MUPRO SDK

Release v0.0.1

Xiaoxing Cheng

Jun 15, 2023

CONTENTS

CHAPTER

ONE

LICENSE

We sell the SDK license based on your end application type. For example, we have ferroelectric license, ferromagnetic license, metal license, and effective properties license, each give you access to the necessary solvers for the specific type of simulation.

For each of the license type, we provide an open sourced main program that our licensed users can access and modify for their own use cases.

On purchase of the Mu-PRO SDK you are allowed

- to use the software for you or your business's internal usage

You are not allowed

- share your license information with public or other users that does not appear on the license agreement
- to include the Mu-PRO SDK in any commercial product
- to sell any software that linked to Mu-PRO SDK library

Warning: Redistribution of the Mu-PRO SDK is forbidden unless special agreement is signed.

Disclaimer

Our program does not contain or use any functions from unlicensed commercial codes or non-permissive open sourced codes. The distributed library only uses external subroutines from intel oneMKL which is licensed under the Intel Simplified Software License.

Mu-PRO Commercial Software License

This Mu-PRO Commercial Software License (the "License") is hereby granted to the person or entity ("Customer") who has purchased a license to use the software product ("Software") provided by Mu-PRO LLC.

1. Grant of License

Subject to the terms and conditions of this License, Mu-PRO LLC grants to Customer a non-exclusive, non-transferable, and non-sublicensable license to use, modify, and compile the Software solely for Customer's internal business purposes.

2. Restrictions

Customer may not: a. distribute, sublicense, or transfer the Software, in whole or in part, in source code or compiled form, to any third party or non-customer; b. use the Software for any purpose other than Customer's internal business

purposes; c. reverse engineer, disassemble, or decompile the Software, except as expressly permitted by applicable law, without the prior written consent of Mu-PRO LLC.

3. Ownership

All rights, title, and interest in and to the Software, including all intellectual property rights therein, are and shall remain the exclusive property of Mu-PRO LLC. This License does not convey any ownership rights in the Software to Customer, but only a limited right to use the Software as expressly provided in this License.

4. Termination

This License is effective until terminated. Mu-PRO LLC may terminate this License immediately upon written notice to Customer if Customer breaches any term of this License. Upon termination of this License, Customer shall cease all use of the Software and destroy all copies, modifications, and merged portions in any form.

5. Disclaimer of Warranty

The Software is provided "AS IS" and without warranty of any kind, express or implied, including, but not limited to, warranties of merchantability, fitness for a particular purpose, and non-infringement.

6. Limitation of Liability

In no event shall Mu-PRO LLC be liable for any damages, including, without limitation, direct, indirect, incidental, special, consequential, or punitive damages, arising out of or in connection with the use of or inability to use the Software, even if Mu-PRO LLC has been advised of the possibility of such damages.

7. Miscellaneous

This License shall be governed by and construed in accordance with the laws of Pennsylvania, without regard to its conflict of law provisions. Any disputes arising under or in connection with this License shall be subject to the exclusive jurisdiction of the courts located in State College, Pennsylvania.

This License constitutes the entire agreement between the parties concerning the subject matter hereof and supersedes all prior and contemporaneous agreements, understandings, negotiations, and discussions, whether oral or written, between the parties relating thereto.

If any provision of this License is held to be unenforceable or invalid, the remaining provisions shall remain in full force and effect.

CHAPTER

TWO

OVERVIEW

In 2020, Mu-PRO released several commercial simulation programs, including two phase-field models, the Ferroelectric module and the Ferromagnetic module, and an Effective Properties calculation program. Inside Mu-PRO, we have a shared library that powered all these seemingly separate programs.

Since then, we have been planning to release the core library itself and allow our users to modify the main programs so that they can get the maximum value from our solvers. This is also one of our most frequently heard requests that "Can we change some simple things in the main program?".

In 2023, we have our answer to this, the Mu-PRO Phase-Field SDK.

The SDK aims to provide phase-field model developers a:

- distributed parallel simulation framework
- multiple solvers based on fast fourier transform
  - elastic equilibrium
  - poisson equation
  - Landau-Lifshitz-Gilbert equation
  - Time-dependent Ginzburg Landau equation
  - Allen-Cahn equation
  - Cahn-Hilliard equation
- free format parameter input
- uniform data input and output

CHAPTER

THREE

USER GUIDE

In this guide we will guide you through how to start using our SDK.

