INDUCTIVA

FOAM@IBERIA 2024

Easily Deploy OpenFOAM Applications with Python and Cloud HPC

Rodrigo Valério, Hugo Penedones, Luís Sarmento

www.inductiva.ai

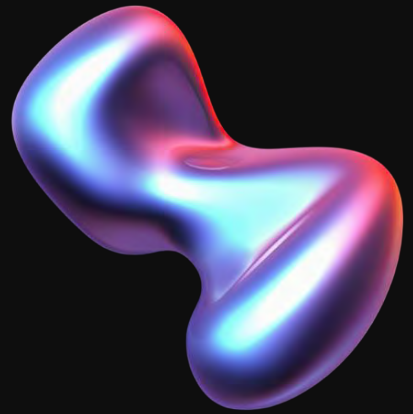
INDUCTIVA

About Inductiva

We are a cloud-based **High-Performance Computing (HPC) platform** designed to make running large-scale simulations easy, fast, and cost-effective.

With our flexible **Python API**, you can run simulations seamlessly—without worrying about complex infrastructure.

With our **streamlined, user-friendly platform**, Inductiva empowers you to focus on innovation and research while cutting costs and speeding up your work.



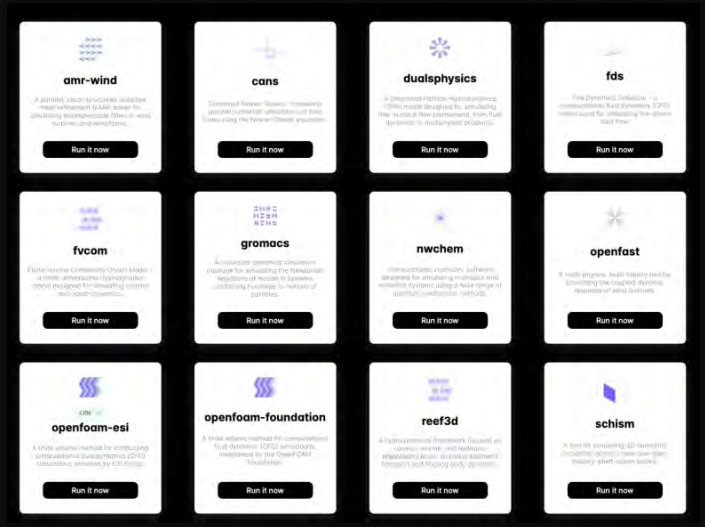
INDUCTIVA

What Makes Inductiva.AI Special?

We are Open Source

Inductiva is built on open-source principles, providing transparency and flexibility.

Our API also includes a variety of **open-source simulators**, such as OpenFOAM, DualSPHysics, and Reef3D, with more being added regularly.



INDUCTIVA.AI/SIMULATORS

INDUCTIVA

What Makes Inductiva.AI Special?

We are Transparent

Inductiva allows you to track your simulations with ease through an intuitive **Web console**.

Gain clearer insights into your simulation and platform usage with updated dashboards, offering a detailed overview of task performance, disk usage, and active machine groups.

CONSOLE.INDUCTIVA.AI

INDUCTIVA

What Makes Inductiva.AI Special?

We are Cost Effective

Inductiva helps you reduce computational costs and scale resources as needed through:

Preemptible machines: Take advantage of cheaper, short-term computing resources.
Flexible execution: Choose to run simulations on the cloud or locally.
Elastic Machine Groups: Automatically scale resources based on demand.
MPI clusters: Distribute workloads efficiently for faster, cost-effective simulations.
Checkpointing: Save your progress and resume simulations without starting over. (soon)

```
import inductiva
machine_group = inductiva.resources.MachineGroup(
    machine_type="c2-standard-16",
    num_machines=2,
    data_disk_gb=100,
    spot=True)
```

INDUCTIVA

What Makes Inductiva.AI Special?

We are Easy to Use

Inductiva is designed with an **API-first approach.**

Build your simulations directly on top of our API, integrating seamlessly with your existing workflows.

No need to manage complex infrastructure—focus on your simulations, not the backend.

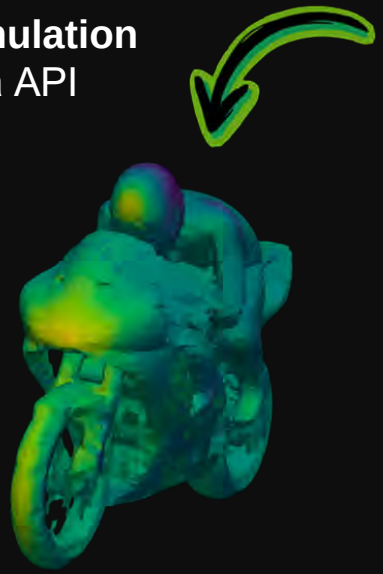
```
# Set the simulation commands
commands = [
 "runApplication surfaceFeatures",
 "runApplication blockMesh",
 "runApplication decomposePar -copyZero",
 "runParallel snappyHexMesh -overwrite",
 "runParallel potentialFoam",
 "runParallel simpleFoam",
 "runApplication reconstructParMesh -constant",
 "runApplication reconstructPar -latestTime"
]
```

