Solving the 1-Dimensional Schrödinger Equation numerically on a Grid

Scientific Computing and Programming, Vrije Universiteit Amsterdam, 2022

This project is part of the Scientific Computing and Programming course at the VU in Amsterdam.

External sources for this project are the LAPACK and BLAS libraries (imported via the MakeFile) and the diagonalization module which was provided in an earlier course exercise.

Input

The input.f90 file is used to define parameters for several other files. The first parameters are used for setting up the grid (number of grid points and length). Also the boundary conditions are set for the shooting method, which is usually approaching 0 on both sides.

Then the type of potential should be specified by a string: 'InfiniteBox', 'FiniteBox', 'GaussianWell', or 'GaussianBox'. Each type uses different parameters of the input file. Walls are set at a fraction of the total grid length. So if the grid length is 4 (around 0), the wall will be at -1 and 1 when the PotentialLengthofTotal is 0.5.

- Do not comment out the unused potential parameters. -

Grid setup

The method starts with defining a grid. The operations of the grid setup are located in GridSetup.f90. The subroutine uses the amount of grid points, the lower bound (derived from Length in the input file), and the upper bound to make a grid that has a uniform distance between grid points.

Potential

The potential module is used throughout the program. Four types of potentials are possible in the potential module: 'InfiniteBox', 'FiniteBox', 'GaussianWell', and 'GaussianBox'. This module is connected to the input file in which the potential type and parameters are specified.

Three-point matrices

A matrix is created in this module which depends on the grid and potential. The resulting matrix is provided to the diagonalization module to calculate eigenvalues of the matrix. These eigenvalues will be used as approximate eigenvalues for the shooting method.

Diagonalization is the most demanding operation of the program and the speed is determined by the amount of grid points (size of the matrix).

