

Wrapping du code avec Makefile

2 Juillet 2020

1 Introduction

Le fichier **Makefile** écrit a pour but de générer automatiquement des fichiers d'interface Python-Fortran en utilisant la générateur **f90wrap**, une version améliorée de la générateur d'interface **f2py**, utilisée spécialement pour le code du type dérivé. Les répertoires contenant les fichiers source se placent dans **src/cases_test** et les résultats se trouvent dans **build/wrappers**. Le wrapping et le test basent sur 3 étapes:

- Copier et compiler les fichiers sources Fortran:

```
$ make files
```

- Faire le wrapping du code:

```
$ make
```

- Lancer le fichier Python placé dans **bin** pour tester:

```
$ source ./scripts/env.sh  
$ make run
```

Notons que l'on doit d'abord modifier un bon chemin pour la variable **PYTHONPATH** dans le fichier **./scripts/env.sh**. On peut supprimer les fichiers wrapped chaque fois qu'on relance par la commande:

```
$ make clean
```

On considère les 3 cas à tester ci-dessous.

2 Cas de test

2.1 Wrapping pour une seule fichier

On peut voir dans le fichier **src/cases_test/simple_case/test.f90** le type **real_array** qui est défini comme un vecteur de dimension 6 et le subroutine **testf()** qui prend une variable de type de **real_array** et renvoie un vecteur de même dimension dont l'indice 4 est modifié. Le fichier **bin/simple_case.py** permet de vérifier si le wrapping est correct en appelant et lancer les modules **real_array** et **testf()**.

2.2 Wrapping pour deux fichiers

On considère maintenant le fichier calculant le volume d'un cylindre **cyldnad.f90** dans **src/cases_test/example_cylinder** en prenant une variable de dimension définie par un autre fichier dans un même répertoire **DNAD.f90** [1]. La difficulté ici est que l'on doit faire le wrapping pour les deux fichiers dans lesquels l'un dépend de l'autre. La solution est que l'on compile et combine tous ces fichiers dans un fichier **libsrc.a** et génère les fichiers d'interface basés sur ce fichier. Le fichier Python **bin/example_cylinder.py** calcule le volume d'un cylindre en appelant les modules des fichiers viennent d'être wrappé.

References

- [1] James R Kermode. f90wrap: an automated tool for constructing deep python interfaces to modern fortran codes. *J. Phys. Condens. Matter*, March 2020.