INDUCTIVA

Simulations **Made Easy**

An Example: Run a Wind Tunnel Simulation

1. **Submit your simulation** with the Inductiva API
1. Get your **result**



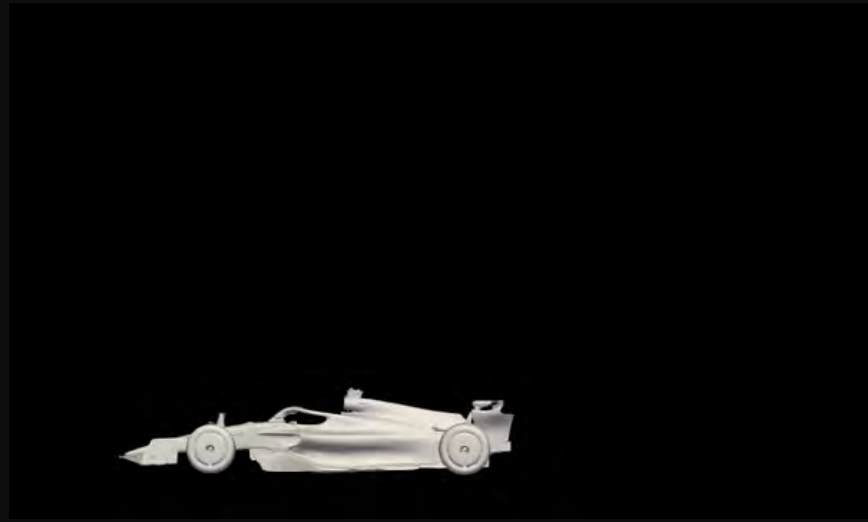
```python
import inductiva

# Instantiate machine group
machine_group = inductiva.resources.MachineGroup('c2-standard-4')
machine_group.start()

# Set simulation input directory
input_dir = inductiva.utils.download_from_url(
 "https://storage.googleapis.com/inductiva-api-demo-files/"
 "openfoam-input-example.zip", unzip=True

# Initialize the Simulator
openfoam = inductiva.simulators.OpenFOAM(distribution="foundation")

# Run simulation with config files in the input directory
task = openfoam.run(input_dir=input_dir,
 commands=commands,
 n_vcpus=4,
 on=machine_group)

task.wait()
task.download_outputs()

machine_group.terminate()
```

INDUCTIVA

From a Simulation Workflow to a **Simulation App**

An Example: Build a Wind Tunnel App

1. Define **OpenFOAM Input Files.**
2. Make relevant **parameters configurable.**
3. Define a **Python interface** with the Inductiva API.



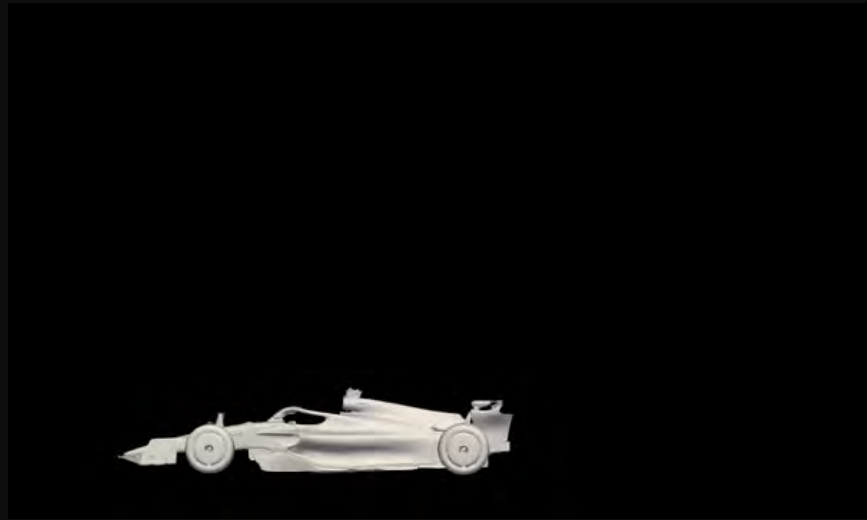
INDUCTIVA

An Example: Build a Wind Tunnel App

Step 1: Define **OpenFOAM Input Files.**

Gather all the required input files for OpenFOAM and place them in a directory.

For the wind tunnel simulation, you'll need the **0**, **constant**, and **system sub-directories**, each containing the appropriate initial and boundary conditions, physical properties, and solver settings.



INDUCTIVA

An Example: Build a Wind Tunnel App

Step 2: Make relevant **parameters configurable.**

Modify the OpenFOAM input files so that key simulation parameters become variables.

