Sign Function Usage

X

October 2023

In this document, we describe the compilation flags (in `Makefile`) and the input parameters (in `sample.ini`) that can be changed in order to have different working "versions" for the computation of the sign function in Lattice QCD implemented in this code.

1 Compiling and running

But, before describing flags/params, let us just mention how to compile and run this code:

- compile : `make -j N`
- run : `.  run -i sample.ini` (note the dot at the beginnin)

2 `Makefile`

- `-DORTH_MGS` : as part of `OPT_FLAGS`, this flag can be changed to `-DORTH_CGS`, allowing the switch from Modified Gram-Schmidt to Classical Gram-Schmidt in the orthogonalizations in the Arnoldi relation

3 `sample.ini`

- `iterations between restarts` : this is the length of the Arnoldi used in the computation of the sign function

4 TO-DOs

We list here now some TO-DOs for the code:

- ~~enable choosing betwen MGS and CGS~~

- modify the code to apply the operator in Eq. (5) in the document `sign_function_notes.pdf`. Right now, the polynomial preconditioner (i.e. π) just takes the form of the identity

- ~~implement/call $H_m^{-1/2}$, via SLEPc~~

- include the SLEPc code for $H_m^{-1/2}$ into DD-αAMG

- ~~implement a check for $\mathrm{sgn}^2(A) = I$~~

- re-integrate code for polynomial preconditioner, and make sure that Eq. (5) in the document `sign_function_notes.pdf` is being correctly implemented. Note that, in single precision, the Dirac operator in DD-αAMG experiences a re-ordering of the lattice w.r.t. the double precision one, but this should not affect the Leja values obtained. For the polynomial preconditioner, in general, incorporate the flag `-DPOLYPREC`, and for its construction, enable using either `-DPOLYPREC_CONSTR_DOUBLE` or `-DPOLYPREC_CONSTR_SINGLE`

- set appropriate values for μ and m_w, and include corresponding input parameters `sample.ini`:

  - `chem_pot_mu`
  - `overlap_op_shift`