

Spatial Modeling Algorithms for Reaction-Transport (SMART)

Installation

SMART has been installed and tested on Linux for AMD, ARM, and x86_64 systems, primarily via Ubuntu 20.04 or 22.04. On Windows devices, we recommend using Windows Subsystem for Linux to run the provided docker image (see below). SMART has also been tested on Mac OS using docker. Installation using docker should take less than 30 minutes on a normal desktop computer.

Using Docker (recommended)

1. Install [Docker Desktop](#) for Windows, Mac, or Linux. In Windows, it is recommended to use Docker with the WSL2 backend, requiring you to first [install WSL2](#) and then follow the [Docker instructions for using the WSL2 backend](#).
2. From your command line (WSL, Mac, or Linux), pull the desired Docker image from the Github registry:

```
docker pull ghcr.io/rangamanilabucsd/smart:latest
```

It is possible to pull a specific version by changing the tag, e.g.

```
docker pull ghcr.io/rangamanilabucsd/smart:v2.0.1
```

will use version 2.0.1.

3. In order to start a container you can use the [docker run](#) command. For example the command

```
docker run -v $(pwd):/home/shared -w /home/shared -ti --name smart-tutorial  
ghcr.io/rangamanilabucsd/smart:latest
```

will run the latest version and mount your current working directory within the container at the location /home/shared. \$(pwd) can be replaced with any local directory on your computer. Within the container, you will then be accessing that same content from your local directory at the location /home/shared. The source code of SMART is located at /repo in the Docker container. Note: including the option --rm will automatically delete the Docker container upon closing. I generally do not include this, allowing me to stop and restart the container at any point using Docker Desktop, without needing to create a new container each time (see image

below)

Containers [Give feedback](#)

Container CPU usage **76.61% / 2400%** (24 CPUs available) Container memory usage **8.66GB / 15.12GB** [Show charts](#)

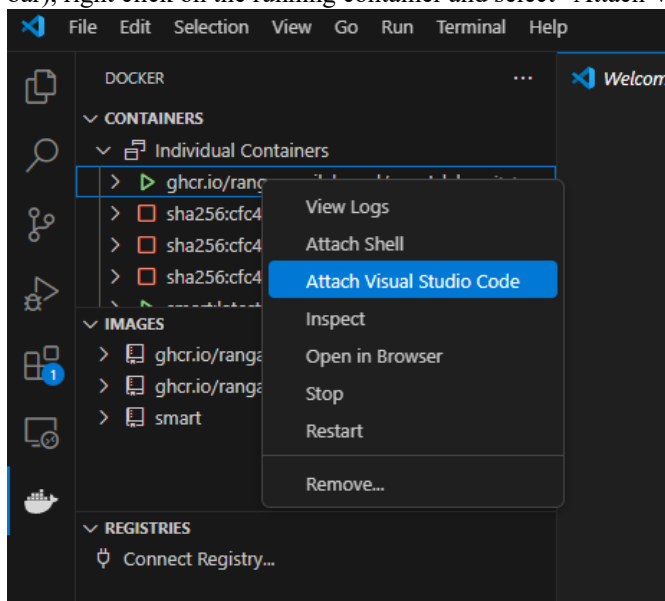
Search ☐ Only running

	Name	Container ID	Image	Port(s)	Actions
<input type="checkbox"/>	smart-test	5ea7bf744576	rangamanil -		▶ ⋮ 🗑️
<input type="checkbox"/>	smart-github	2c0d2ed77192	rangamanil -		▶ ⋮ 🗑️
<input type="checkbox"/>	smart	5447566dee73	rangamanil -		▶ ⋮ 🗑️
<input type="checkbox"/>	smart-with-scratch	6240bfa0d6c	smart:lates 8080:8888		▶ ⋮ 🗑️

Showing 4 items

RAM 13.02 GB CPU 5.19% Disk --- GB avail. of --- GB BETA [Terminal](#) [🔔](#)

4. (recommended workflow in VS Code) SMART can be run within the Docker container from the command line, but I find it most convenient to run in VS Code. To do this, install VS Code for [Windows with WSL](#), or [Mac or Linux](#). In Windows, open VS Code from within WSL (code . from within Ubuntu or other distribution), or simply open the program normally within Mac or Linux systems. Install the Docker and Dev Container extensions. From within the Docker extension (look for the Docker icon on the left bar), right click on the running container and select “Attach Visual Studio Code”.



This will open a new window attached to the Docker container, where you can access files in the container and run associated code. Within this new window, you should minimally install the Python and Jupyter extensions.

Alternate Docker option - Jupyter Lab

SMART can be run with Jupyter Lab in your web browser if preferred.

1. Install Docker as outlined above.
2. To run the example notebooks, one can use `ghcr.io/rangamanilabucsd/smart-lab`

```
docker run -ti -p 8888:8888 --rm ghcr.io/rangamanilabucsd/smart-lab
```

to run interactively with Jupyter lab in your browser.

Note on converting notebooks to Python files

In the `smart` and `smart-lab` images, these files exist under `/repo/examples/**/example*.py`.

If you clone the git repository or make changes to the notebooks that should be reflected in the python files, you can run

```
python3 examples/convert_notebooks_to_python.py
```

to convert all notebooks to python files. **NOTE** this command overwrites existing files.

Using pip

`fenics-smart` is also available on [pypi](#) and can be installed with

```
python3 -m pip install fenics-smart
```

However this requires FEniCS version 2019.2.0 or later to already be installed. Currently, FEniCS version 2019.2.0 needs to be built [from source](#) or use some of the [pre-built docker images](#)

Example usage

The SMART repository contains a number of examples in the `examples` directory which also run as continuous integration tests (see “Automated Tests” below):

- [Example 1](#): Formation of Turing patterns in 2D reaction-diffusion (rectangular domain)
- [Example 2](#): Simple cell signaling model in 2D (ellipse)
- [Example 2 - 3D](#): Simple cell signaling model in 3D (realistic spine geometry)
- [Example 3](#): Model of protein phosphorylation and diffusion in 3D (sphere)
- [Example 4](#): Model of second messenger reaction-diffusion in 3D (ellipsoid-in-an-ellipsoid)
- [Example 5](#): Simple cell signaling model in 3D (cube-in-a-cube)
- [Example 6](#): Model of calcium dynamics in a neuron (sphere-in-a-sphere)