

Université Toulouse Paul Sabatier - KUPR9AC5

2024-11-04

Guilhem Saurel

Available at

**[https://nim65s.github.io/teach/
2024-2025/3A_SRI/a-tp-1.pdf](https://nim65s.github.io/teach/2024-2025/3A_SRI/a-tp-1.pdf)**

Under License



<https://creativecommons.org/licenses/by-sa/4.0/>

Source

<https://github.com/nim65s/teach> :
2024-2025/3A_SRI/a-tp-1.md

Contact

Matrix: @gsaurel:laas.fr
Mail: gsaurel@laas.fr

- curl: **curl --version**
- Git: **git --version**
- uv: **curl -LsSf **
<https://astral.sh/uv/install.sh> | sh

```
$ uv init tp-multithreading  
$ cd tp-multithreading
```

- Sur <https://github.com>, créer un compte puis un dépôt:
 - sans fichier README.md
 - sans license
 - sans .gitignore

- Sur <https://github.com>, créer un compte puis un dépôt:
 - sans fichier README.md
 - sans license
 - sans .gitignore

```
$ ssh-keygen -t ed25519
```

```
$ cat ~/.ssh/id_ed25519.pub
```

<https://github.com/settings/ssh/new>

```
$ git config --global user.name "John Doe"
```

```
$ git config --global user.email johndoe@example.fr
```

```
$ git remote add origin \  
    git@github.com:votre-nom/votre-dépôt.git  
$ git add .  
$ git commit -m "start uv project"  
$ git branch -M main  
$ git push -u origin main
```



```
$ curl https://gitlab.laas.fr/gsaurel/teach\  
    /-/raw/main/.pre-commit-config.yaml \  
    -o .pre-commit-config.yaml  
$ uv add --dev pre-commit  
$ uv run pre-commit install  
$ uv run pre-commit run -a
```

```
$ curl https://gitlab.laas.fr/gsaurel/teach\  
    /-/raw/main/.pre-commit-config.yaml \  
    -o .pre-commit-config.yaml  
  
$ uv add --dev pre-commit  
$ uv run pre-commit install  
$ uv run pre-commit run -a  
  
$ uv run pre-commit run -a  
$ git add .  
$ git commit -m "setup tooling"  
$ git push
```

Depuis l'UI github:

- Ajoutez un fichier appelé **LICENSE** et choisissez un template

puis **git pull**

Ajoutez un fichier **.github/workflows/ci.yml**, qui vérifie:

- les outils: **uv run pre-commit run -a**
- les tests: **uv run python -m unittest**

et qui s'exécute **on: [push]**

ref. <https://docs.astral.sh/uv/guides/integration/github>

Ajoutez un fichier **.github/workflows/ci.yml**, qui vérifie:

- les outils: **uv run pre-commit run -a**
- les tests: **uv run python -m unittest**

et qui s'exécute **on: [push]**

ref. <https://docs.astral.sh/uv/guides/integration/github>

En python 3.10, 3.11, 3.12 et 3.13

```
$ uv add numpy
$ curl https://gitlab.laas.fr/gsaurel/teach\
    /-/raw/main/src/task.py \
    -o task.py
```

Ce fichier contient une classe **Task** qui permet de résoudre $Ax = B$ en mesurant le temps nécessaire à cette résolution.

Ajoutez un fichier **test_task.py** en suivant l'exemple basique **unittest**.

Vérifiez avec **numpy.testing.assert_allclose** que Ax est égal à B

Not equal to tolerance $\text{rtol}=1\text{e-}07$, $\text{atol}=0$

Max absolute difference among violations: 0.99605365

Max relative difference among violations: 1.

ACTUAL: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,

DESIRED: array([4.986837e-01, 5.483427e-01, 4.227059e-01, 4.451978e-01, 5.737015e-03, 1.497679e-01, 8.135694e-01, 6.948812e-01, 6.933447e-01, 2.165097e-01, 8.151978e-01])

Ran 1 test in 0.020s


```
$ git add .  
$ git commit -m "task & test"  
$ git push
```