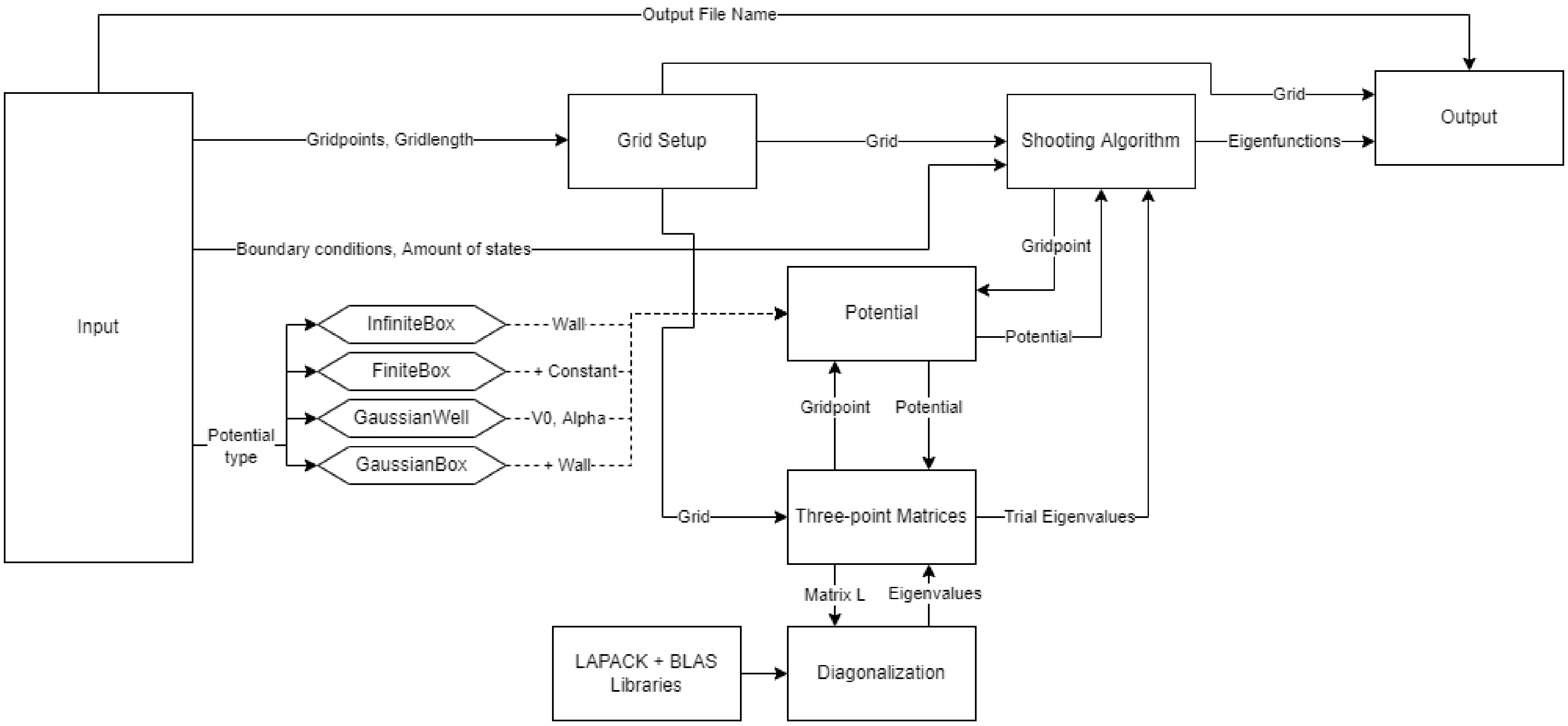
Shooting method

With the trial eigenvalues, boundaries of the eigenfunctions, and the grid it is possible to optimize the eigenvalues and ultimately perform the one-dimensional shooting method.
To start, an array is created with boundaries provided by an argument. This is the array that is used to perform the shooting method on. Then the shooting method is performed from left to right and from right to left until a matching point in the middle is reached. Both sides are normalized and then used to calculate a correction for the trial eigenvalue. If the correction falls under a threshold, the eigenvalue is accepted and the final shooting method is performed. The final shooting method happens solely from left to right.

Output

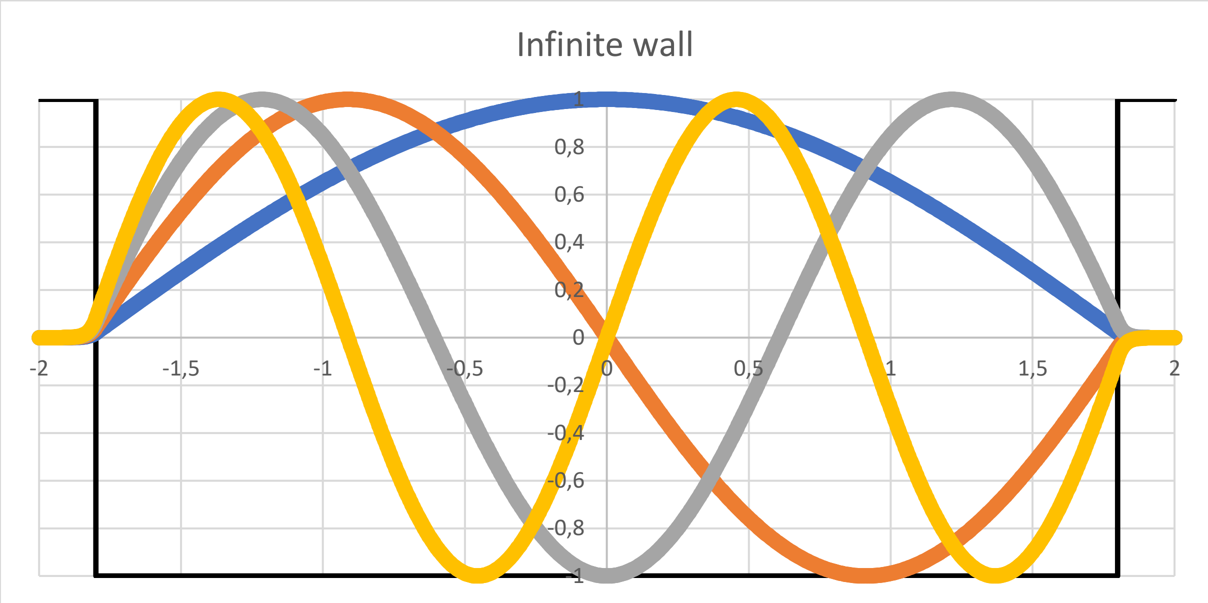
The shooting method is performed for a fixed amount of eigenvalues and can be specified in the input file. The result is stored in an output file of the name specified in the input file. This is a space separated file with the x-values in the first column and corresponding eigenfunctions in the next columns.

