

**EXCALIBUR
10**



SiMLInt

**Simulation and Machine
Learning Integration**

Bout++ User Meeting

Jan 9.-11. 2023

Moritz Linkmann

**School of Mathematics and Maxwell Institute for
Mathematical Sciences**

The University of Edinburgh



**UK Research
and Innovation**



**UK Atomic
Energy
Authority**

SiMLInt

People

- Amy Krause (EPCC)
- Jacob Page (School of Mathematics)
- Anna Roubíčková (EPCC)
- Johnny Hay (EPCC)
- Elena Breitmoser (EPCC)



SiMLInt

ExCALIBUR cross-cutting project

ExCALIBUR programme

UK research programme that aims to deliver the next generation of high-performance simulation software for the highest-priority fields in UK research;
Funded by UKRI/EPSRC

Cross-cutting theme

A coordinated approach addressing a known technology or infrastructure issue, which, if resolved, will lead to significant progress across a range of exascale software development challenges

Developed by EPCC and the School of Mathematics at the University of Edinburgh, UK, SiMLInt provides infrastructure that allows embedding Machine Learning capability to large-scale numerical simulations.

More information

<https://excalibur.ac.uk/>
<https://excalibur.ac.uk/projects/simlint/>

Contact us

simlint@mlist.is.ed.ac.uk

Motivation

$$\partial_t u_i + \partial_j \left(u_i u_j + p \delta_{ij} - \frac{1}{Re} S_{ij} \right) = f_i$$

fully resolved simulation → ground truth

Aim → simulation on coarser grid

$$\partial_t \bar{u}_i^l + \partial_j \left(\bar{u}_i^l \bar{u}_j^l + \bar{p}^l \delta_{ij} - \frac{1}{Re} \bar{S}_{ij}^l - \tau_{ij}^l \right) = \bar{f}_i^l$$

$$\tau_{ij}^l = \overline{u_i u_j}^l - \bar{u}_i^l \bar{u}_j^l$$

subgrid-scale stress tensor

$$\tau_{ij}^{\text{model}} = (\ell C_S)^2 \sqrt{\bar{S}_{ij}^l \bar{S}_{ij}^l \bar{S}_{ij}^l}$$

Smagorinsky parametrisation

$$\tau_{ij}^{\text{model}} = f(a, b, c, \dots) \bar{S}_{ij}^l$$

multiple parameters → ML

$$\tau_{ij}^{\text{model}} = X_{ij}(a, b, c, \dots)$$

learn subgrid stresses

$$\tau_{ij}^{\text{error}} = X_{ij}(\bar{\mathbf{u}}^l, \mathbf{u})$$

learned corrections



Motivation

$$\partial_t \bar{u}_i^l + \partial_j \left(\bar{u}_i^l \bar{u}_j^l + \bar{p}^l \delta_{ij} - \frac{1}{Re} \bar{S}_{ij}^l - \tau_{ij}^{\text{model/error}} \right) = \bar{f}_i^l$$

$$\tau_{ij}^{\text{model}} = (\ell C_S)^2 \sqrt{\bar{S}_{ij}^l \bar{S}_{ij}^l \bar{S}_{ij}^l}$$

Smagorinsky parametrisation

$$\tau_{ij}^{\text{model}} = f(a, b, c, \dots) \bar{S}_{ij}^l$$

multiple parameters \rightarrow ML

$$\tau_{ij}^{\text{model}} = X_{ij}(a, b, c, \dots)$$

learn subgrid stresses

$$\tau_{ij}^{\text{error}} = X_{ij}(\bar{\mathbf{u}}^l, \mathbf{u})$$

learned corrections

Motivation

$$\partial_t \bar{u}_i^l + \partial_j \left(\bar{u}_i^l \bar{u}_j^l + \bar{p}^l \delta_{ij} - \frac{1}{Re} \bar{S}_{ij}^l - \tau_{ij}^{\text{model/error}} \right) = \bar{f}_i^l$$

$$\tau_{ij}^{\text{model}} = (\ell C_S)^2 \sqrt{\bar{S}_{ij}^l \bar{S}_{ij}^l \bar{S}_{ij}^l}$$

Smagorinsky parametrisation

$$\tau_{ij}^{\text{model}} = f(a, b, c, \dots) \bar{S}_{ij}^l$$

multiple parameters \rightarrow ML

$$\tau_{ij}^{\text{model}} = X_{ij}(a, b, c, \dots)$$

learn subgrid stresses

$$\tau_{ij}^{\text{error}} = X_{ij}(\bar{\mathbf{u}}^l, \mathbf{u})$$

learned corrections

Motivation

$$\partial_t \bar{u}_i^\ell + \partial_j \left(\bar{u}_i^\ell \bar{u}_j^\ell + \bar{p}^\ell \delta_{ij} - \frac{1}{Re} \bar{S}_{ij}^\ell - \tau_{ij}^{\text{model/error}} \right) = \bar{f}_i^\ell$$

$$\tau_{ij}^{\text{model}} = (\ell C_S)^2 \sqrt{\bar{S}_{ij}^\ell \bar{S}_{ij}^\ell \bar{S}_{ij}^\ell}$$

Smagorinsky parametrisation

$$\tau_{ij}^{\text{model}} = f(a, b, c, \dots) \bar{S}_{ij}^\ell$$

multiple parameters → ML

$$\tau_{ij}^{\text{model}} = X_{ij}(a, b, c, \dots)$$

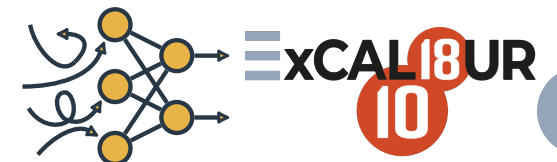
learn subgrid stresses

$$\tau_{ij}^{\text{error}} = X_{ij}(\bar{\mathbf{u}}^\ell, \mathbf{u})$$

learned corrections

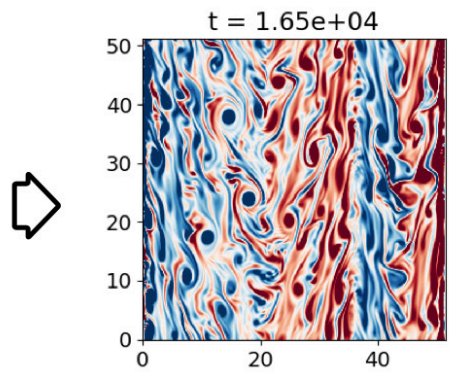
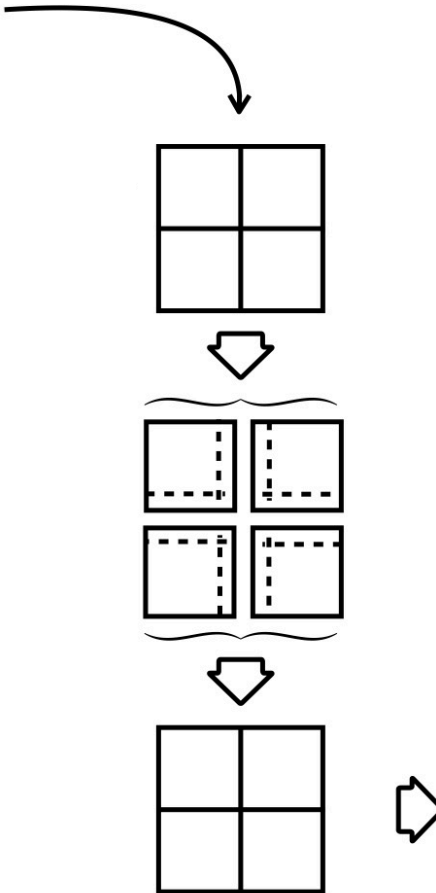
→ provide infrastructure for any of these data-driven approaches

