1 Design

The current version of the CP2K/SMEAGOL interface is based on the original version of SMEAGOL code linked with SIESTA 1.3f1. Although the SMEAGOL and SIESTA code are tightly coupled, the idea behind the CP2K interface is to reuse the original SMEAGOL code as much as possible. To do so, the original SMEAGOL-related source files became a part of a standalone SMEAGOL library (`libsmeagol.a`) linkable with alternative Quantum Chemistry software packages. SIESTA-specific parts of the original SMEAGOL Fortran source code were wrapped with preprocessor directives

```
#ifndef NOSIESTA
```

that enables us to compile SMEAGOL core as a library,

Instead of implementing CP2K-specific version of the original SIESTA-specific routines of the interface from scratch (which is about 6,000 lines of Fortran code in length) and keep it up to date with ongoing development of SMEAGOL code, we decided to reuse k-point-aware part of the original SIESTA/SMEAGOL interface and to convert CP2K data types into their corresponding SIESTA equivalents instead. The original interface code forms parts of the standalone SMEAGOL library as well.

2 Building CP2K/SMEAGOL interface

Before compiling CP2K with SMEAGOL support, please build the standalone SMEAGOL library first by running `make` program in its root directory.

The supplied version of `Makefile` is gfortran-specific. In case of other compilers, please amend `Makefile` accordingly. The following preprocessor flags are likely needs to be defined as part of the Makefile's `FCFLAGS` variable:

- `-DNOSIESTA` – disables SIESTA-specific parts (such as reading fdf input files) of the original SMEAGOL code;
- `-DMPI` – enables distributed-memory parallel version of SMEAGOL code. Must be enabled in case of distributed-memory parallel versions of CP2K, and disabled otherwise.

There is no need to run `make install`. Upon successful compilation, the standalone library `libsmeagol.a` can be found in `lib` directory created within the library's root directory. The directory `obj` contains individual object files along with corresponding Fortran modules. Please do not remove this directory, as these Fortran module files are required to build the CP2K/SMEAGOL interface.

To build CP2K with SMEAGOL support, the following changes need to be applied to CP2K's arch file:

- Append `-D__SMEAGOL` flag to `DFLAGS` variable;
- Append `-I"/path/to/libsmeagol/obj"` to `FCFLAGS` variable;
- Append `-I"/path/to/libsmeagol/lib"` to `LDFLAGS` variable;
- Prepend `-lsmeagol` to `LIBS` variable. The order of the libraries is important. Generally speaking, the SMEAGOL library should be linked with CP2K prior linking it with a LAPACK library.

3 SMEAGOL-related input keywords

A separate file `keywords.txt` lists all keywords of the CP2K input section `&FORCE_EVAL / &DFT / &SMEAGOL` alongside relevant components of `smeagol_control_type` datatype. Parameters with `global` scope are passed to SMEAGOL via setting identically-named global variables in SMEAGOL's `negfmod` module. These variables are then processed by SMEAGOL core. In contrast, the variables with `local` scope are processed by the interface itself. For instance, they can be passed to SMEAGOL via subroutines' arguments, or used to allocate an array defined in global scope of an appropriate size. At the moment, CP2K accepts all input keywords found in the original SIESTA/SMEAGOL interface, even if the corresponding variables are newer accessed.

4 SIESTA format of sparse matrices

Sparse matrices in SIESTA 1.3f1 are stored in a block-distributed-row compressed-sparse-column format. The blocking factor is defined at compile time via the global parameter `BlockSize` in SIESTA's source file `parallel.f` which is usually equal to 8. There are a number of subroutines to convert between global and local row indices, as well as to query the MPI rank of a parallel process local to a particular global row index. In contrast with ScaLAPACK matrices which use block-cyclic distribution scheme, all non-zero matrix elements on any single row are stored on one MPI process. In fact, SMEAGOL does not rely on a ScaLAPACK library at all.

Non-zero matrix elements local to the particular MPI process are stored in a compressed one-dimensional array. The following variables are used map local index of each non-zero matrix element with its global (row,column) indices:

SIESTA variable	**CP2K variable**	**description**
`nuotot`		[scalar] total number of rows across all MPI processes
`nuo`	`nrows`	[scalar] number of rows on the given MPI process
`no`	`ncols`	[scalar] number of columns, `nuotot` * `nimages`
`numh`	`n_nonzero_cols`	[array(`nuo`)] number of non-zero matrix elements for each locally stored row
`listhptr`	`row_offset`	[array(`nuo`)] offset (index-1) of the first non-zero matrix element for each locally stored row
`maxnh`	`n_nonzero_elements`	[scalar] number of non-zero matrix elements local to the given MPI process
`listh`	`col_index`	[array(`maxnh`)] column indices for each local non-zero matrix elements
`xij`	`xij`	[array(`maxnh`)] interatomic distance for each local non-zero matrix elements; `xa(:,iaorb(icol))-xa(:,iaorb(irow))`
`indxuo`	`indxuo`	[array(`no`)] indices of equivalent atomic orbitals. Index of the atomic orbital within the primary unit cell which is equivalent to the given atomic orbital
`iaorb`	`iaorb`	[array(`no`)] index of the atom on which the orbital is centred
`xa`	`xa`	[array(3,`natoms` * `nimages`)] atomic positions

where `natoms` is the number of atoms in primary unit cell, and `nimages` is the number of cell images (replicas). Cell images are ordered according to canonical enumeration of integers $0, 1, -1, \ldots, n, -n, \ldots$ (https://oeis.org/A001057) in `x`,`y`,`z` order, where `x` is the fastest index.

The mentioned CP2K variables form part of the derived `siesta_distrib_csc_struct_type` datatype.