

## Quantum Espresso 7.3

### ウェブページ

<https://www.quantum-espresso.org/>

### バージョン

7.3

### ビルド環境

- GCC 10.3.1 (gcc-toolset-10)
- Open MPI 4.1.5
- OpenBLAS 0.3.26 (lp64)
- Scalapack 2.2.0
- ELPA 2023.11.001

### ビルドに必要なファイル

- [qe-7.3-ReleasePack.tar.gz](#)
- [elpa-2023.11.001.tar.gz](#)
- [.gitmodules](#)
  - なぜか ReleasePack には入っていないが、W90 などのビルドには必要？

### ビルド手順

#### ELPA 2023.11.001

```
#!/bin/sh

ELPA_VERSION=2023.11.001
INSTDIR=/apl/qe/7.3/elpa-2023.11.001
WORKDIR=/gwork/users/${USER}

BASEDIR=/home/users/${USER}/Software/ELPA/${ELPA_VERSION}
TARBALL=${BASEDIR}/elpa-${ELPA_VERSION}.tar.gz

PARALLEL=12

#-----
umask 0022
ulimit -s unlimited

module purge
module load gcc-toolset/10
module load openmpi/4.1.5/gcc10
module load openblas/0.3.26-lp64
module load scalapack/2.2.0-mpi416gcc12-lp64

export LANG=C
export LC_ALL=C

export FC=mpif90
export CC=mpicc
export CXX=mpicxx
export CFLAGS="-march=znver3"
export FCFLAGS="-m64"
export LDFLAGS="-lopenblas -lscalapack"

cd ${WORKDIR}
if [ -d elpa-${ELPA_VERSION} ]; then
  mv elpa-${ELPA_VERSION} elpa-erase
  rm -rf elpa-erase &
```

```

fi
tar zxf ${TARBALL}
cd elpa-${ELPA_VERSION}

./configure --prefix=${INSDIR} \
    --enable-openmp \
    --disable-avx512-kernels
make -j ${PARALLEL}
make check
#make check && make install
make install

```

## QE 7.3

```

#!/bin/sh

QE_VERSION=7.3
BASEDIR=/home/users/${USER}/Software/QE/${QE_VERSION}
TARBALL=${BASEDIR}/qe-${QE_VERSION}-ReleasePack.tar.gz
GITMODULES=${BASEDIR}/.gitmodules

ELPA=/apl/qe/7.3/elpa-2023.11.001
ELPAINC=${ELPA}/include/elpa_openmp-2023.11.001/modules \
ELPALIB=${ELPA}/lib/libelpa_openmp.a

INSDIR=/apl/qe/7.3
PARALLEL=12

# -----
umask 0022

module -s purge
module -s load gcc-toolset/10
module -s load openmpi/4.1.5/gcc10
module -s load openblas/0.3.26-lp64
module -s load scalapack/2.2.0-ompi416gcc12-lp64
## gui; not necessary while building
#module -s load itcl/3.4.4
#module -s load itk/3.4.2
#module -s load iwidgets/4.1.1

export LANG=C
export LC_ALL=C
ulimit -s unlimited

if [ ! -d ${INSDIR} ]; then
    mkdir -p ${INSDIR}
fi

cd ${INSDIR}
if [ -d qe-${QE_VERSION} ]; then
    mv qe-${QE_VERSION} qe-erase
    rm -rf qe-erase &
fi

tar zxf ${TARBALL}
mv qe-${QE_VERSION}/* .
mv qe-${QE_VERSION}/.[a-z]* .
rmdir qe-${QE_VERSION}

sed -i -e "s/wget -O/wget --trust-server-names -O/" \
    -e "s/curl -o/curl -L -o/" test-suite/check_pseudo.sh

export MPIF90=mpif90
export MPIFC=mpif90