→ couple BOUT++ with ML-model(s)



Concept and Aims

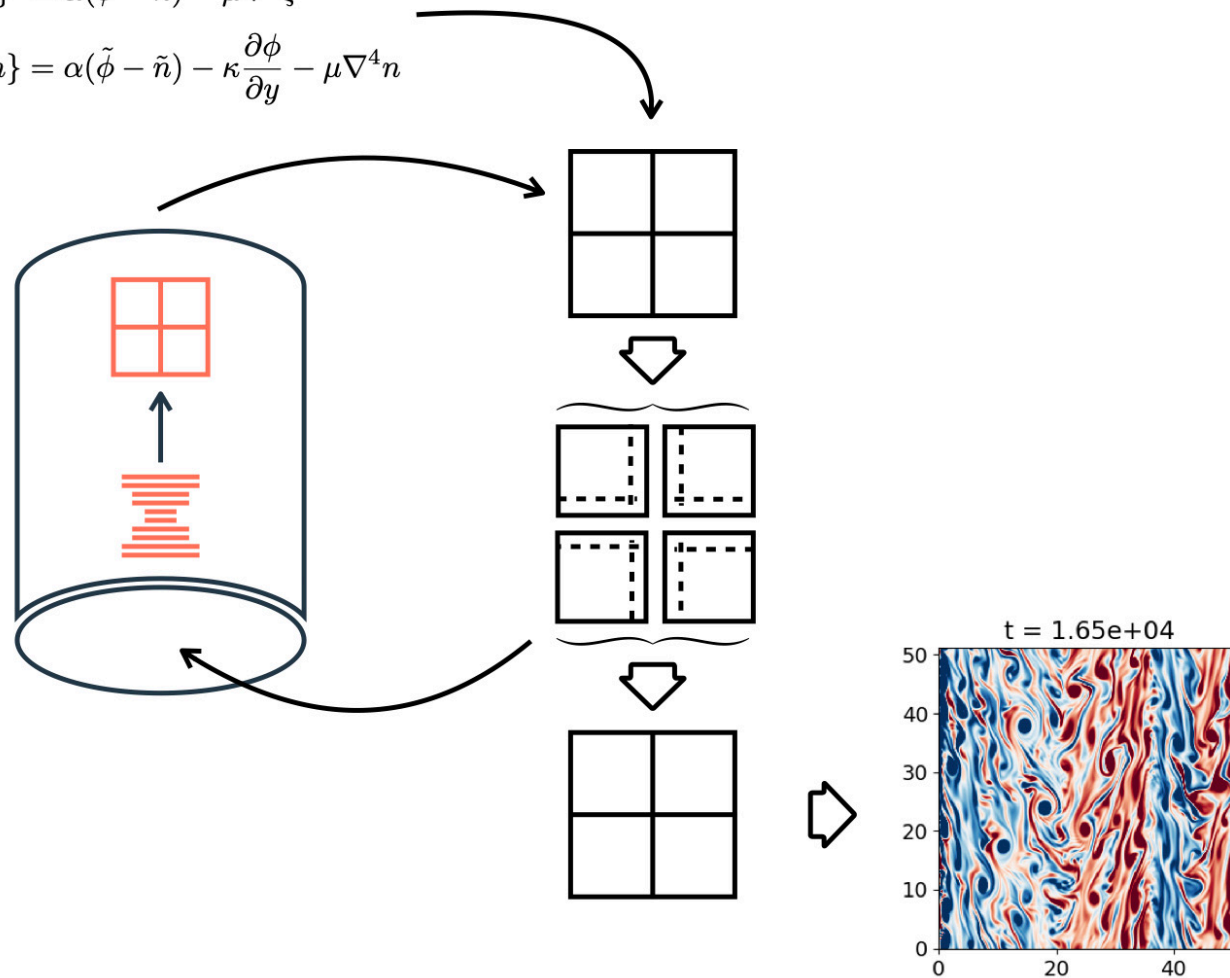
$$\frac{\partial \zeta}{\partial t} + \{\phi, \zeta\} = \alpha(\tilde{\phi} - \tilde{n}) - \mu \nabla^4 \zeta$$
$$\frac{\partial n}{\partial t} + \{\phi, n\} = \alpha(\tilde{\phi} - \tilde{n}) - \kappa \frac{\partial \phi}{\partial y} - \mu \nabla^4 n$$



Concept and Aims

$$\frac{\partial \zeta}{\partial t} + \{\phi, \zeta\} = \alpha(\tilde{\phi} - \tilde{n}) - \mu \nabla^4 \zeta$$

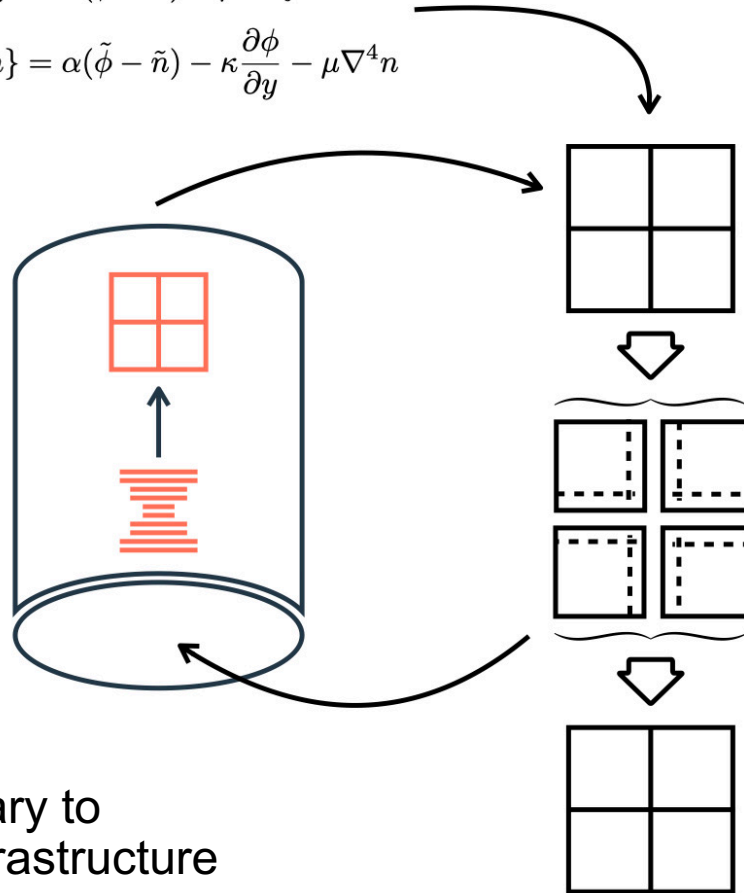
$$\frac{\partial n}{\partial t} + \{\phi, n\} = \alpha(\tilde{\phi} - \tilde{n}) - \kappa \frac{\partial \phi}{\partial y} - \mu \nabla^4 n$$



