https://github.com/mstsuite/MuST

mstsuite / MuST Public

Notifications | Fork 31 | Star 68

Code | Issues 9 | Pull requests 1 | Actions | Projects | Wiki | Security | Insights

master | 2 Branches | 198 Tags | Go to file | Code

wangy2012 Added a new input key: Initial Fermi Energy Setting. bdcfe84 · 2 months ago | 709 Commits

Benchmark/CoCrFeMnNi	Updated the benchmark case for CoCrFeMnNi high entropy ...	9 months ago
KUBO	Fixed a minor issue that could cause git pull command to fail.	last year
MST	Added a new input key: Initial Fermi Energy Setting.	2 months ago
Potentials	Added the starting potential for Y and Pb	5 months ago
Tutorials	Fixed a minor issue when performing the DOS calculation.	8 months ago
architecture	Added two architectures files for vista@TACC.	9 months ago
ase_must	Added keywords for newly added parameters in the ASE inte...	5 years ago
cmake	fix small issues with cmake	3 years ago
docs	Update installation.rst	3 years ago
external	Added an architecture file for using the nvhpc compiler	2 years ago
lsms	Commented out an unsupported subroutine call to flush	4 years ago
scripts	A minor update	10 months ago
.gitignore	fix small issues with cmake	3 years ago
CMakeLists.txt	fix small issues with cmake	3 years ago
Dockerfile	fix typo in docker	2 years ago
LICENSE	Initial commit	6 years ago
Makefile	Fixed an issue in Makefile.	2 years ago
MuST_logo.png	Replaced logo file	3 years ago
README.md	modified: README.md	last year
Wiki Images	Create Wiki Images	2 years ago

README | BSD-3-Clause license

About

Multiple Scattering Theory code for first principles calculations

Readme
BSD-3-Clause license
Activity
Custom properties
68 stars
9 watching
31 forks
Report repository

Releases 16

v1.9.4 Latest
on Oct 11, 2024

+ 15 releases

Packages

No packages published

Contributors 11

Languages

Fortran 59.8% | Roff 20.4% | C++ 8.4% | Cuda 5.2% | C 4.4% | Makefile 0.5% | Other 1.3%

Files

master

Go to file

- Benchmark
- KUBO
- MST
  - Accelerator
  - arch
  - driver
  - external/libxc-5.1.6
  - iolib
  - lib
  - plib
  - sample
  - slulib
  - src
  - util
  - CMakeLists.txt
  - Makefile
  - README
- Potentials
- Tutorials
- architecture
- ase_must
- cmake
- docs
- external
- lsms
- scripts
- .gitignore
- CMakeLists.txt
- Dockerfile
- LICENSE

MuST / MST /

wangy2012 Added a new input key: Initial Fermi Energy Setting. bdcfe84 · 2 months ago History

Name	Last commit message	Last commit date
..		
Accelerator	Changes to be committed:	last year
arch	Added a new feature to enable specifying physical parameters for a sp...	2 years ago
driver	Added a feature that writes mstrun.xml, which contains the force data...	10 months ago
external/libxc-5.1.6	fix cmake build	3 years ago
iolib	Create CMakeLists.txt	3 years ago
lib	Fixed a few minor issues.	last year
plib	Fixed an issue that causes compiling without MPI to fail.	3 years ago
sample	Added an input parameter for the DMFT calculations in the example for V	2 years ago
slulib	This new version allows to build and install MuST with different comp...	5 years ago
src	Added a new input key: Initial Fermi Energy Setting.	2 months ago
util	Added the starting potential for Y and Pb	5 months ago
CMakeLists.txt	fix small issues with cmake	3 years ago
Makefile	Added a feature that writes mstrun.xml, which contains the force data...	10 months ago
README	Update README	2 years ago

README

```
Steps to install MST:
=====================
1. ln -s arch/architecture_file architecture.h
2. make

Note --
make clean: delete *.o, *a, and *.mod files under bin/.
make distclean: delete bin/ directory and architecture.h.

Sub-directories under MST:
==========================
01. Accelerator:
Contains the interface routines for using the Accelerator written in C
```

Ab initio Approaches to Random Alloys using MuST

- Effective medium method with coherent potential approximation: KKR-CPA
- Supercell methods
  - KKR (with a unit cell ~ 100 atoms generated by SQS method) ~ O(N^3) operations
  - Locally Self-consistent Multiple Scattering (**LSMS**), which is linear scaling, ~ O(N) operations, and allows to perform *ab initio* calculations for very large unit cell containing $N >$ 100,000 atoms.
    - ✓ Intrinsically parallel over the atoms in the unit cell

Major Computational Bottleneck of the LSMS Method

KKR matrix inverse to obtain the multiple scattering matrix $\underline{\tau}$, which is used to calculate the Green function G, is performed for each atom in the unit cell and for each energy point along a complex energy contour

- The size of the KKR matrix = $((l_{\max}+1)^2 \cdot M) \times ((l_{\max}+1)^2 \cdot M)$, and usually $l_{\max} = 4$ and $M \sim 100$
- For a spin-canted calculation where the spin-up and spin-down states need to be calculated simultaneously, the KKR matrix size is 4 times larger