```

```

export MPIF77=mpif90
export MPICC=mpicc
export MPICXX=mpicxx
export BLAS_LIBS="-lopenblas"
export SCALAPACK_LIBS="-lopenblas -lscalapack"

cp ${GITMODULES} .
rm -rf external/wannier90
mkdir -p external/wannier90

sed -i -e '/external/wannier90/s/lib/wannier lib/' install/plugins_makefile

FC=gfortran F90=gfortran F77=gfortran CC=gcc CXX=g++ \
./configure --enable-parallel \
--enable-openmp \
--with-scalapack \
--with-elpa-include=$ELPAINC \
--with-elpa-lib=$ELPALIB

for i in w90; do
echo "==== $i ====="
make $i
done

# pwall(pw neb ph pp pwcond acfdt) cp ld1 tddfpt hp xspectra gw
echo "==== all ====="
make -j${PARALLEL} all

# gcc10/11/12 can't build want
#for i in want; do
# echo "==== $i ====="
# make $i
#done

# gipaw for QE 7.3 doesn't seem to be available
# d3q depends on old version of PH code? (setlocq, setlocq_coul)

for i in all_currents epw couple kcw gw gui; do
echo "==== $i ====="
make -j${PARALLEL} $i
done

# gcc10/11/12 can't build yambo
#for i in yambo; do
# echo "==== $i ====="
# make $i
#done

cd test-suite
export OMP_NUM_THREADS=1
make run-tests-pw NPROCS=1
make run-tests-cp NPROCS=1
make run-tests-ph NPROCS=1
make run-tests-epw NPROCS=1
make run-tests-hp NPROCS=1
make run-tests-tddfpt NPROCS=1
make run-tests-kcw NPROCS=1
make run-tests-all_currents NPROCS=1
make run-tests-pp NPROCS=1
make run-tests-zg NPROCS=1
#make run-tests-xsd-pw NPROCS=1
make clean
export OMP_NUM_THREADS=2
make run-tests-pw NPROCS=4
make run-tests-cp NPROCS=4

```

```
make run-tests-ph NPROCS=4
make run-tests-epw NPROCS=4
make run-tests-hp NPROCS=4
make run-tests-tddfpt NPROCS=4
make run-tests-kcw NPROCS=4
make run-tests-all_currents NPROCS=4
make run-tests-pp NPROCS=4
make run-tests-zg NPROCS=4
#make run-tests-xsd-pw NPROCS=4
cd ..
```

## テスト結果(ELPA)

```
# TOTAL: 311
# PASS: 168
# SKIP: 109
# XFAIL: 0
# FAIL: 34
# XPASS: 0
# ERROR: 0
```

### 失敗したテストのリスト

- validate\_c\_version\_complex\_double\_eigenvalues\_2stage\_default\_kernel\_analytic\_default.sh
- validate\_c\_version\_real\_double\_eigenvalues\_2stage\_default\_kernel\_analytic\_default.sh
- validate\_c\_version\_complex\_single\_eigenvalues\_2stage\_default\_kernel\_analytic\_default.sh
- validate\_c\_version\_real\_single\_eigenvalues\_2stage\_default\_kernel\_analytic\_default.sh
- validate\_c\_version\_complex\_double\_eigenvectors\_2stage\_default\_kernel\_random\_explicit\_default.sh
- validate\_c\_version\_complex\_double\_eigenvectors\_2stage\_default\_kernel\_random\_default.sh
- validate\_c\_version\_real\_double\_eigenvectors\_2stage\_default\_kernel\_random\_explicit\_default.sh
- validate\_c\_version\_real\_double\_eigenvectors\_2stage\_default\_kernel\_random\_default.sh
- validate\_c\_version\_complex\_single\_eigenvectors\_2stage\_default\_kernel\_random\_explicit\_default.sh
- validate\_c\_version\_complex\_single\_eigenvectors\_2stage\_default\_kernel\_random\_default.sh
- validate\_c\_version\_real\_single\_eigenvectors\_2stage\_default\_kernel\_random\_explicit\_default.sh
- validate\_c\_version\_real\_single\_eigenvectors\_2stage\_default\_kernel\_random\_default.sh
- validate\_c\_version\_real\_single\_eigenvectors\_2stage\_default\_kernel\_random\_explicit\_default.sh
- validate\_c\_version\_real\_single\_eigenvectors\_2stage\_default\_kernel\_random\_default.sh
- validate\_c\_version\_complex\_single\_eigenvectors\_2stage\_default\_kernel\_random\_explicit\_default.sh
- validate\_c\_version\_complex\_single\_eigenvectors\_2stage\_default\_kernel\_random\_default.sh
- validate\_c\_version\_real\_single\_eigenvectors\_2stage\_default\_kernel\_random\_explicit\_default.sh
- validate\_c\_version\_real\_single\_eigenvectors\_2stage\_default\_kernel\_random\_default.sh
- validate\_real\_double\_eigenvalues\_2stage\_default\_kernel\_frank\_default.sh
- validate\_complex\_double\_eigenvalues\_2stage\_default\_kernel\_toeplitz\_default.sh
- validate\_real\_double\_eigenvalues\_2stage\_default\_kernel\_toeplitz\_default.sh
- validate\_complex\_single\_eigenvalues\_2stage\_default\_kernel\_toeplitz\_default.sh
- validate\_real\_single\_eigenvalues\_2stage\_default\_kernel\_toeplitz\_default.sh
- validate\_double\_instance\_openmp\_default.sh
- validate\_real\_2stage\_banded\_openmp\_default.sh
- validate\_complex\_2stage\_banded\_openmp\_default.sh
- validate\_single\_real\_2stage\_banded\_openmp\_default.sh
- validate\_single\_complex\_2stage\_banded\_openmp\_default.sh

