08995062285804697, 0.19644933366041056),
    ]


output_mesh = pv.read("calibrated_value/output/output-vtk-files/structure-00001.
↪pvtu")
output_calibrated_value = output_mesh["element_cauchy_stresses_xyz"]
difference = output_calibrated_value - experimental_elementwise_cauchy_stress.
↪reshape(
    -1, 6
)
relative_error = tensor_norm(difference) / tensor_norm(
    experimental_elementwise_cauchy_stress.reshape(-1, 6)
)

plotter = pv.Plotter()
plot_tensor_norm(
    output_mesh,
    "relative",
    relative_error,
    plotter,
    "Relative error of the Cauchy stress norm difference between calibrated␣
  ↪value and experimental data\n",
)
plotter.add_text(
    f"E={np.round(optimum[0], decimals=2)}, nu={np.round(optimum[1],␣
  ↪decimals=2)}",
    "upper_right",
)
plotter.show()
```

```
/tmp/ipykernel_1340701/163401757.py:58: PyVistaDeprecationWarning:
../../../../../tmp/ipykernel_1340701/163401757.py:58: Argument 'position' must
be passed as a keyword argument to function 'BasePlotter.add_text'.
From version 0.50, passing this as a positional argument will result in a
TypeError.
  plotter.add_text(
```


E=123.0, nu=0.4
Relative error of the Cauchy stress norm difference between calibrated value and experimental dat
6.94e-09
9.40e-09
1.19e-08
1.43e-08
1.68e-08
z
y
x

We successfully conducted a calibration of the Young's modulus and Poisson ratio ν! The largest relative error

$$\frac{\sqrt{(\sigma_e(E,\nu)-\sigma_{\text{ex},e}):(\sigma_e(E,\nu)-\sigma_{\text{ex},e})}}{\sqrt{\sigma_{\text{ex},e}:\sigma_{\text{ex},e}}}$$

between calibrated model output and experimental data is $< 1.7 \cdot 10^{-8}$.

This means we have a good state estimation, i.e., the model output matches the provided data! In fact, the data was not experimental at all, but was generated with $E = 123$ and $\nu = 0.4$, which the optimizer identified quickly.

5 Why QUEENS?

- Hide complexity in handling complex simulation codes:
  - `Driver`: make 4C callable via a Python function call
  - `Scheduler`: schedule and handle simulations in parallel
  - `Model`: hide the scheduler and driver logic for the QUEENS user
- Nested parallelism
  - Multiple processes per 4C run
  - Multiple parallel 4C runs at the same time
- Modularity:

  - Switched from grid evaluation to optimization simply by changing the iterator
  - 4C simulation model did not change!

5.1 Let's play around

Let your creativity flow and try out stuff.

5.1.1 Inspirations

- Change the parameters ranges
- Look at the 4C input
- Look at the driver outputs
- Why did the simulations fail? (hint: two different reasons)
- Plot the displacement field