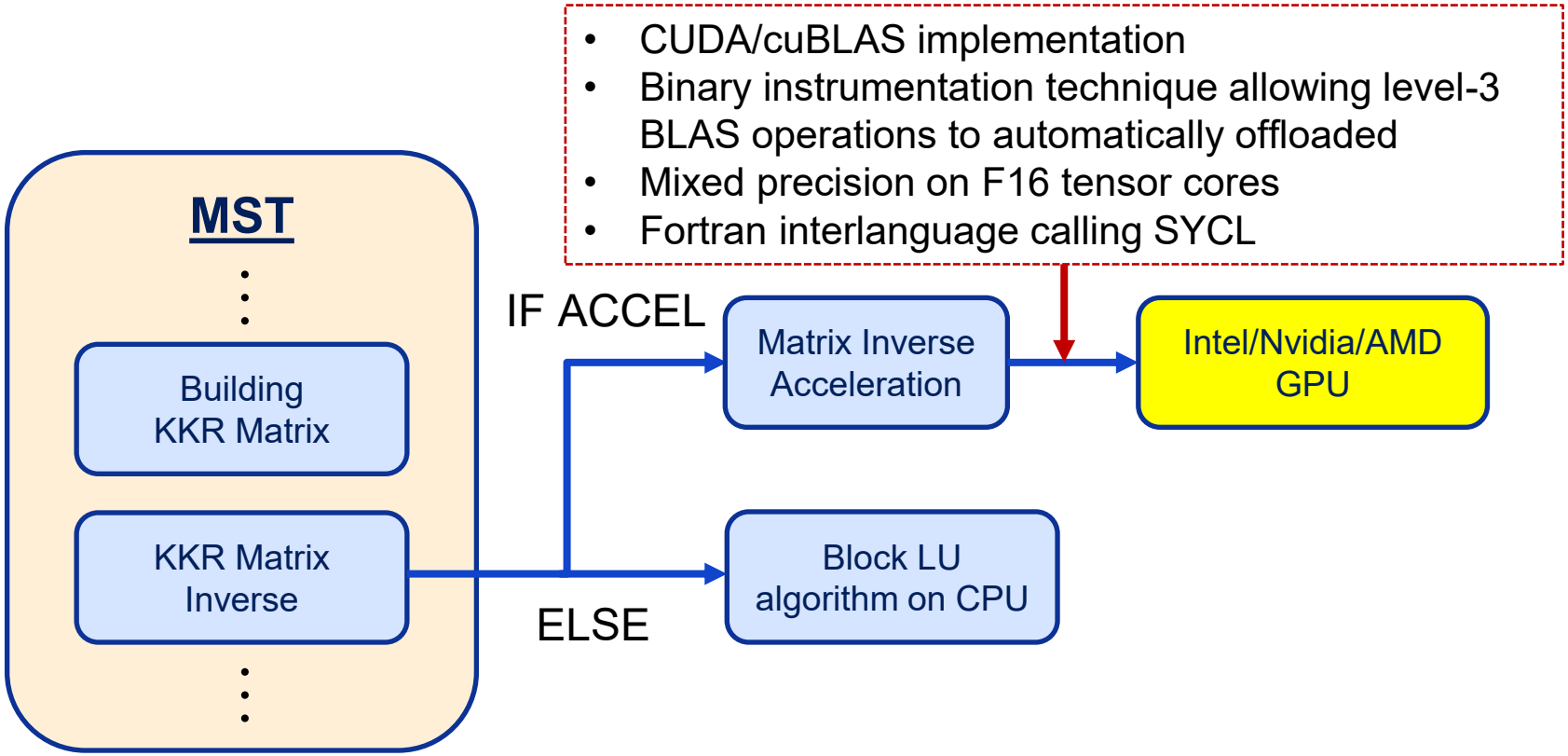
$$\underline{\tau}^{11}(\varepsilon) = \begin{bmatrix} \underline{t}_1^{-1}(\varepsilon) & \cdots & -\underline{g}_{1M}(\varepsilon) \\ \vdots & \ddots & \vdots \\ -\underline{g}_{M1}(\varepsilon) & \cdots & \underline{t}_M^{-1}(\varepsilon) \end{bmatrix}^{-1}_{11}$$

KKR Matrix (double complex)
~ 2500 × 2500 or more

In this KKR matrix expression, $\underline{t}_n$ is the single site scattering matrix of atom n, g_{nm} is the free-electron propagator matrix between n and m atomic sites.

Each matrix block size is 25 × 25

Acceleration of the LSMS Method



- We found a 4.0X - 7.3X speedup ratio for LSMS calculations can be achieved with GPU acceleration
- We allow multiple MPI processes offload the calculations to a single GPU, with each MPI process computes the matrix inverse independently

Acceleration of LSMS

MuST/MST/src/ClusterMatrixModule.F90::
calClusterMatrix

```
#ifdef ACCEL
      pBigMatrix => aliasArray2_c(BigMatrix, dsize, dsize)
      call invertMatrixLSMS_CUDA(my_atom, pBlockMatrix, kkrsz_ns,      &
                                 pBigMatrix, dsize )
#else
      call invertMatrixBlock( my_atom, pBlockMatrix, kkrsz_ns, kkrsz_ns,  &
                              BigMatrix, dsize, dsize )
#endif
```

Without GPU Acceleration

MuST/MST/src/MatrixBlockInversionModule.F90::
invertMatrixBlock

With GPU Acceleration

MuST/MST/Accelerator/invertMatrixLSMS_CUDA.F90

```
#ifdef CUDA
      …,…
      call cusolver_lsms_c(dsize,pBigMatrix,kkrsz_ns,tmpmatrixC)
      …,…
#else
!     The following line under this condition seems from an unfinished
!     work and needs to be corrected.
!     However, under the current setting , this condition is not reached
      call zblock_lu_CPU(a,lda,blk_sz,nblk,ipvt,mp,idcol,k)
#endif
```

MuST/MST/Accelerator/cusolver_LSMS_c.cu

```
extern "C"
void cusolver_lsms_c_(int *m, double _Complex *a,
int *block_size, double _Complex *b)
{
    …,…
}
```