3.1 Get started

3.1.1 Install

1. Download the archive file from our website, usually it has a name like this `wget muproPFSDK-0.0.2-Linux.tar.xz`
2. Depress the archive file. `tar -xJf muproPFSDK-0.0.2-Linux.tar.xz`
3. Move the content to location you want the SDK to be installed, such as your home directory `mv muproPFSDK-0.0.2-Linux/opt/mupro/phasefieldsdk ${HOME}/mupro`

3.1.2 Build the main program

We are using cmake to help us create the build system and compile our program. You should check the documentation of each main program. In general, the procedure is like this

1. Create your own CMakeUserPresets.json, `cp CMakePresets.json CMakeUserPresets.json`
2. Update the mupro_ROOT cache variable in your CMakeUserPresets.json
3. Configure and build the project

3.1.3 Obtain the license

1. When you first run the main program, or more specifically call any of the setup subroutines from SDK, the program will detect that no license file is available and it will collect your local machine information, then create a client file which you need to send to MuPRO at mesoscale-modeling@mupro.co
2. Next, we will respond you with a license.lic file that you should put to designated location that the main program can read in the next execution.
3. For a distributed machine, your calculation node needs to have access to the same file system as your login node. And, you need to run the program once on your login node (or whichever node you obtain the client file), because the first run will check the license.lic file and if valid will write the verification result in your home directory, your future run will rely on this file.

3.1.4 Development environment

Operating System

We recommend using a linux computer with intel processors for your main program development using the Mu-PRO PhaseFieldSDK, since we rely on the intel oneAPI to develop the program.

You can check here for more information on the hardware and system requirement.

Linux

The SDK is developed on an linux server with Ubuntu 22.04.1 LTS.

Mac

Only Mac computer with intel processors are supported. Mac with M1 chip is not officially supported, you have to test it by yourself.

Windows

The SDK is tested on Windows 10 PRO with Visual Studio Community Edition 2019.

External dependencies

You need to have three softwares installed:

1. intel oneapi basekit
2. intel oneapi hpckit
3. cmake 3.20 and above

Parallelisim

We have implemented distributed parallelization with MPI, while no shared-memory parallelization in used. We use the intel mpi provided by intel oneapi hpckit.

Compiler

We use the intel compilers that is part of the intel oneapi basekit.

Code editor

We recommend using Visual Studio Code as the editor with **Modern Fortran** extension and **fortls** installed.

Programming language

The Mu-PRO PhaseFieldSDK is developed mainly using Fortran, it is highly recommended you also using Fortran for your main program. Theoretically, you can use C or C++ to interoperate with Fortran, but we have not perform such tests, so you have to use C/C++ at your own risk.

3.2 Guide for absolute beginner

In this section we will treat you as absolute beginner for code development and linux, and walk you through all of the necessary steps to start using our SDK.

3.2.1 Preparation

3.2.2 Usage

3.2.3 Process data

3.3 Creating a main program

The general procedure of creating a main program based on the SDK is as follows:

1. Read necessary parameters from the input files. You may also hard coded all parameters in the program with reading from external input files.
2. Normalize the parameters. This is purely for numerical benefits, as avoiding multiplication of very large and very small value can improve the solver accuracy.
3. Simulation system setup using the normalized parameters.
4. Start the main iteration loop.
5. Finalize the whole program

3.3.1 Read input

We recommend using the toml file for input parameters. You can call the `mupro_toml_read_file` subroutine and `get_value` subroutine to obtain desired parameter from.

3.3.2 Using the SDK

Some of the subroutines from the SDK needs to be called directly, while others needs to be called as procedure of a derived type

Direct called subroutines

These are usually either global functions that can only needs to be called once or input file reader which is a thin wrapper around the procedures provided by the toml-f library. Here are a list of them:

- mupro_size_setup
- mupro_fft_setup
- mupro_toml_read_file
- get_value
- mupro_toml_evaluate_value

Call subroutines from a derived type

For most subroutines from the SDK, they are attached to some derived type, this is because most subroutines needs to be called within specific data context and those necessary data are kept in the derived type. A few examples of these subroutines include:

- setup from the type_mupro_electricContext
- solve from the type_mupro_electricContext

There are many more, and you should read the modules section of this manual to learn available solvers that you can use for your program.

CHAPTER

FOUR

EXAMPLES

CHAPTER

FIVE

SDK MODULES

All *module* provided by the SDK is prefixed with **mod_mupro**. All *type* provided by the SDK is prefixed with **type_mupro**

Most of the modules are quite independent, they do not rely on each other to be built and there are no hidden connections between them, except for two modules, the **size** module and the **fft** module.

CHAPTER

SIX

REFERENCES