Siesta 5.4.1

ウェブページ

https://gitlab.com/siesta-project/siesta

バージョン

5.4.1 (+ hdf5-1.14.6, NetCDF 4.9.3, NetCDF Fortran 4.6.2, libxc 7.0.0)

ビルド環境

- GCC 14.2.1 (gcc-toolset-14)
- Open MPI 4.1.8
- MKL 2025.2.0
- Python 3.9
    - rumael.yaml (pip3.9 install ruamel.yaml --user)

ビルドに必要なファイル

- siesta-5.4.1.tar.gz
- wannier90-3.1.0.tar.gz
- hdf5-1.14.6.tar.gz
- netcdf-c-4.9.3.tar.gz
- netcdf-fortran-4.6.2.tar.gz
- libxc-7.0.0.tar.bz2
- (下記手順内で複数ソフトが自動的に導入されています)

ビルド手順

```
#!/bin/sh

SIESTA_VERSION=5.4.1
INSTDIR=/apl/siesta/5.4.1

WORKDIR=/gwork/users/${USER}
BASEDIR=/home/users/${USER}/Software/Siesta/${SIESTA_VERSION}
TARBALL=${BASEDIR}/siesta-${SIESTA_VERSION}.tar.gz

HDF5_VERSION=1.14.6
BASEDIR_HDF5=/home/users/${USER}/Software/HDF5/${HDF5_VERSION}
TARBALL_HDF5=${BASEDIR_HDF5}/hdf5-${HDF5_VERSION}.tar.gz
NETCDF_C_VERSION=4.9.3
NETCDF_F_VERSION=4.6.2
BASEDIR_NETCDF=/home/users/${USER}/Software/NETCDF
TARBALL_NETCDF_C=${BASEDIR_NETCDF}/c${NETCDF_C_VERSION}/netcdf-c-${NETCDF_C_VERS
ION}.tar.gz
TARBALL_NETCDF_F=${BASEDIR_NETCDF}/f${NETCDF_F_VERSION}/netcdf-fortran-${NETCDF_
F_VERSION}.tar.gz
LIBXC_VERSION=7.0.0
BASEDIR_LIBXC=/home/users/${USER}/Software/libxc
TARBALL_LIBXC=${BASEDIR_LIBXC}/${LIBXC_VERSION}/libxc-${LIBXC_VERSION}.tar.bz2
WANNIER90_VERSION=3.1.0
BASEDIR_WANNIER90=/home/users/${USER}/Software/wannier90/${WANNIER90_VERSION}
TARBALL_WANNIER90=${BASEDIR_WANNIER90}/wannier90-${WANNIER90_VERSION}.tar.gz

PARALLEL=12

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

module -s purge
module -s load gcc-toolset/14
```

```
module -s load openmpi/4.1.8/gcc14
module -s load mkl/2025.2.0

export LANG=C
export LC_ALL=C
export OMP_NUM_THREADS=1

# libxc

if [ ! -f ${INSTDIR}/exts/lib/libxc.a ]; then
 cd ${WORKDIR}
 if [ -d libxc-${LIBXC_VERSION} ]; then
   mv libxc-${LIBXC_VERSION} libxc-erase
   rm -rf libxc-erase &
 fi
 tar xf ${TARBALL_LIBXC}
 cd libxc-${LIBXC_VERSION}
 autoreconf -i
 ./configure --prefix=${INSTDIR}/exts
 make -j${PARALLEL}
 make -j${PARALLEL} check
 make install
fi

# hdf5

if [ ! -f ${INSTDIR}/exts/lib/libhdf5.a ]; then
 cd ${WORKDIR}
 if [ -d hdf5-${HDF5_VERSION} ]; then
   mv hdf5-${HDF5_VERSION} hdf5-erase
   rm -rf hdf5-erase &
 fi
 tar zxf ${TARBALL_HDF5}
 cd hdf5-${HDF5_VERSION}
 mkdir build && cd build
 cmake .. \
     -DCMAKE_INSTALL_PREFIX=${INSTDIR}/exts \
     -DHDF5_BUILD_FORTRAN=ON \
     -DHDF5_ENABLE_PARALLEL=ON \
     -DMPIEXEC_MAX_NUMPROCS=${PARALLEL}
 make -j${PARALLEL}
 make install
 make test # very long...
fi

export CC=mpicc
export CXX=mpicxx
export FC=mpif90
export F90=mpif90

export PATH="${INSTDIR}/exts/bin:${PATH}"
export CPATH="${INSTDIR}/exts/include:${CPATH}"
export LD_LIBRARY_PATH="${INSTDIR}/exts/lib:${LD_LIBRARY_PATH}"
export LIBRARY_PATH="${INSTDIR}/exts/lib:${LIBRARY_PATH}"

# netcdf-c

if [ ! -f ${INSTDIR}/exts/lib/libnetcdf.a ]; then
 cd ${WORKDIR}
 if [ -d netcdf-c-${NETCDF_C_VERSION} ]; then
   mv netcdf-c-${NETCDF_C_VERSION} netcdf-c-erase
   rm -rf netcdf-c-erase &
 fi
 tar zxf ${TARBALL_NETCDF_C}
 cd netcdf-c-${NETCDF_C_VERSION}
```

```
 LDFLAGS="-L${INSTDIR}/exts/lib" \
   ./configure --prefix=${INSTDIR}/exts \
               --enable-parallel-tests
 make -j${PARALLEL}
 make -j${PARALLEL} check
 make install
fi

# netcdf-f

if [ ! -f ${INSTDIR}/exts/lib/libnetcdff.a ]; then
 cd ${WORKDIR}
 if [ -d netcdf-fortran-${NETCDF_F_VERSION} ]; then
   mv netcdf-fortran-${NETCDF_F_VERSION} netcdf-fortran-erase
   rm -rf netcdf-fortran-erase &
 fi
 tar zxf ${TARBALL_NETCDF_F}
 cd netcdf-fortran-${NETCDF_F_VERSION}

 LDFLAGS="-L${INSTDIR}/exts/lib" \
     ./configure --prefix=${INSTDIR}/exts \
                 --enable-parallel-tests
 make -j${PARALLEL}
 make -j${PARALLEL} check
 make install
fi

# siesta

cd ${WORKDIR}
if [ -d siesta-${SIESTA_VERSION} ]; then
 mv siesta-${SIESTA_VERSION} siesta-erase
 rm -rf siesta-erase
fi

tar zxf ${TARBALL}
cd siesta-${SIESTA_VERSION}

unset CC
unset CXX
unset FC
unset F90
export WANNIER90_PACKAGE=${TARBALL_WANNIER90}

mkdir build && cd build
cmake .. \
 -DCMAKE_INSTALL_PREFIX="${INSTDIR}" \
 -DCMAKE_PREFIX_PATH="${INSTDIR}/exts" \
 -DCMAKE_C_COMPILER=mpicc \
 -DCMAKE_CXX_COMPILER=mpicxx \
 -DCMAKE_Fortran_COMPILER=mpif90 \
 -DPython3_EXECUTABLE=/usr/bin/python3.9 \
 -DSIESTA_WITH_MPI=ON \
 -DBLAS_LIBRARY="-m64 -L${MKLROOT}/lib -Wl,--no-as-needed -lmkl_scalapack_lp64 -lmkl_gf_lp64 -lmkl_sequential -lmkl_core -
lmkl_blacs_openmpi_lp64 -lpthread -lm -ldl" \
 -DLAPACK_LIBRARY=NONE \
 -DSCALAPACK_LIBRARY=NONE \
 -DNetCDF_PARALLEL=ON \
 -DNetCDF_ROOT="${INSTDIR}/exts" \
 -DSIESTA_WITH_OPENMP=OFF \
 -DSIESTA_WITH_DFTD3=ON \
 -DSIESTA_WITH_LIBXC=ON \
 -DSIESTA_WITH_ELSI=ON \
 -DSIESTA_WITH_WANNIER90=ON

make -j ${PARALLEL}
```

```
SIESTA_TESTS_VERIFY=1 ctest
make install
cp -r Testing/Temporary ${INSTDIR}/testlog

cd ../
cp -r Examples ${INSTDIR}
```