Concept and Aims

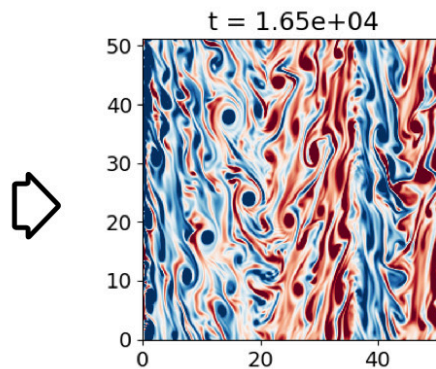
$$\frac{\partial \zeta}{\partial t} + \{\phi, \zeta\} = \alpha(\tilde{\phi} - \tilde{n}) - \mu \nabla^4 \zeta$$

$$\frac{\partial n}{\partial t} + \{\phi, n\} = \alpha(\tilde{\phi} - \tilde{n}) - \kappa \frac{\partial \phi}{\partial y} - \mu \nabla^4 n$$



Objective 1:

Develop a library to provide the infrastructure



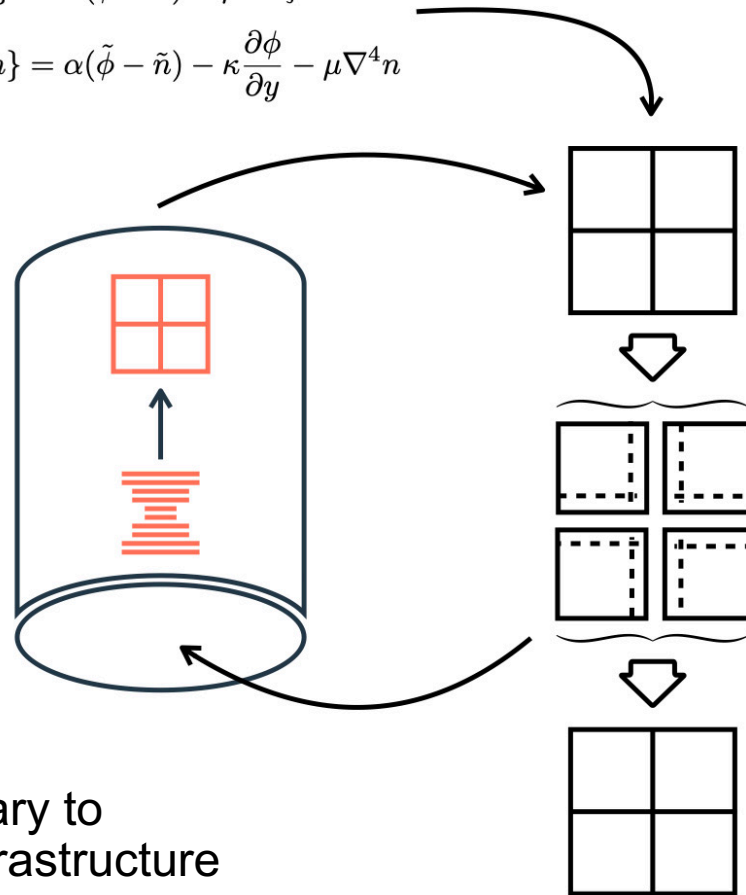
Concept and Aims

$$\frac{\partial \zeta}{\partial t} + \{\phi, \zeta\} = \alpha(\tilde{\phi} - \tilde{n}) - \mu \nabla^4 \zeta$$

$$\frac{\partial n}{\partial t} + \{\phi, n\} = \alpha(\tilde{\phi} - \tilde{n}) - \kappa \frac{\partial \phi}{\partial y} - \mu \nabla^4 n$$

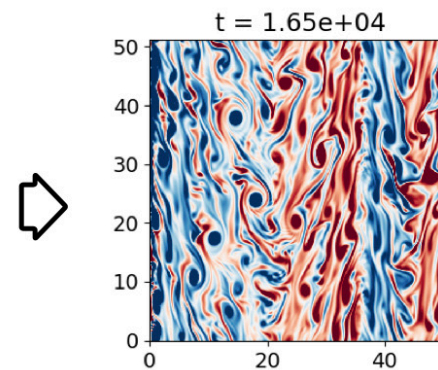
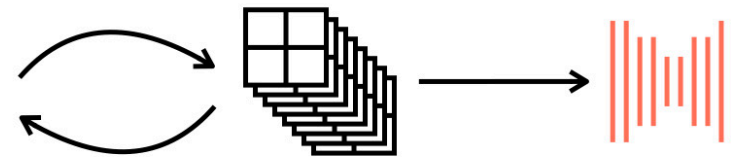
Objective 1:

Develop a library to provide the infrastructure



Objective 2:

