

QUANTAPLUS Manual and User Guide

Muhammad Hussein Muhammadahmed Ahmed

*Department of Physics, Faculty of Sciences,
Sudan University Of Science & Technology Khartoum, Sudan*

Mohammed Maher Abdelrahim Mohammed

*Department of Physics, Faculty of Sciences,
Sudan University Of Science & Technology Khartoum, Sudan*

September 7, 2022

Abstract

QUANTAPLUS is a modern general-purpose multi-threaded quantum computing mechanics written in C++17 and composed solely of header files.

Contents

Contents

1



Introduction

High-performance numerical algorithms play a central role in theoretical calculations and analysis of experimental data in (non-relativistic) quantum mechanical problems, which is the problem of predicting the properties of systems of several quantum particles at a much larger scale [?]. The latter is a crucial ingredient in the field of today's computational physics due to its technological interest and potential applications in various problems of fundamental importance in quantum chemistry, condensed matter physics, and materials science.

Files, installation and testroutines

The package can be downloaded from github repository.

Files

The gzipped tarred file (QuantaPlus.vvvv.tar.gz) will produce a directory QuantaPlus.vvvv with a number of subdirectories. vvvv is version information. The created directory is called the main directory in the remainder. The main directory contains the files LICENSE, README.md, and a Makefile.

The subdirectory doc contains the documentation. The subdirectory lib will after compiling contain the compiled libraries libquantaplu.a.

The subdirectory include contains all the needed header files. examples contains the testing and example programs. outputs contains the output obtained after execute some of the example files. There are a few extra files around as well. These typically contain inputs needed or large sets of constants.

Installation

The main steps are to run make in the main directory. This should produce the files libquantaplu.a and also send a copy to the lib subdirectory. You might have to change the variables CC, CFLAGS and CFLAGTESTS. CC should specify the C++ compiler and the options to be used for everything. CFLAGS can be used to specify additional options in compiling the libraries and CFLAGTESTS to specify additional options for the testing programs. "make clean" can be used to remove many of the files created during compiling. The actual installation is by putting the contents of the include directory somewhere in the include path of your compiler and the two files libjbnlib.a and libchiron.a somewhere in the library path. For many C++ compilers the paths are given in the environment variables CPLUS_INCLUDE_PATH and LIBRARY_PATH respectively.

testroutines

For every file `xxx.h` included for QuantaPlus there is a testing/example code `testxxx.cpp` in the subdirectory `test`. These can be compiled using “`make testxxx`” in the main directory. Executing the resulting file `QUANTA.out` should then produce output identical (up to the precision specified and possible randomly generated cases) to the file `testxxx.dat` in the subdirectory `testoutputs`.