1 Compilation

1.1 NETLIB

Dependencies

- Fortran compiler (`ifort, gfortran, etc.`)

1.2 FFTW

Dependencies

- Fortran compiler (`ifort, gfortran, etc.`), for the main code
- C compiler (`icc, gcc, etc.`), for the FFTW libraries
- FFTW libraries

** For FFTW to make use of OpenMP, the compiler itself needs to be compiled with OpenMP support! This may sound trivial but e.g. this is not the case for the stock 'gcc'* on macOS!

** The compiler suite you use to compile QDD has to match the one you used to compile FFTW! E.g., if you used the Intel compiler suite ('icc') to compile FFTW, you MUST use the same suite ('ifort') to compile QDD!.* Actually this only happens in this 'direction'. When the FFTW libs are compiled with `gcc` and the QDD code with `ifort` everything works fine. Intel-specific problem?

1.3 MKL

Dependencies

- Fortran compiler (`ifort, gfortran, etc.`) for the main code
- C compiler (`icc, gcc, etc.`) for the FFTW wrapper library
- Intel MKL
- Compiled FFTW wrappers to Intel MKL (`$MKLROOT/interfaces/fftw3xf` for Fortran)

1.4 Preprocessor flags

Flags from old '`define.h`'

- `IVERSION` Allows to communicate a version number
- `gridfft` Switches to Fourier definition of derivatives. Excludes `findiff/numerov`.
- `findiff` Switches to 3-point finite differences for derivatives. Excludes `gridfft/numerov`.
- `numerov` Switches to 5-point finite differences for derivatives. Excludes `gridfft/findiff`.
- `coufou` Switches to Coulomb solver using FAlr. Excludes `coudoub`.
- `coudoub` Switches to exact Coulomb solver on doubled grid. Excludes `coufou`.
- `coudoub3D` Switches to 3D FFT in connection with exact Coulomb solver. Requires `coudoub=1`.
- `twostsic` Includes all subroutines for full SIC in compilation.
- `cmplxsic` Switches to full SIC using complex wavefunctions. Requires `twostsic=1`.
- `raregas` Includes all subroutines for polarizable environment (MgO, raregases) in compilation.

Flags from `*.F90`-files related to parallelisation (all are set in `makefile`)

- `dynopenmp` Deduced parameter, determined in `makefile`. Switches to OpenMP arrangement which associates CPU wavefunction-wise. Relevant in connection with `paropenmp=1`.
- `parano` Switches to compilation for purely sequential run. Excludes `parayes/paropenmp/...`.
- `parayes` Switches to compilation for MPI parallel run. Excludes `parano/paropenmp/...`.
- `paraworld` Switches to compilation for MPI parallel run. Similar to `parayes`, but with more communication statements (??). Excludes `parano/paropenmp/parayes/...`.
- `paropenmp` Switches to compilation for OpneMp parallel run. Excludes others as `parano/parayes/...`.
- `selpara` Obsolete and can be erased.
- `simpara` Obsolete and can be erased.

Miscellaneous flags from `*.F90`-files

- `COMPLEXSWITCH/REALSWITCH` These are internal switches which are automatically set in the `makefile`. They serve to compile independently a REAL and a COMPLEX version of some subroutines. The user cannot and shoult not touch them.
- `exonly` Obsolete and not actually used.
- `fftw_cpu` Deduced parameter, determined in `makefile`.

- `fftw_gpu` Obsolete and must be erased.
- `fftwnomkl` Deduced parameter, determined and used in `makefile`.
- `gunnar` Obsolete and not actually used.
- `hamdiag` Obsolete and not actually used.
- `lda_gpu` Obsolete and must be erased.
- `locsic` Obsolete and can be erased.
- `netlib_fft` Deduced parameter, determined and used in `makefile`.
- `oldkinprop` Obsolete and can be erased.
- `pw92` Obsolete and can be erased.
- `symmcond` Somewhat mysterious. Seems to be obsolete together with the file `symmcond_step.F90`. Check performance of full SIC.
- `tfindiff` Should be replaced by `findiff`.
- `tnumerov` Should be replaced by `numerov` where 'numerov' means derivates with 5-point precision. But we should discuss whether we want to maintain this option.