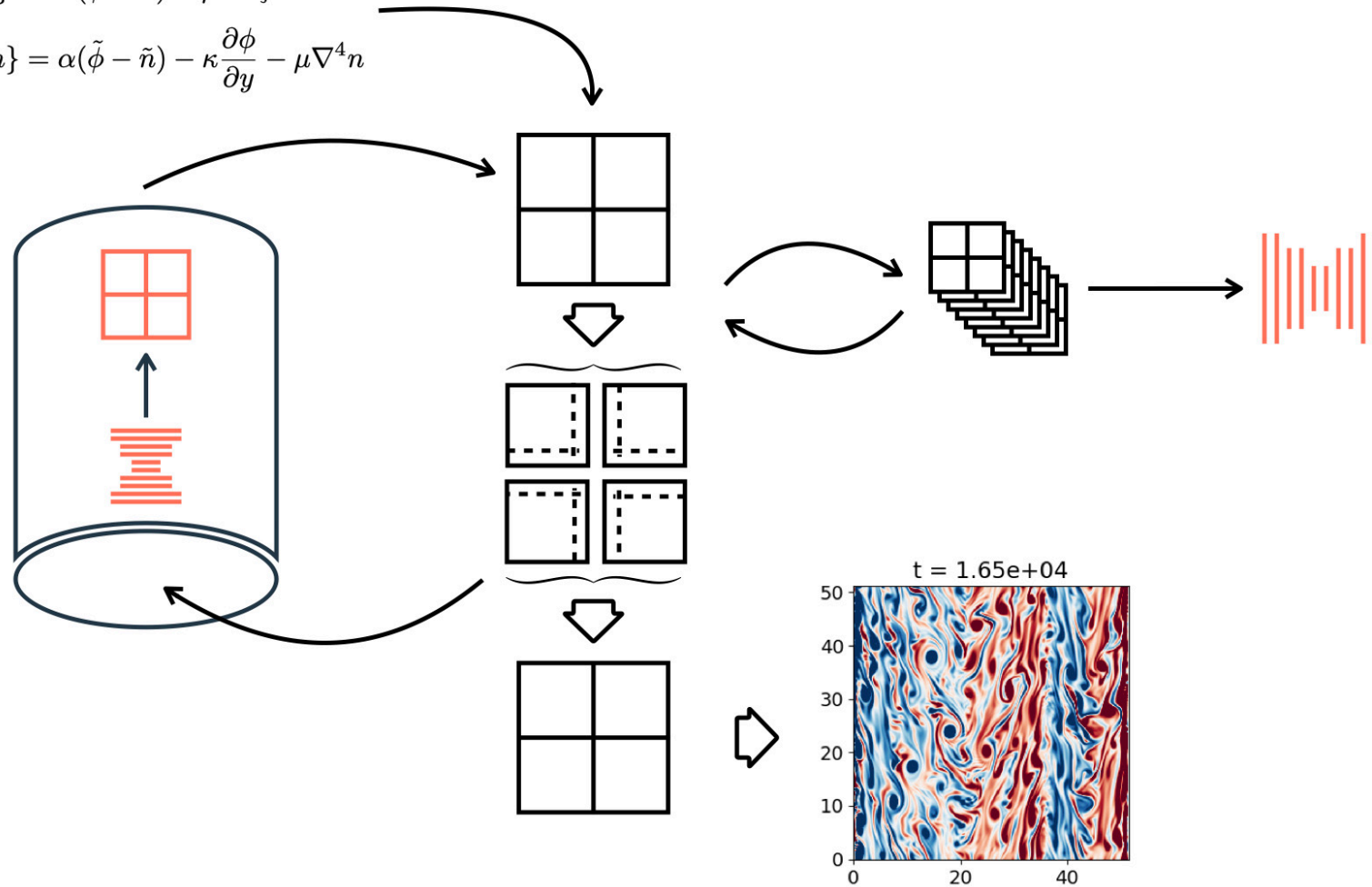
Support generation of training data, building problem-specific CNN architectures and fitting the ML model



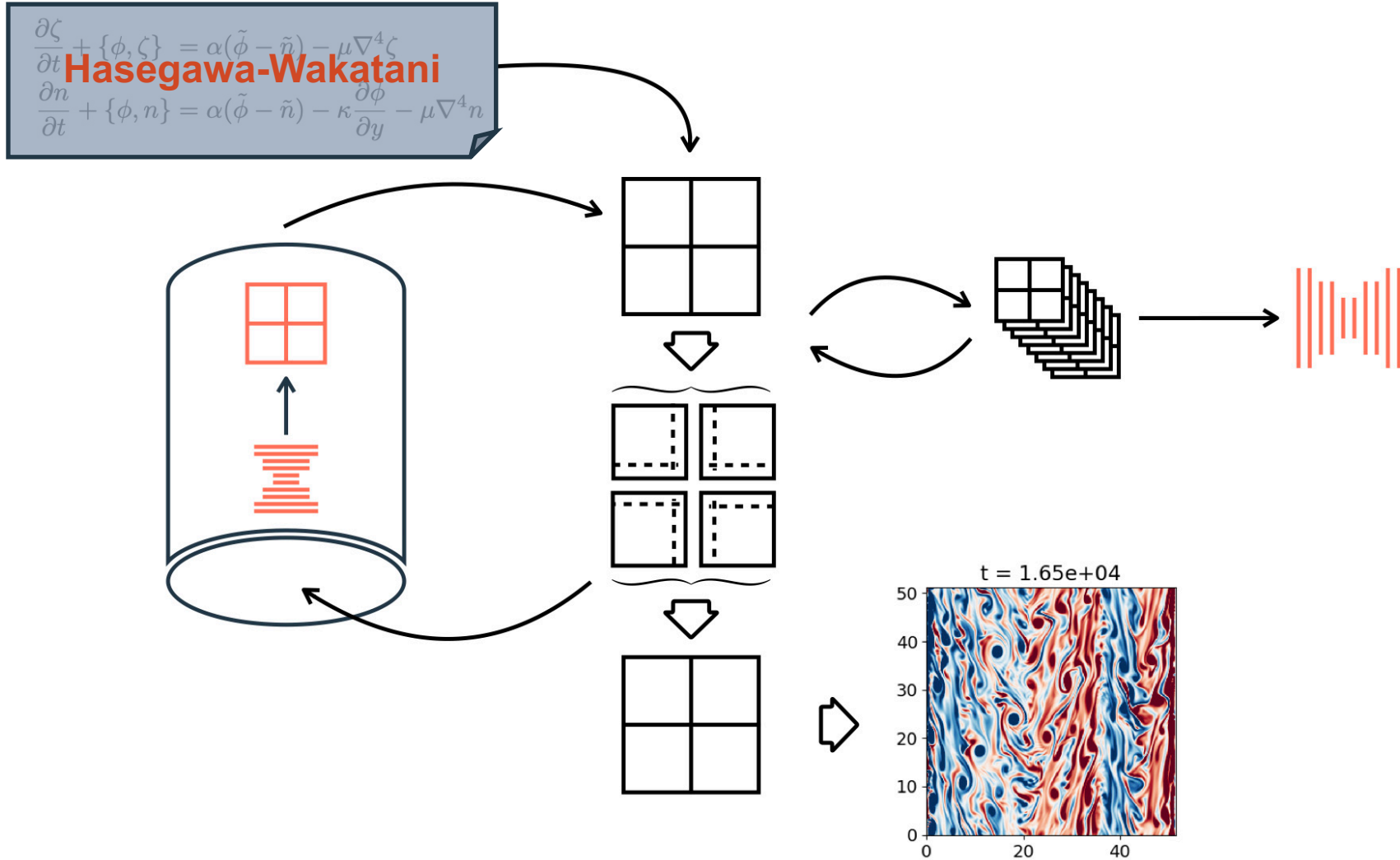
Identified tools/technologies

$$\frac{\partial \zeta}{\partial t} + \{\phi, \zeta\} = \alpha(\tilde{\phi} - \tilde{n}) - \mu \nabla^4 \zeta$$