These variables can be dynamically filled by the **Inductiva TemplateManager**, allowing for customized simulations.

```
1    dragDir          (1 0 0);
2    CofR             (1 0 0);  // Axle midpoint on ground
3    pitchAxis        (0 1 0);
4    magUInf          {{ wind_speed }};    // Wind-speed
5    lRef             {{ length }};        // Wheelbase length
6    Aref             {{ area }};          // Object projection area
```

Detail view of the system/forceCoeffs configuration file.

INDUCTIVA

An Example: Build a Wind Tunnel App

Step 3: Define a **Python interface** with the Inductiva API.

```python
import inductiva
class WindTunnel:

    def __init__(self, dimensions: tuple[int, int, int] = (20, 10, 8)):
        ...

    def set_object(self,
                   object_path: str,
                   rotate_z_degrees: float = 0,
                   normalize: bool = True,
                   center: bool = True):
        ...

    def simulate(self,
                 wind_speed_ms: float,
                 num_iterations: int,
                 resolution: int,
                 machine_group_name: str = None,
                 inputs_dir: str = "./inductiva_input"):
        ...
```

Creating python package

INDUCTIVA

What Can You Do with Your New **Wind Tunnel App?**

Run a Simulation and Share the App

You can easily share your Python package with your team or collaborators, and use it to submit simulations—all through a **simple Python interface**.

```python
from windtunnel import WindTunnel

wind_tunnel = WindTunnel(dimensions=(20, 10, 8))

wind_tunnel.set_object(object_path="tractor.obj")
task = wind_tunnel.simulate(wind_speed_ms=20, num_iterations=300, resolution=5)
```

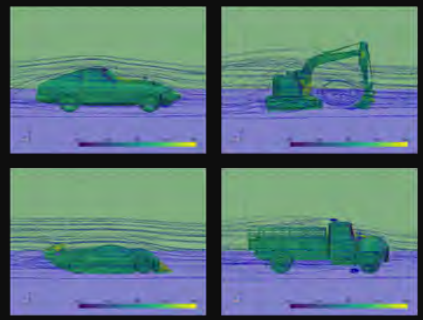
Submitting a Simulation Using Your Python Package

INDUCTIVA

What Can You Do with Your New **Wind Tunnel App?**

Run Multiple Simulations in Parallel

```python
from windtunnel import WindTunnel

wind_tunnel = WindTunnel(dimensions=(20, 10, 8))

for fname in ["car", "excavator", "race_car", "truck"]:
  wind_tunnel.set_object(object_path=f"assets/{fname}.obj")

  for wind_speed_ms in [10, 30, 50]:
    task = wind_tunnel.simulate(
      wind_speed_ms=wind_speed_ms, num_iterations=300, resolution=5
    )
```



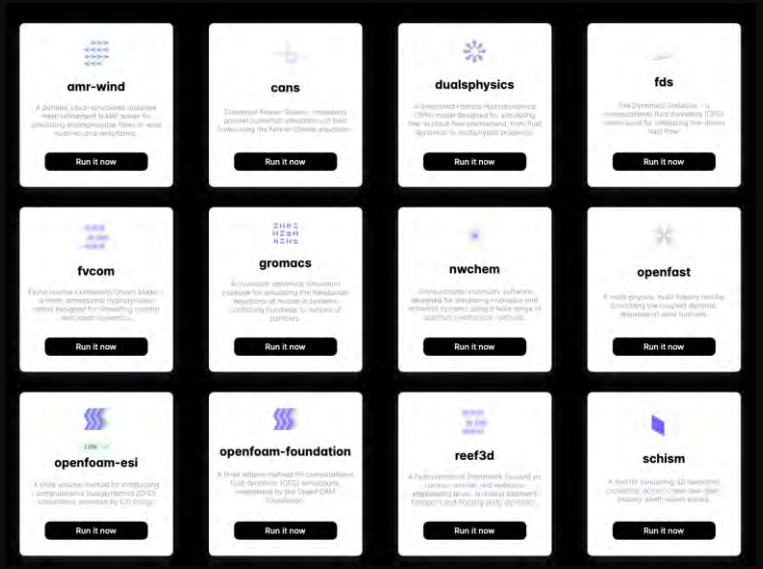
Running Multiple Simulations in Parallel
Using Your Python Package

INDUCTIVA

Use Our API—No Matter the Domain

You can apply the same recipe across different domains, from fluid and coastal dynamics to structural mechanics and even molecular dynamics

- ➔ Choose from a variety of built-in, ready-to-use **open-source simulators**, no setup headaches required
- ➔ Build your own app using **the same simple approach**.



INDUCTIVA

We Generated a 20k Wind Tunnel Dataset

Using the Inductiva API and our wind tunnel package, we generated a dataset of automobile-like objects:

- ➔ The dataset is **open source** with **20,000 samples**.
- ➔ It includes **force coefficients**, **pressure maps**, and **streamlines**.
- ➔ And best of all? It's **free to download!**

INDUCTIVA

Thank You!

Any questions?

Get in Touch

rvalerio@inductiva.ai
–
contact@inductiva.ai

www.inductiva.ai
–
linktr.ee/inductiva_windtunnel

company/inductiva-ai