

<https://haydn2005.u-bourgogne.fr/dnum-ccub/spip.php?article787>

Compiler vos codes en FORTRAN, C ou C++

- Site Public - Calcul - Logiciels -

Date de mise en ligne : lundi 8 avril 2013

Copyright © Site du Centre de Calcul de l'Université de Bourgogne - Tous
droits réservés

Compilateurs et librairies disponibles

Les compilateurs disponibles sont :

- ▶ **Intel (recommandé)** : ifort pour fortran, icc/icpc pour C/C++
- ▶ **GNU** : gfortran pour fortran, gcc/g++ pour C/C++
- ▶ **Portland** : pgf90 pour fortran, pgcc/pgCC pour C/C++

Remarque : l'outil d'Intel pour générer les bonnes variables de compilation :

<https://software.intel.com/en-us/articles/intel-mkl-link-line-advisor>

Les librairies permettant le développement d'applications parallèles sont :

- ▶ **open-mpi**

L'accès aux compilateurs et librairies se fait par l'intermédiaire de l'[Environnement logiciel : modules](#)

Exemple :

- ▶ pour charger le module correspondant au compilateur Intel fortran (v. 2018) :

```
module load intel/2018
```

- ▶ pour charger le module correspondant au compilateur Intel fortran (v. 2018) et la librairie open-mpi :

```
module load openmpi/3.1.6/intel/2018
```

Compilation en séquentiel

Différentes options sont disponibles (cf. le manuel du compilateur) :

- ▶ Debug : -g (ifort -g -o code.exe code.f)
- ▶ Optimisation : -axCORE-AVX512 -O3 (ifort -axAVX -O3 -o code.exe code.f)
- ▶ Précision : -fp-model precise -pc 64

Exemple compiler du C avec Intel :

```
module load intel/2018
icc -axCORE-AVX512 -O3 -fp-model precise -pc 64 -o code.exe code.c
```

Exemple compiler du C/C++ avec Intel :

```
module load intel/2018  
icpc -xCORE-AVX512 -O3 -fp-model precise -pc 64 -o code.exe code.c
```

Exemple compiler du fortran avec Intel :

```
module load intel/2018
ifort -axCORE-AVX512 -O3 -fp-model precise -pc 64 -o code.exe code.f
```

Exemple utiliser les librairies BLAS et LAPACK (avec des entiers 32 bits) et le compilateur Intel :

```
module load intel/2018
OPTI_FLAGS="-axCORE-AVX512 -O3 -fp-model precise -pc 64"
FFLAGS="-I$MKLROOT/include"
LD_FLAGS="-L$MKLROOT/lib/intel64 -lmkl_blas95_lp64 -lmkl_lapack95_lp64
-lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm"
ifort -o test test.f $OPTI_FLAGS $FFLAGS $LD_FLAGS
```

Exemple utiliser les librairies BLAS et LAPACK (avec des entiers 64 bits) et le compilateur Intel :

```
module load intel/2018
OPTI_FLAGS="-axCORE-AVX512 -O3 -fp-model precise -pc 64 -i8"
FFLAGS="-I$MKLROOT/include"
LD_FLAGS="-L$MKLROOT/lib/intel64 -lmkl_blas95_ilp64 -lmkl_lapack95_ilp64
-lmkl_intel_ilp64 -lmkl_sequential -lmkl_core -lpthread -lm"
ifort -o test test.f $OPTI_FLAGS $FFLAGS $LD_FLAGS
```

Vous trouverez sur le site internet d'intel une interface qui vous permettra de préciser les options à utiliser pour linker votre application avec la librairie MKL : <http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor>

En cas de doute, veuillez svp contacter l'équipe du ccub.

Compilation en parallèle avec Intel

Cas de MPI

On utilise un compilateur associé à la librairie mpi : mpif90, mpicc ou mpicxx

Pour utiliser les fonctions, méthodes, routines MPI, les fichiers mpi.h ou mpif.h doivent être inclus au sein des codes source et le code doit être lié avec une librairie MPI.

Compilation C :

```
module load openmpi/3.1.6/intel/2018
mpicc -axCORE-AVX512 -O3 -fp-model precise -pc 64 -o code.exe code.c
```

Compilation C++ :

```
module load openmpi/3.1.6/intel/2018
mpicxx -axCORE-AVX512 -O3 -fp-model precise -pc 64 -o code.exe code.c
```

Compilation fortran :

```
module load openmpi/3.1.6/intel/2018
mpif90 -axCORE-AVX512 -O3 -fp-model precise -pc 64 -o code.exe code.f
```

Cas de OpenMP

Si OpenMP pour la parallélisation, l'option -openmp doit être passée au compilateur intel.

Compilation C/C++ :

```
module load openmpi/3.1.6/intel/2018
icc -openmp -axCORE-AVX512 -O3 -fp-model precise -pc 64 -o code.exe code.c
```

Compilation C++ :

```
module load openmpi/3.1.6/intel/2018
icpc -openmp -axCORE-AVX512 -O3 -fp-model precise -pc 64 -o code.exe code.c
```

Compilation fortran :

```
module load openmpi/3.1.6/intel/2018
ifort -openmp -axCORE-AVX512 -O3 -fp-model precise -pc 64 -o code.exe code.f
```

Résumé de l'utilisation du compilateur INTEL avec AVX

	Séquentiel	MPI	OpenMP	Options conseillées
Fortran	ifort	mpif90	ifort -openmp	-axCORE-AVX512 -O3 -fp-model precise -pc 64
C	icc	mpicc	icc -openmp	-axAVX -O3 -fp-model precise -pc 64

C++	icpc	mpicxx	icpc -openmp	-axCORE-AVX512 -O3 -fp-model precise -pc 64
-----	------	--------	--------------	---

Résumé de l'utilisation du compilateur PGI - PORTLAND sans AVX

	Séquentiel	MPI	OpenMP	Options conseillées
Fortran	pgf90	mpif90	pgf90 -mp	-O3 -Kieee -pc 64
C	pgcc	mpicc	pgcc -mp	-O3 -Kieee -pc 64
C++	pgCC	mpicxx	pgcc -mp	-O3 -Kieee -pc 64

Résumé de l'utilisation du compilateur GNU sans AVX

	Séquentiel	MPI	OpenMP	MPI et OpenMP
Fortran	ifort	mpif90	ifort -fopenmp	-O3 -mieee -fp
C	gcc	mpicc	gcc -fopenmp	-O3 -mieee -fp
C++	g++	mpicxx	g++ -fopenmp	-O3 -mieee -fp