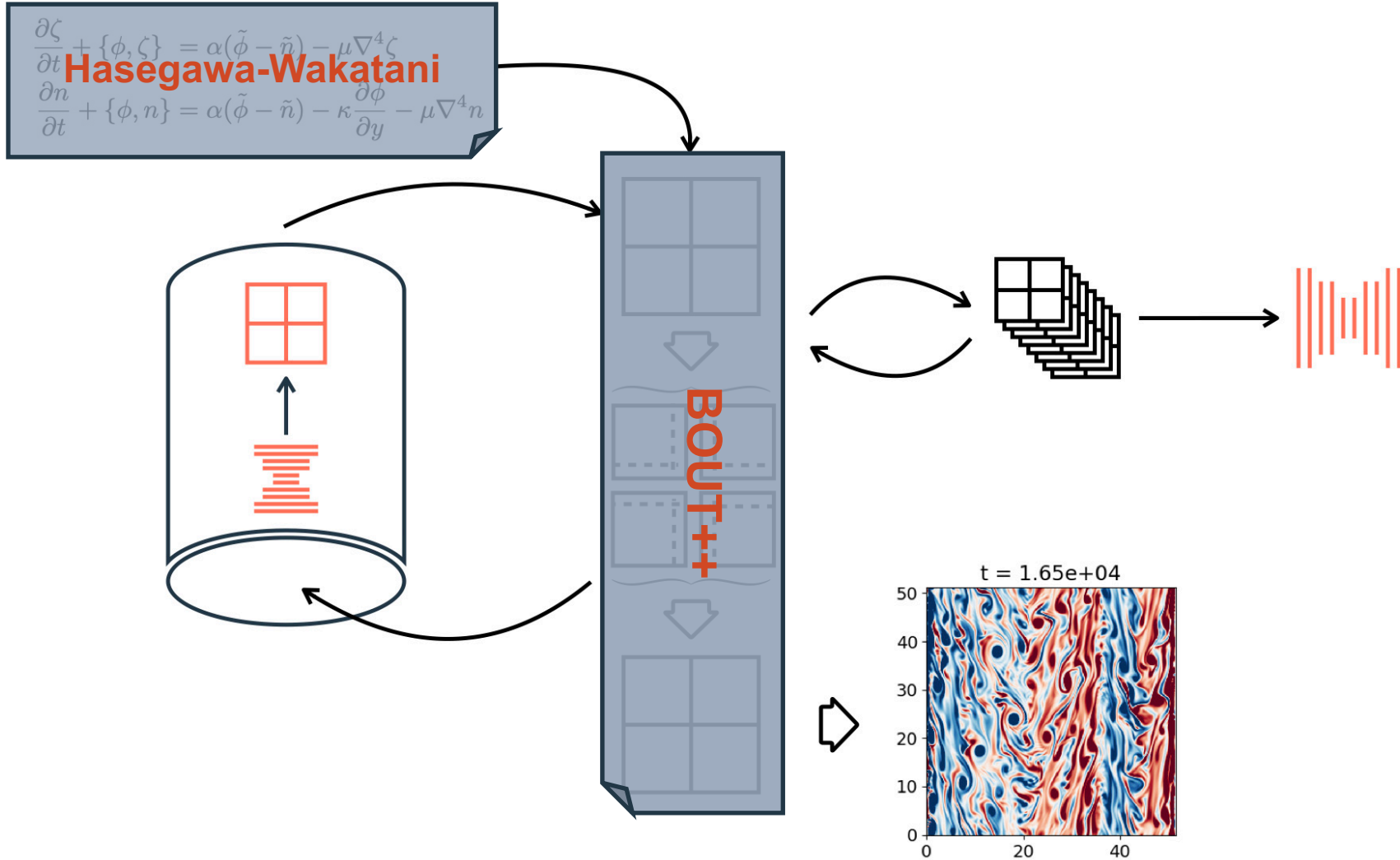
$$\frac{\partial n}{\partial t} + \{\phi, n\} = \alpha(\tilde{\phi} - \tilde{n}) - \kappa \frac{\partial \phi}{\partial y} - \mu \nabla^4 n$$



