Wavepacket Propagation Program (w/ Dr Nadav Avidor)

Lorenzo Basso, Jack Lee, Matthew Zhang, Feiyang Chen

April 2020

Contents

1 Wavepacket Propagation Introduction

1.1 About

The Wavepacket propagation project is an extension of a Part III project by Ocean Haghighi-Daly. It is a 3D working version of a Working attempt at a Split-Operator method of wavepacket propagation.

1.2 Prerequisites

To begin with, a computer with a CUDA compatible GPU (List of compatible GPUs) is required.The MEX/CUDA dependent files have already been compiled for an x64 Linux and x64 Windows system with CUDA 10.1 compatibility. However, it is potentially necessary to install a CUDA/C++ compiler depending on the CUDA capability your GPU has.

Installation Guides:

- Latest CUDA Package Download (If you believe you don't need to follow the installation guides as the Windows install is reasonably simple)
- Windows Installation Guide
- Linux Installation Guide
- Mac OS X Installation Guide

These guides also include the relevant instructions to install CUDA/C++ compilers for your relevant OS. If for some reason the documentation is too tedious, we will (maybe) try to provide a simplified version that produces functionality in the Appendix

The required Matlab Toolboxes for the program to function are as follows:

- Parallel Computing Toolbox
- Bioinformatics Toolbox

2 Usage

2.1 Custom Paths and Potentials

Functionality to include custom adsorbate paths and potentials was added by Jack Lee.

1. Custom potential: this allows you to specify a custom potential profile in the Z direction. To apply this, set decaytype to 4 and specify a .txt file (in beta4_2) as a string in potfile. This should contain several numbers separated by spaces (or new lines) which are the values for the potential in Joules in each cell of the simulation out from the corrugation function (+zoffset), i.e. the values are separated by a distance of lz/nz. The potential is zero for all

z not specified by the file. Zoffset should have a negative value, and can be used to extend the potential back from the gaussians to the surface, as if zoffset is 0 the area between the surface and the gaussian peaks will have 0 potential. The potential specified should be very high near the surface to prevent the wavepacket from getting past it, because the propagation algorithm leads to cyclic boundary conditions so any psi that reaches the end will appear at the other side, which is non-physical.

2. Custom paths: this allows you to specify the paths that adsorbates take, rather than having them be generated randomly. To enable this, set custompaths to true and specify in pathfile a text file (in beta4_2) formatted as follows: the first entry on each line should be a time, then for each adsorbate there should be its x position and then its y position at that time, separated by spaces. There can be any number of times, as long as the first is $\leq$ tStart and the last is $\geq$ tFinish, and they'll be interpolated to get the paths.

3. Saving paths: this allows you to save the randomly generated paths adsorbates take this time in the simulation. It doesn't do anything if custom paths is on. To enable it, set savingBrownianPaths to true and name a .txt file in Browniefile, which will be overwritten or created in beta4_2. This saves the paths in the same format as Custom paths reads them, with one line for each timestep of the simulation. These files can later be read by custom paths to reproduce this simulation (and could be used to vary the potential, detail etc.), although there is a small error introduced here so that the final psi from the custom paths simulation is slightly different to that of the original. Two custom paths simulations from the same source will be the same, however.

3 Appendix

3.1 CUDA + Compiler Install

3.1.1 Windows

Installation for Windows is reasonably straightfoward.