Siesta 5.2.2

ウェブページ

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

バージョン

5.2.2 (+ hdf5-1.14.5, NetCDF 4.9.2, NetCDF Fortran 4.6.1, libxc 6.2.2)

ビルド環境

- GCC 13.1.1 (gcc-toolset-13)
- Intel MPI 2021.14.1
- OpenBLAS 0.3.29 (lp64)
- ScaLAPACK 2.2.2
- Python 3.9
    - rumael.yaml (pip3.9 install ruamel.yaml --user)

ビルドに必要なファイル

- siesta-5.2.2.tar.gz
- wannier90-3.1.0.tar.gz
- hdf5-1.14.5.tar.gz
- netcdf-c-4.9.2.tar.gz
- netcdf-fortran-4.6.1.tar.gz
- libxc-6.2.2.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=5.2.2
INSTDIR=/apl/siesta/5.2.2

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

HDF5_VERSION=1.14.5
BASEDIR_HDF5=/home/users/${USER}/Software/HDF5/${HDF5_VERSION}
TARBALL_HDF5=${BASEDIR_HDF5}/hdf5-${HDF5_VERSION}.tar.gz
NETCDF_C_VERSION=4.9.2
NETCDF_F_VERSION=4.6.1
BASEDIR_NETCDF=/home/users/${USER}/Software/NETCDF
TARBALL_NETCDF_C=${BASEDIR_NETCDF}/c${NETCDF_C_VERSION}/netcdf-c-${NETCDF_C_VERSION}.tar.gz
TARBALL_NETCDF_F=${BASEDIR_NETCDF}/f${NETCDF_F_VERSION}/netcdf-fortran-${NETCDF_F_VERSION}.tar.gz
LIBXC_VERSION=6.2.2
BASEDIR_LIBXC=/home/users/${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/13
module -s load intelmpi/2021.14.1
```

```
module -s load openblas/0.3.29-lp64
module -s load scalapack/2.2.2-impi2021gcc-lp64

export LANG=C
export LC_ALL=C
export OMP_NUM_THREADS=1
#export I_MPI_HYDRA_TOPOLIB=ipl

# 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
 export FC="mpif90 -I/apl/oneapi/mpi/2021.14.1/include"
 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="-lopenblas -lscalapack" \
 -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

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)

以下のテストでエラー

- 2930 - MPI_TEST_H5_f90_ph5_f90_filtered_writes_no_sel (Failed)

NetCDF-C, NetCDF-Fortran, libxc

全てパス

Siesta

ログは /apl/siesta/5.2.2/testlog 以下にあります

- 61 - siesta-02.SpinPolarization-fe_spin_mpi4[verify] (Failed)
- 63 - siesta-02.SpinPolarization-fe_spin_directphi_mpi4[verify] (Failed)
- 67 - siesta-02.SpinPolarization-fe_noncol_gga_mpi4[verify] (Failed)
- 71 - siesta-02.SpinPolarization-fe_noncol_sp_mpi4[verify] (Failed)
- 73 - siesta-03.SpinOrbit-FePt-X-X_mpi4[verify] (Failed)
- 87 - siesta-04.SCFMixing-chargemix_mpi4[verify] (Failed)

メモ

- Intel MPI を使っていて、ある程度以上並列した場合のエラー(参考: 5.0.0 導入時の情報)は export I_MPI_HYDRA_TOPOLIB=ipl で解消することを確認
  - こちらで使っているキューイングシステム環境依存のエラーである可能性が高いです。通常の環境では発生しない可能性が高いと思われます。
  - 上記環境変数を設定した場合、ログの最後に IPL WARN> IPL_init_numa_nodes: can not define numa node num のようなメッセージが出てしまいます
  - 上記 siesta 5.2.2 ビルド手順で export I_MPI_HYDRA_TOPOLIB=ipl を設定した場合、HDF5 のテストでエラーが大量に発生します
  - このエラーが発生するケースはそれなりに限られているため、多くの場合は unset I_MPI_HYDRA_TOPOLIB しても大丈夫かと思います。
- Open MPI (4.1.8 で試行)を使うより Intel MPI を使う方が若干性能が良さそうに見えます
- MKL を使うより OpenBLAS を使う方が若干速いように見える。
- AOCC ではビルドが大変そうな気配だったため検討せず。
- Intel oneAPI コンパイラ(icx,icpx,ifx)を使うと Siesta のテスト中 90 個弱(全体の約 30 %)で失敗してしまうため、今回は回避しました。
  - ただし、計算の一部が妙に速いため、目的や状況次第では非常に有用かもしれません。
  - Classic コンパイラ(oneAPI 2023, icc+icpc+ifort)ではエラーは出ませんが速度もあまり出ませんでした。
- Python 3.9 と ruamel.yaml はテストの際に使っていますが、通常の Siesta の利用には必要ありません。(参考:5.0.0 導入時の情報)