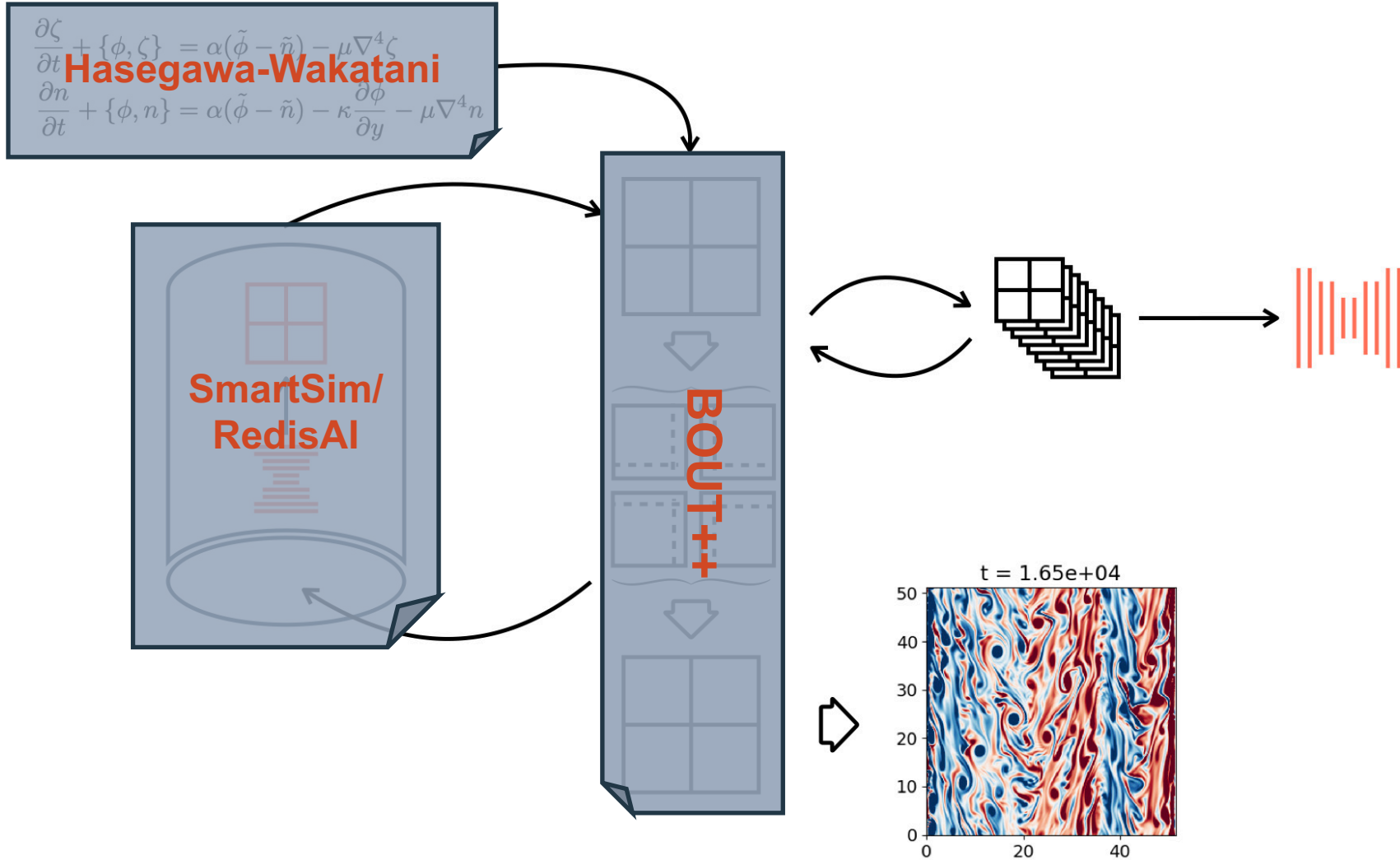
Identified tools/technologies



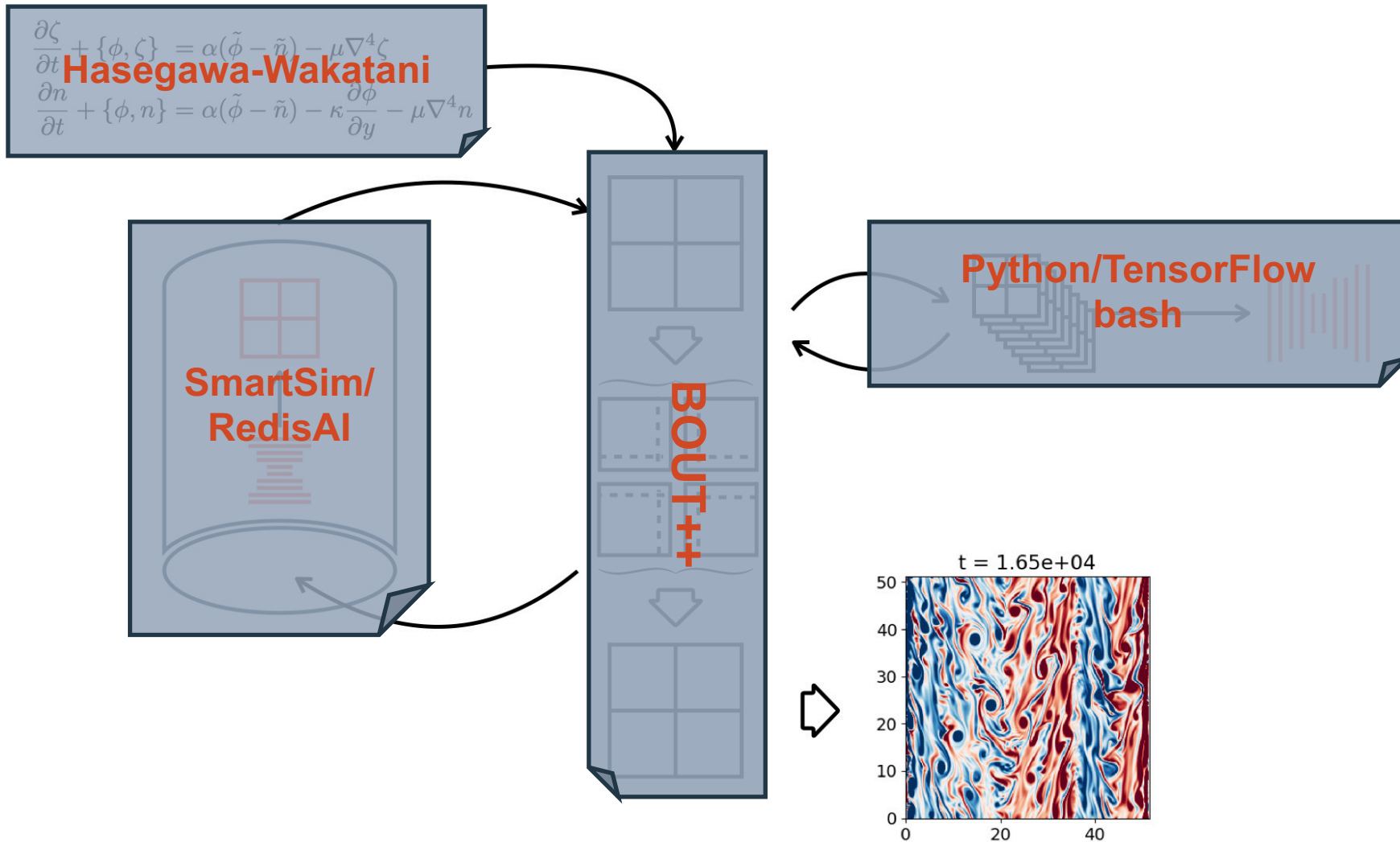
Identified tools/technologies



Identified tools/technologies



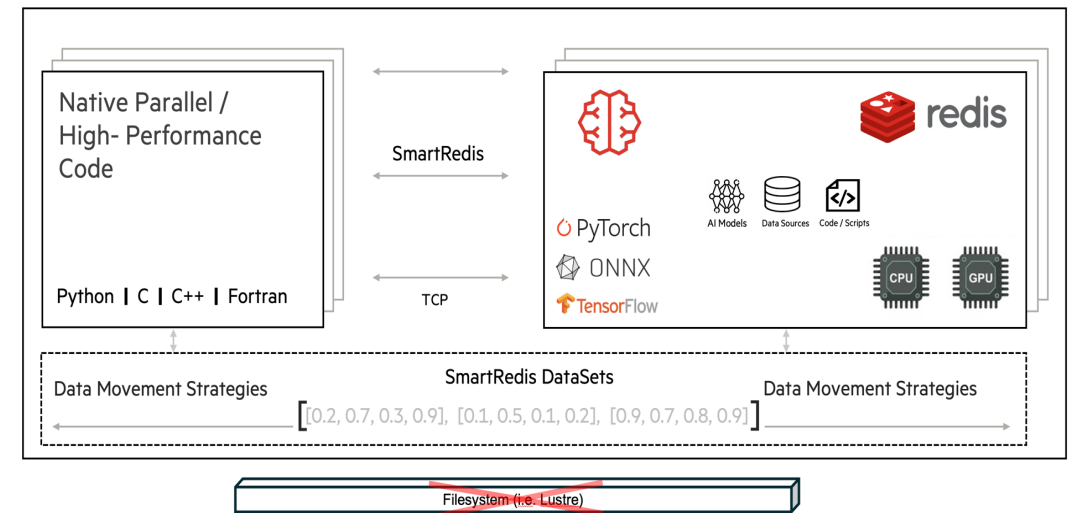
Identified tools/technologies



SmartSim

github.com/CrayLabs/SmartSim

- A library to facilitate data exchange between HPC simulations and Machine Learning workflows
- Two main components:
 1. **SmartSim**: The infrastructure to call Python ML models from Fortran, C and C++ (and Python)
 2. **SmartRedis**: A collection of clients to exchange data (tensors, models) with a distributed, in-memory database (Redis cluster) across multiple compute nodes

