

Installation de JAX-FEM, JAX-CPFEM et Neper

1 Introduction

Ce document décrit les étapes d'installation de JAX-FEM, JAX-CPFEM, et Neper sur une machine virtuelle Ubuntu, permettant de configurer un environnement de simulation CPFEM fonctionnel.

2 Installation de Conda et Git

2.1 Installation de Git

Git est nécessaire pour cloner les dépôts de JAX-FEM et JAX-CPFEM.

1. Installez Git :

```
1 sudo apt-get update
2 sudo apt-get install git
```

2. Vérifiez l'installation de Git :

```
1 git --version
```

2.2 Installation de Conda

Miniconda est une version allégée de Conda, un gestionnaire d'environnements et de paquets pour Python.

1. Téléchargez et installez Miniconda :

```
1 wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
2 bash Miniconda3-latest-Linux-x86_64.sh
```

2. Vérifiez l'installation de Conda :

```
1 conda --version
```

3. Mettez à jour Conda :

```
1 conda update conda
```

3 Installation de JAX-FEM et JAX-CPFEM

1. Clonez les dépôts de JAX-FEM et JAX-CPFEM depuis GitHub :

```
1 git clone https://github.com/deepmodeling/jax-fem.git
2 git clone https://github.com/SuperkakaSCU/JAX-CPFEM.git
```

2. Naviguez dans le dossier JAX-FEM : Il faut se situer dans le dossier JAX-FEM pour créer l'environnement Conda :

```
1 cd JAX-FEM
```

3. Créez un environnement Conda spécifique pour JAX-FEM :

```
1 conda env create -f environment.yml
```

4. Activez l'environnement Conda :

```
1 conda activate jax-fem-env
```

5. Installation de JAX:

Pour installer JAX, suivez les instructions disponibles sur le site officiel de JAX.

4 Dépendances nécessaires à l'installation de JAX-CPFEM

Certaines bibliothèques sont essentielles au bon fonctionnement de JAX-CPFEM. Si elles ne sont pas installées, certaines fonctionnalités du projet risquent de ne pas fonctionner correctement. Voici la liste des bibliothèques nécessaires :

- **fenics-basix** : utilisé pour l'intégration avec les outils FEniCS.

```
1 pip install fenics-basix==0.10.0.dev0
```

- `petsc4py` : nécessaire pour le support de la bibliothèque PETSc.

```
1 pip install petsc4py==3.17.1
```

- `scikit-fem` : pour les calculs d'éléments finis.

```
1 pip install scikit-fem==10.0.2
```

- `pygmsh` : utilisé pour la génération de maillage.

```
1 pip install pygmsh==7.1.17
```

- `mpi4py` : pour l'exécution parallèle et l'utilisation de MPI.

```
1 pip install mpi4py==3.1.3
```

- `h5py` : pour manipuler les fichiers HDF5.

```
1 pip install h5py==3.8.0
```

Il est fortement recommandé d'installer ces bibliothèques pour garantir la bonne exécution du code de JAX-CPFEM.

5 Installation des Dépendances de Neper

Neper nécessite des outils de compilation spécifiques.

1. Installez les outils de compilation :

```
1 sudo apt-get install build-essential libtool libfftw3-dev gfortran cmake
```

6 Installation de GSL (GNU Scientific Library)

1. Installez GSL via Conda :

```
1 conda install -c conda-forge gsl
```

7 Installation de Neper

1. Clonez le dépôt de Neper depuis GitHub :

```
1 git clone https://github.com/neperfepx/neper.git  
2 git pull origin main
```

2. Allez dans le répertoire 'src' de Neper : Il faut se situer dans le dossier 'src' pour effectuer la compilation :

```
1 cd neper/src
```

3. Créez les fichiers de compilation avec CMake :

```
1 cmake .
```

4. Compilez Neper avec la commande 'make' :

```
1 make
```

8 Problèmes éventuels et solutions

8.1 Problèmes avec les dépendances JAX et jaxlib

Lors de l'installation de JAX-FEM ou de JAX-CPFEM, il est possible de rencontrer des erreurs liées à 'jaxlib', telles que des erreurs de compatibilité avec certaines architectures de CPU (par exemple, la version de 'jaxlib' construite avec les instructions AVX ne fonctionne pas sur certains processeurs). Si cela se produit, il est possible de contourner ce problème en réinstallant 'jaxlib' avec la commande suivante :

```
1 pip install --no-binary jaxlib jaxlib
```

Cela forcera une compilation de 'jaxlib' à partir des sources, ce qui permet de résoudre les problèmes liés à l'AVX. En revanche, cette méthode peut être longue et nécessite un compilateur C++ compatible.

8.2 Problèmes lors de la compilation de Neper

Lors de la compilation de Neper, si vous rencontrez des erreurs liées à des dépendances manquantes (par exemple, ‘libscotch’), assurez-vous que toutes les bibliothèques requises sont installées avec les commandes suivantes :

```
1 sudo apt-get install libscotch-dev
2 sudo apt-get install libmetis-dev
```

Ces bibliothèques sont nécessaires pour que Neper fonctionne correctement et qu’il puisse être compilé avec succès.

9 Test de l’Installation

Après avoir installé toutes les dépendances et logiciels nécessaires, il est important de vérifier que tout fonctionne correctement. Voici les étapes pour tester l’installation de JAX-FEM, JAX-CPFEM et Neper.

9.1 Test de JAX-FEM et JAX-CPFEM

1. Testez JAX-FEM en exécutant un exemple simple. Par exemple, pour tester l’exemple d’hyperélasticité :

```
1 python -m demos.hyperelasticity.example
```

Cette commande lance un test sur un modèle d’hyperélasticité pour vérifier que l’installation de JAX-FEM est correcte.

2. Ensuite, pour tester un exemple dans JAX-CPFEM, comme celui de ‘polycrystal_304steel’, vous pouvez d’abord déplacer le dossier ‘polycrystal_304steel’ de JAX-CPFEM dans le dossier ‘applications’ de JAX-FEM. Une fois cela fait, vous pourrez exécuter la commande suivante (terminal ouvert dans le dossier jax-fem) pour vérifier que la simulation de plasticité cristalline fonctionne bien :

```
1 python -m applications.polycrystal_304steel.polycrystal_304steel
```