## テスト結果: QE serial

pw: 246 out of 246 tests passed (1 skipped).  
cp: 33 out of 33 tests passed (2 skipped).  
ph: 62 out of 62 tests passed.  
epw: 135 out of 135 tests passed.  
hp: 39 out of 41 tests passed.

- hp\_soc\_UV\_paw\_magn - bn.hp.in (arg(s): 3): **\*\*FAILED\*\***.
- hp\_soc\_UV\_paw\_magn - bn.hp.in (arg(s): 4): **\*\*FAILED\*\***.

tddfpt: 9 out of 9 tests passed.

kcw: 11 out of 11 tests passed.

all\_currents: 10 out of 10 tests passed.

pp: 2 out of 2 tests passed.

zq: 1 out of 1 test passed.

#### テスト結果: QE parallel

pw: 246 out of 246 tests passed (1 skipped).

cp: 33 out of 33 tests passed (2 skipped).

ph: 62 out of 62 tests passed.

epw: 133 out of 135 tests passed.

- epw\_wfpt - epw22.in (arg(s): 3): **\*\*FAILED\*\***.
- epw\_wfpt - epw23.in (arg(s): 3): **\*\*FAILED\*\***.

hp: 39 out of 41 tests passed.

- hp\_soc\_UV\_paw\_magn - bn.hp.in (arg(s): 3): **\*\*FAILED\*\***.
- hp\_soc\_UV\_paw\_magn - bn.hp.in (arg(s): 4): **\*\*FAILED\*\***.

tddfpt: 9 out of 9 tests passed.

kcw: 11 out of 11 tests passed.

all\_currents: 10 out of 10 tests passed.

pp: 2 out of 2 tests passed.

zq: 1 out of 1 test passed.

#### メモ

- 1 ノード 128 コアを使う規模になると OpenMP で速度が上がる可能性が高いため、OpenMP を有効化
- ELPA を有効化。gcc10,11,12 ではテストを完全にパスしない。インテルコンパイラでは全て通る
- gipaw は QE 7.3 向けのものがまだ存在しないようであるためパス
- d3q (QE 7.1 向けが最新?)は古いバージョンの PH のコードに依存しているようで、ビルドできないためパス
- gcc10,11,12 では want と yambo のビルドに失敗する
  - インテルコンパイラであれば問題なく通る
- QE (主に PW)についてはインテルコンパイラよりも gcc でビルドした方が速度が出やすい
- gcc10,11,12 ではテスト結果が大体同じで、PW の速度もほぼ同水準に見える。PW のテストが全部通る gcc10 をとりあえず採用。
- MKL を使うよりも openblas+scalapack を使う方が PW の速度が出る。
  - インテルコンパイラを使った場合でも MKL よりも若干 openblas+scalapack の方が PW の速度が出る。