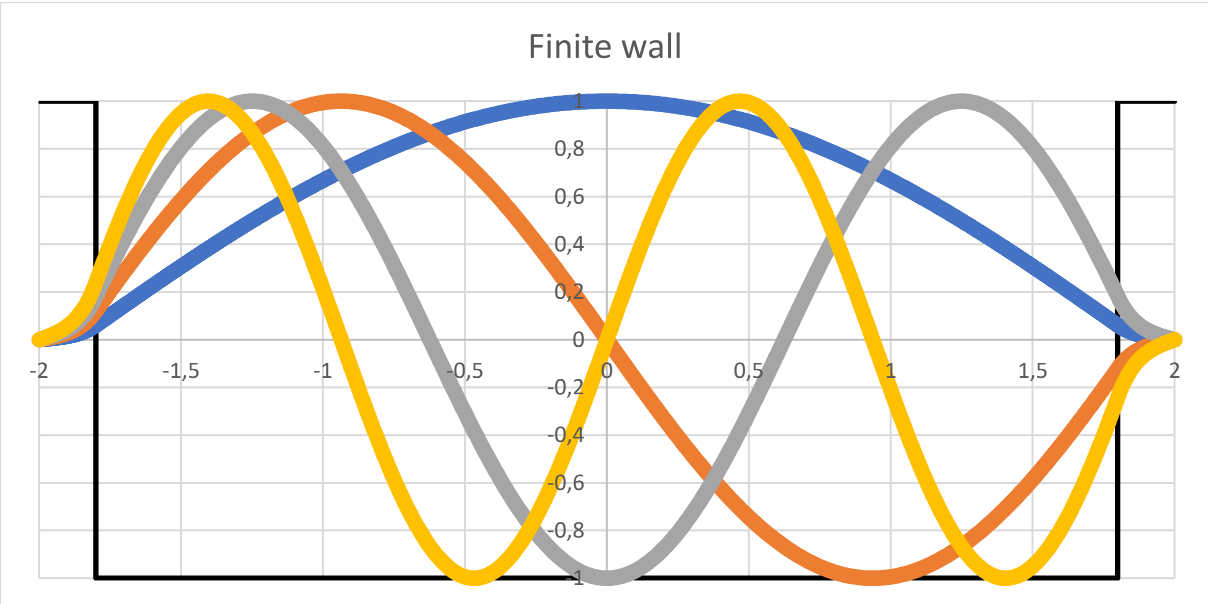
Flowchart

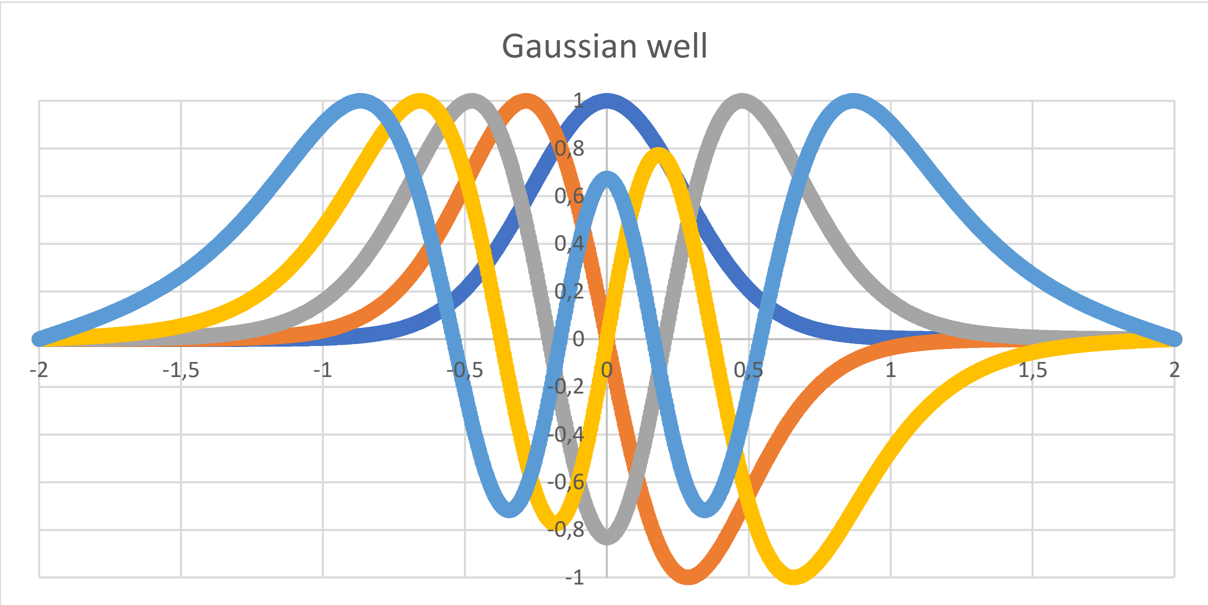


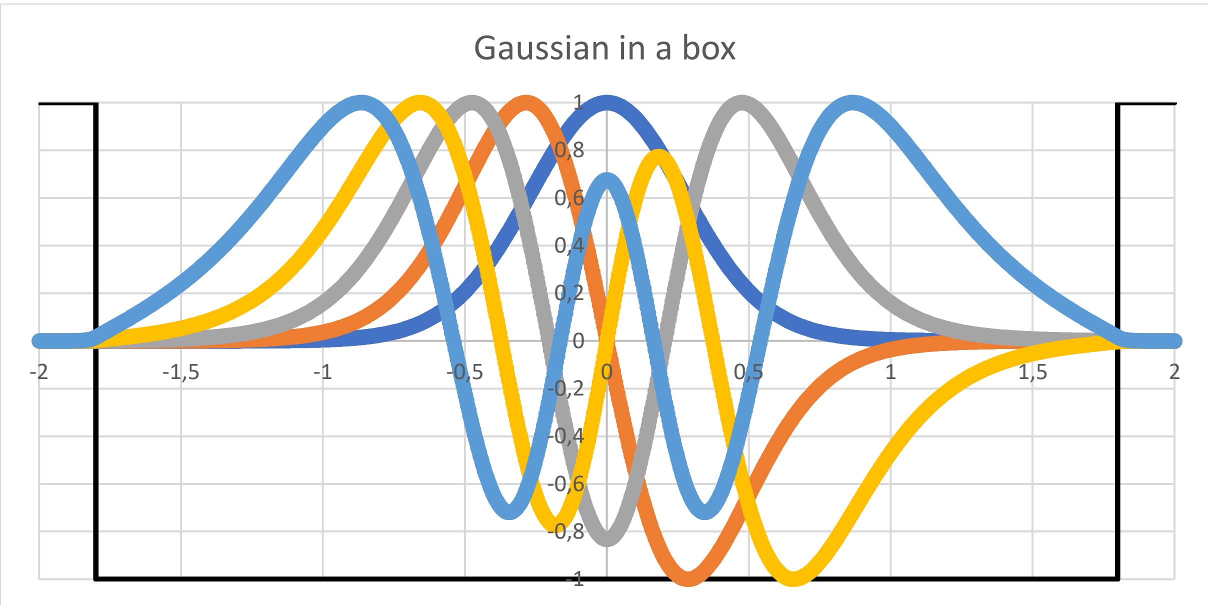
Output File Name
Input
Gridpoints, Gridlength
Grid Setup
Grid
Shooting Algorithm
Grid
Eigenfunctions
Output
Boundary conditions, Amount of states
Gridpoint
Potential
Potential
InfiniteBox
Wall
FiniteBox
+ Constant
GaussianWell
V0, Alpha
GaussianBox
+ Wall
Potential type
Gridpoint
Potential
Grid
Three-point Matrices
Trial Eigenvalues
Matrix L
Eigenvalues
LAPACK + BLAS Libraries
Diagonalization

Plotted output examples of the four potential types






Gaussian well
1
0,8
0,6
0,4
0,2
0
-0,2
-0,4
-0,6
-0,8
-1
-2
-1,5
-1
-0,5
0
0,5
1
1,5
2


Gaussian in a box
1
0,8
0,6
0,4
0,2
0
-0,2
-0,4
-0,6
-0,8
-1
-2
-1,5
-1
-0,5
0
0,5
1
1,5
2