テスト

hdf5-1.14.6

以下のテストは実行されず。

- 915 - H5REPACK-szip_individual (Disabled)
- 916 - H5REPACK-szip_all (Disabled)
- 933 - H5REPACK-all_filters (Disabled)
- 937 - H5REPACK-szip_copy (Disabled)
- 938 - H5REPACK-szip_remove (Disabled)
- 987 - H5REPACK-remove_all (Disabled)
- 988 - H5REPACK-deflate_convert (Disabled)
- 989 - H5REPACK-szip_convert (Disabled)

libxc, NetCDF-C, NetCDF-Fortran

すべてパス

Siesta

以下のテストに失敗。テストログのコピーは /apl/siesta/5.4.1/testlog 以下にあります。

- 226 - siesta-02.SpinPolarization-fe_spin_mpi4[verify] (Failed)
- 230 - siesta-02.SpinPolarization-fe_spin_directphi_mpi4[verify] (Failed)
- 238 - siesta-02.SpinPolarization-fe_noncol_gga_mpi4[verify] (Failed)
- 246 - siesta-02.SpinPolarization-fe_noncol_sp_mpi4[verify] (Failed)
- 250 - siesta-03.SpinOrbit-FePt-X-X_mpi4[verify] (Failed)
- 278 - siesta-04.SCFMixing-chargemix_mpi4[verify] (Failed)

メモ

- 5.2.2 とは異なり、Open MPI と MKL を利用。
  - OpenBLAS よりも少し速度が出そうです。
- Intel コンパイラ(2024, 2025 で検証。Classic コンパイラは未検証)を使うとテストの失敗が増えるため GCC を利用。
- Intel MPI と GCC の組み合わせではビルドに失敗するために回避。
  - Intel MPI の mpi_f08.mod が使える場合は問題ありませんが、mpi.mod を使う場合、MPI_Allreduce の MPI_IN_PLACE 指定が原因でビルドに失敗します。
    - 参考: https://community.intel.com/t5/Intel-MPI-Library/Bug-in-Intel-MPI-MPI-Allreduce-quot-use-mpi-quot-implementation/m-p/1532900
  - Intel MPI では gfortran 用の mpi.mod は存在しますが、mpi_f08.mod は存在しないようです。
  - インテルコンパイラ + mpi_f08.mod の組み合わせならビルド可能です。
    - インテルコンパイラを使い、-DSIESTA_WITH_MPI_INTERFACES=legacy を指定して mpi.mod を使うと GCC の時と同様にビルドに失敗します。
    - 今のところ Intel MPI で mpi.mod を使う条件では Siesta 5.4.1 をビルドできていません。
  - Open MPI では -DSIESTA_WITH_MPI_INTERFACES=legacy を指定して mpi.mod を使っても問題なく動作します。
- MVAPICH を使った場合もエラーが増えるようです。