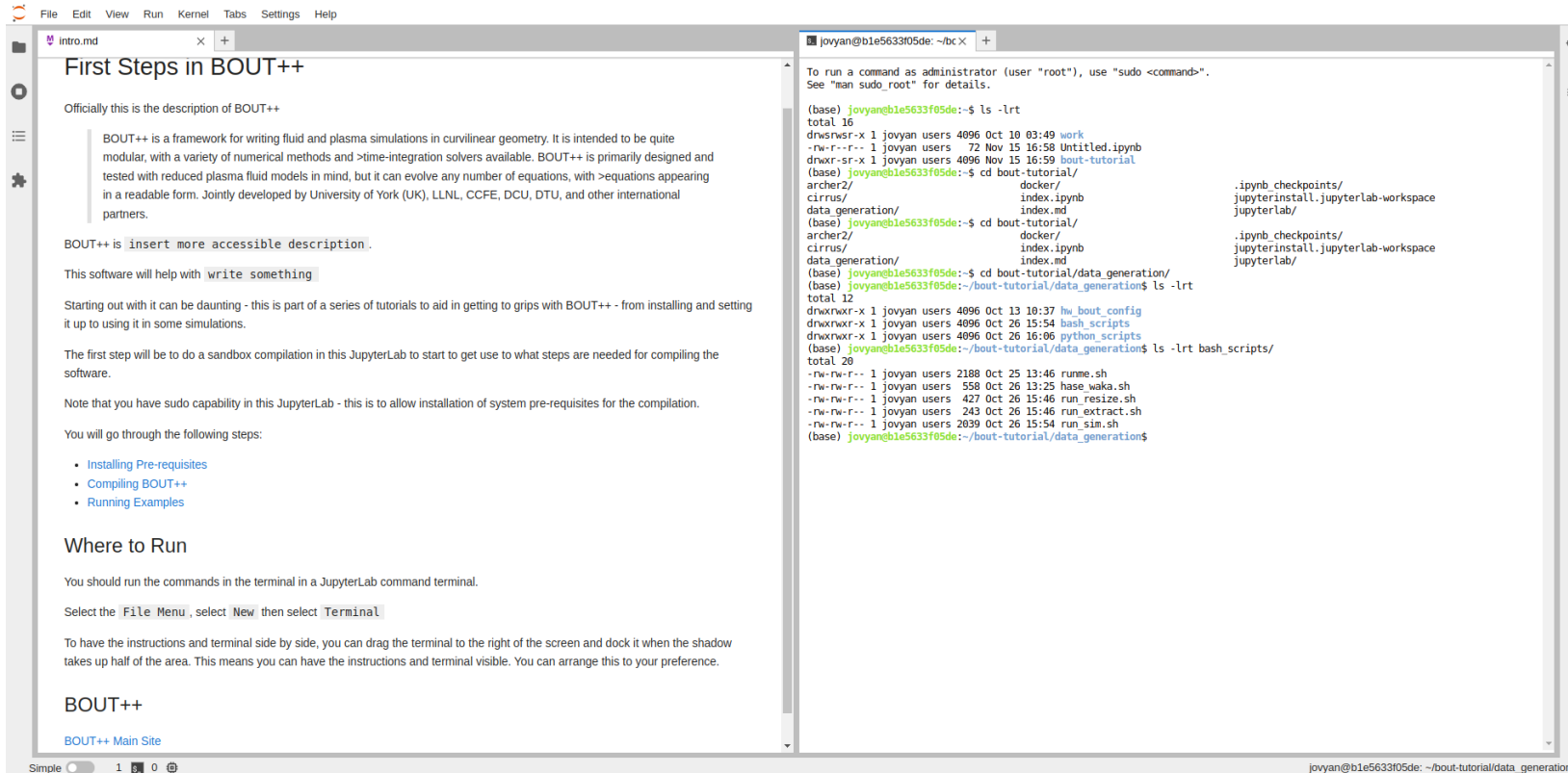


From: <https://www.craylabs.org/docs/overview.html>

- Provides uniform API for pulling/pushing simulation data from C, C++ and Fortran at run-time
- Orchestration ability to coordinate the runs of simulation and machine learning in distributed environment

Our Tutorials

(local and remote/HPC)



The screenshot displays a JupyterLab environment. On the left, a document titled 'intro.md' is open, showing the 'First Steps in BOUT++' tutorial. The text describes BOUT++ as a framework for writing fluid and plasma simulations in curvilinear geometry. On the right, a terminal window shows the user 'jovyan' at a remote host 'b1e5633f05de'. The terminal output includes the command 'ls -lrt' and its output, which lists files and directories such as 'docker/', 'index.ipynb', and 'index.md'. The terminal also shows the execution of 'cd bout-tutorial/' and 'ls -lrt' in the subdirectory, listing files like 'runme.sh', 'hase_waka.sh', 'run_resize.sh', 'run_extract.sh', and 'run_sim.sh'.

