

TBKOSTER

TIGHT-BINDING KERNEL OF OBJETS FOR SPINTRONICS ENHANCED RESEARCH

C. Barreteau, P. Thibaudeau*

Release 0.0.4
March 18, 2025

Abstract

Contents

1	Background theory	1
2	Installation	3
2.1	Linux	3
2.2	MacOS	4
2.3	MS Windows	5
3	Running the code	7

*Supported by CEA

4	Getting Started	9
5	Input File Command Reference	11

Preface

CHAPTER
1

Background theory

Installation

The first step is to download the latest release of TBKOSTER from its github repository. To proceed, you have to check that git is installed :

```
$ locate git
```

Then, clone the distant repository to your local one :

```
$ git clone https://github.com/spindynamics/TBKOSTER.git
```

In order to build this documentation, a LaTeX distribution is mandatory, including some external packages such as pdflatex, bibtex and htlatex.

2.1 Linux

Preferred method : on Ubuntu 22.04 and later, install gfortran and cmake to compile TBKOSTER. Be sure to get the mandatory related dependencies of these packages.

```
$ sudo apt install cmake gfortran clang doxygen graphviz \
libblas-dev liblapack-dev libomp-dev texlive-latex-base \
texlive-latex-extra tex4ht
```

Be sure to update to cmake release 3.9 or higher. In order to build the code, to the root of TBKOSTER directory :

```
$ cmake -B build
$ cmake --build build
```

To get access to the OpenMP implementation, then update your packages with openmp support.

If you want to use Intel compiler, MKL and Intel Lapack libraries, you have to tell this to cmake as, in sequential :

```
$ BLA_VENDOR=Intel10_64lp_seq FC=ifort cmake -B build
```

Be sure that the variable MKLROOT is set accordingly.

In order to get good numerical performance, you have to produce a Release version as :

```
$ cmake -B build -DCMAKE_BUILD_TYPE=Release
```

You can combine all these options. If you prefer to prepare an installation with a given installed Lapack library and gfortran try :

```
$ BLA_VENDOR=OpenBLAS FC=gfortran cmake -B build
```

You can also use an optional Fortran-lang stdlib package. Set the option ON in the CMakeLists.txt file at the top of the root directory to enable it. Then type

```
$ git clone https://github.com/fortran-lang/stdlib
$ cd stdlib
$ cmake -B build -DCMAKE_MAXIMUM_RANK=4 \
  -DCMAKE_INSTALL_PREFIX=$HOME/.local
$ cmake --build build
$ cmake --install build
```

2.2 MacOS

Preferred method : on MacOS 10.11 and later, install the cmake, lapack and gfortran with llvm support, with the ports subsystem (<http://www.macports.org>)

```
$ sudo port install cmake gcc6 libgcc6 gcc_select \
  llvm-3.9 llvm_select lapack libomp
```

In order to build the code, to the root of TBKOSTER directory :

```
$ cmake -B build
$ cmake --build build
```


For both Linux and MacOS platform, you can invoke cmake with Release or Debug option in order to deploy these releases. Simply try

```
$ cmake -B build -DCMAKE_BUILD_TYPE=Release/Debug
$ cmake --build build
```

To get access to the OpenMP implementation, then update your ports with openmp package. You can easily change your settings with

```
$ port select --summary
$ port select --set llvm mp-llvm-3.9
$ port select --set gcc mp-gcc6
```

In order to get good numerical performance, you may use the OpenBLAS library and produce a Release version as :

```
$ port install openblas
$ BLA_VENDOR=OpenBLAS cmake -DCMAKE_BUILD_TYPE=Release -B build
```

2.3 MS Windows

For Windows earlier than 10 release 1709, you have to consider the following method: download and follow the instructions to install MSYS2 software distro and building platform (<http://www.msys2.org/>). Open an MSYS console and first upgrade the whole system :

```
$ pacman -Syu
```

get the compilation toolchain :

```
$ pacman -S mingw-w64-x86_64-toolchain
```

get the gfortran compiler and lapack library :

```
$ pacman -S mingw-w64-x86_64-gcc-libgfortran
$ pacman -S mingw-w64-x86_64-openblas
```

get the cmake program to control the software compilation process using simple platform and compiler independent configuration files, and to generate native makefiles and workspaces that can be used in the compiler environment of your choice :

```
$ pacman -S cmake
```

In order to build the code, open a MSYS MINGW 64-bit console. To the root of TBKOSTER directory :

```
$ if ! test -d win; then mkdir win; fi  
$ cd win  
$ cmake -G"MinGW_Makefiles" -DCMAKE_SH="CMAKE_SH-NOTFOUND" ..  
$ mingw32-make
```

In order to run TBKOSTER.exe, be sure you have the TERM and TERMINFO environment variables up to date into your .bashrc file :

```
$ export TERM=xterm  
$ export TERMINFO=/c/Program Files/msys/mingw64/share/terminfo
```

No OpenMP implementation for MS Windows has been tested.

For Windows higher than 10 release 1709, you simply have to activate the optional Windows SubSystem for Linux, download the Ubuntu Package from the Microsoft Marketplace and follow the instructions of the Linux section of this manual.

CHAPTER
3

Running the code

CHAPTER
4

Getting Started

CHAPTER
5

Input File Command Reference

Bibliography