1. Training container (docker) with a toy example to run locally, on personal machines

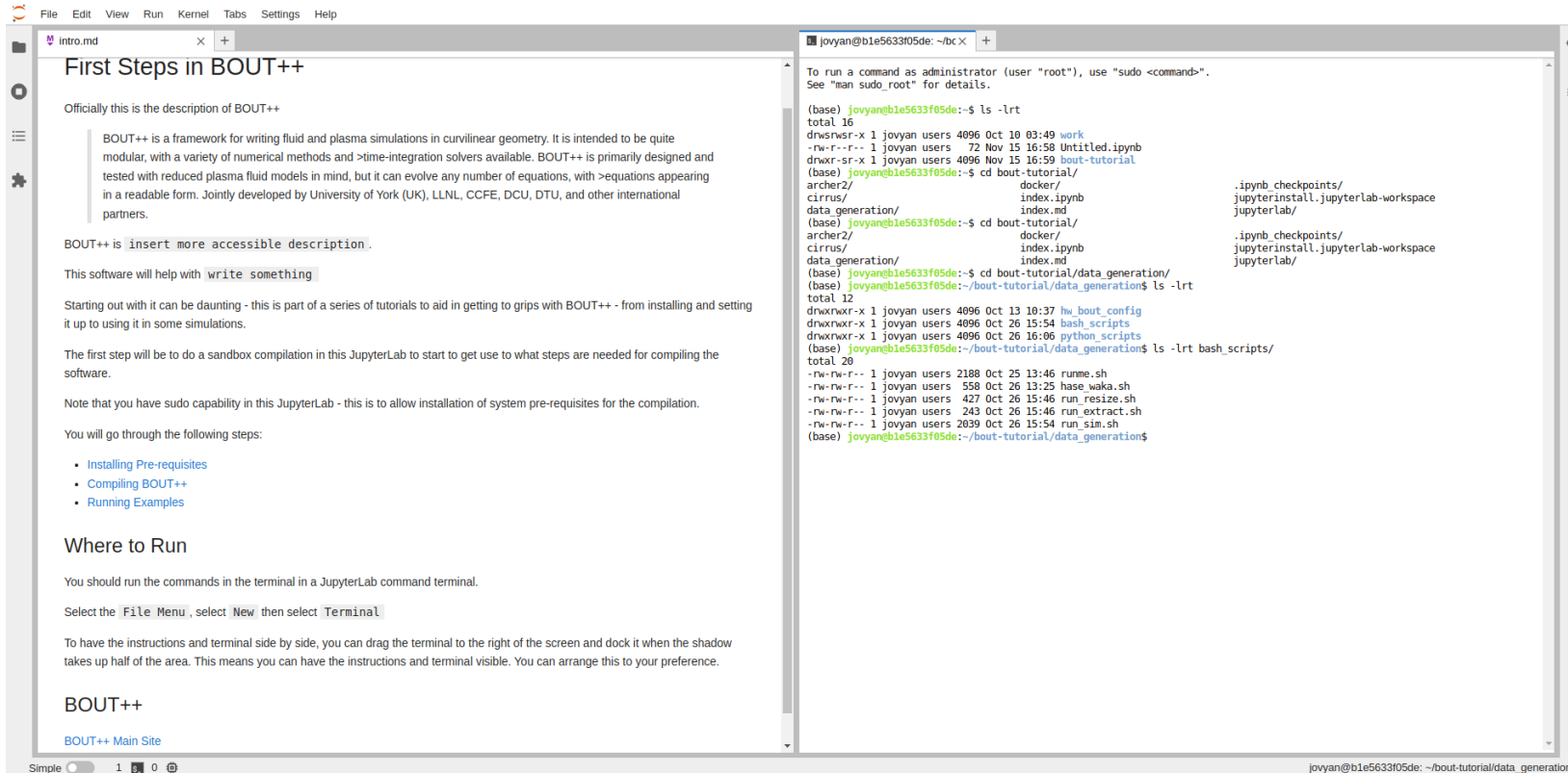
1. Remote visual environment, backed by UK HPC services

1. Prototype and a hands-on workshop in 2023

Our Tutorials

(local and remote/HPC)

Trial available on request



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

(base) jovyan@b1e5633f05de:~$ ls -lrt
total 16
drwxrwxr-x 1 jovyan users 4096 Oct 10 03:49 work
-rw-r--r-- 1 jovyan users 72 Nov 15 16:58 Untitled.ipynb
drwxr-sr-x 1 jovyan users 4096 Nov 15 16:59 bout-tutorial
(base) jovyan@b1e5633f05de:~$ cd bout-tutorial/
archer2/          docker/          .ipynb_checkpoints/
cirrus/          index.ipynb     jupyterinstall.jupyterlab-workspace
data_generation/ index.md         jupyterlab/
(base) jovyan@b1e5633f05de:~$ cd bout-tutorial/
archer2/          docker/          .ipynb_checkpoints/
cirrus/          index.ipynb     jupyterinstall.jupyterlab-workspace
data_generation/ index.md         jupyterlab/
(base) jovyan@b1e5633f05de:~$ cd bout-tutorial/data_generation/
(base) jovyan@b1e5633f05de:~/bout-tutorial/data_generation$ ls -lrt
total 12
drwxrwxr-x 1 jovyan users 4096 Oct 13 10:37 fw_bout_config
drwxrwxr-x 1 jovyan users 4096 Oct 26 15:54 bash_scripts
drwxrwxr-x 1 jovyan users 4096 Oct 26 16:06 python_scripts
(base) jovyan@b1e5633f05de:~/bout-tutorial/data_generation$ ls -lrt bash_scripts/
total 20
-rw-rw-r-- 1 jovyan users 2188 Oct 25 13:46 runme.sh
-rw-rw-r-- 1 jovyan users 558 Oct 26 13:25 hase_waka.sh
-rw-rw-r-- 1 jovyan users 427 Oct 26 15:46 run_resize.sh
-rw-rw-r-- 1 jovyan users 243 Oct 26 15:46 run_extract.sh
-rw-rw-r-- 1 jovyan users 2039 Oct 26 15:54 run_sim.sh
(base) jovyan@b1e5633f05de:~/bout-tutorial/data_generation$
```

1. Training container (docker) with a toy example to run locally, on personal machines

1. Remote visual environment, backed by UK HPC services

1. Prototype and a hands-on workshop in 2023

Collaborations and links

1. BOUT++ users with simulations
1. Similar problems/simulations, other numerical solvers
1. Machine Learning specialists and Data Scientists
 - errors, visualisations, uncertainty, verification/validation of models, ...

simlint@mlist.is.ed.ac.uk

Thanks for listening!

Questions?

References

- D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner and S. Hoyer: **Machine learning–accelerated computational fluid dynamics**. Proceedings of the National Academy of Sciences, 118(21), 2021
- Maulik, R., San, O., Rasheed, A., & Vedula, P. (2019). **Subgrid modelling for two-dimensional turbulence using neural networks**. Journal of Fluid Mechanics, 858, 122-144. doi:10.1017/jfm.2018.770
- P. Watson: **Applying machine learning to improve simulations of a chaotic dynamical system using empirical error correction**. Journal of Advances in Modeling Earth Systems, May 2019.
- M. Wakatani and A. Hasegawa: **A collisional drift wave description of plasma edge turbulence**. The Physics of fluids, 27(3):611–618, 1984
- A. Hasegawa and M. Wakatani. **Self-organization of electrostatic turbulence in a cylindrical plasma**. Phys. Rev. Lett., 59:1581–1584, Oct 1987.
- TensorFlow: [tensorflow.org](https://www.tensorflow.org)
Keras: keras.io
- S. Partee, M. Ellis, A. Rigazzi, S. Bachman, G. Marques, A. Shao, and B. Robbins: **Using machine learning at scale in HPC simulations with smartsim: An application to ocean climate modeling**. CoRR, abs/2104.09355, 2021.
github.com/CrayLabs/SmartSim

