

TDS Tools

Time Dependent Schroedinger equation
simulation Tool
User's Guide

Andrea Bertoni
Thomas Serafini

Istituto Nazionale Fisica della materia - CNR

1 Installation

Currently only the binary version is available. In the next releases of the software also the source code will be distributed.

As a prerequisite for the Linux version, you need to install on your system the OpenMotif library. It can be downloaded from <ftp://ftp.ics.com/openmotif/> For the Fedora distribution, we suggest to install the 2.3.2 version for Fedora 9 even if you are on a Fedora 10 system. After all the prerequisites are installed, just unpack the TDS Tools archive in a folder of your choice. The archive contains the executable file and some example namelists with simulation tests.

2 Software Description

The purpose of TDS Tool is to compute a numerical approximation of the solution of the time dependent Schroedinger equation:

$$i\hbar \frac{\partial}{\partial t} \psi(t) = H\psi(t) \quad (1)$$

$$\psi \in D \quad (2)$$

$$\psi = 0 \quad \text{on} \quad \partial D \quad (3)$$

$$\psi(0) = \psi_0 \quad (4)$$

$$t \in [0, T_f] \quad (5)$$

where the operator H is

$$H = -\frac{\hbar^2}{2m^*} \partial_x^2 + V(x)$$

In current TDS Tool version, 2D domains are supported, so $D = C([0, S_x] \times [0, S_y])$.

The space discretization is based on the box integration method, while for the time discretization the second order implicit Crank-Nicholson method has been employed. The solution of the inner linear system is entrusted to the Pardiso solver in the Intel MKL library.

TDS Tool is equipped with a very basic user interface which lets the user to input all the problem description data and the simulation parameters. In particular, input parameters can describe the following sections:

- **Grid:** describe the space discretization: it is cartesian and it can be both uniformly or non-uniformly spaced. The grid spacing can also be determined by an adaptive algorithm, which computes the grid based on the potential function or on the wave function.
- **Initial wave function:** the complex wave function at the initial time-step can be specified by the user
- **Potential:** describe the potential function on the discretization domain. It can be read from a user file or described with a scripting language.

- **Time simulation:** describe the time discretization parameters
- **Output:** different output formats can be specified. The output can also be downsampled in space and time for an easier visualization.

2.1 Grid

TDS Tool works on a wave function which is space-discretized on a grid. The grid is cartesian and can be non-uniform. It is defined as follows: let $N_x, N_y \in \mathbb{N}$ be the number of discretization points along the x and y axis. Let $x_0 < x_1 < \dots < x_{N_x} < x_{N_x+1}$ and $y_0 < y_1 < \dots < y_{N_y} < y_{N_y+1}$ be the grid points along the two axis. The Grid is

$$G = \{(x_i, y_j) : 0 \leq i \leq N_x + 1, \quad 0 \leq j \leq N_y + 1\}$$

Note that there are N_x internal points on the x axis and N_y internal points on the y axis. The grid can be of four types:

- **Uniform:** TDS Tools generates a uniform cartesian grid with the size and the number of internal points specified by the user. More in details, it is assumed that $x_0 = 0$ and $y_0 = 0$; the user can input the size of the grid, i.e. x_{N_x+1} and y_{N_y+1} , and the number of internal points N_x and N_y .
- **File:** The grid points are read from a user file. The file format is the following:

```
<Nx> <Ny>
<x 0>
<x 1>
...
<x Nx+1>
<y 0>
<y 1>
...
<y Ny+1>
```

- **Pot:** It uses the grid read from the potential file. This way it is not necessary to interpolate the potential for fitting it into the discretization grid. This option is only valid if a Potential file is specified.
- **Estimate:** Not yet implemented. The user can input the number of internal points x_{N_x+1} and y_{N_y+1} and TDS Tools will create a non-uniform grid according to the structure of the potential.

2.2 Initial wave function

Initial wave function can be read by a user file or can be a gaussian packet generated by the TDS Tool. If you specify a gaussian initial wave function, the TDS Tool will initialize the wave function with a Gaussian packet defined as

$$\Psi(x,y) = \Psi_x(x)\Psi_y(y)$$

where

$$\Psi_x(x) = \frac{1}{\sqrt{\sigma_x}\sqrt{2\pi}} e^{-\left(\frac{x-x_0}{2\sigma_x}\right)^2 + i2m^*E_x(x-x_0)}$$

and the same definition holds for Ψ_y . The parameters $x_0, y_0, \sigma_x, \sigma_y, E_x, E_y$ can be specified by the user with the respective names

X0, y0, SigX, SigY, Xnrg, Ynrg

.

If file mode is selected, you can set the filename from which the initial wave function state is read. The file must have the following format:

```
#TDS <Nx> <Ny>
# ... put any comment in a line beginning with '#'

<X coord 1> <Y coord 1> (<real f val 1>, <imag f val 1>)
<X coord 2> <Y coord 2> (<real f val 2>, <imag f val 2>)
<X coord 3> <Y coord 3> (<real f val 3>, <imag f val 3>)
...
```

Nx and Ny are the number of discretization points along the x and y axis. They must be equal to the Nx and Ny values specified in the Grid section. Any following line represent the function value on one point of the grid: you must put exactly Nx*Ny points, which represent the internal grid points. The function on the border is considered to be 0. The points must be on a cartesian grid: if not the software will detect it and terminate with an error.

Example:

```
#TDS 100 100
0          0          (1.0e-4, -0.87e-5)
0.1e-6     0          (0.41e-4, 0.26e-4)
0.2e-6     0          (0.89e-5, 0.87e-5)
...
0          0.15e-6     (0.11e-4, 0.77e-5)
0.1e-6     0.15e-6     (0.34e-4, -0.68e-5)
0.2e-6     0.15e-6     (0.47e-4, 0.23e-5)
...
0          0.3e-6      (-0.18e-4, -0.10e-5)
0.1e-6     0.3e-6      (-0.66e-4, 0.42e-5)
0.2e-6     0.3e-6      (-0.94e-4, 0.88e-5)
...
```

2.3 Potential

The potential function has the same domain of the wave function and is discretized with the same grid previously described. It can be input with three different methods

- **Zero:** the potential function is zero in all the domain. In the borders it is supposed to be infinite.
- **File:** The potential is read from a file. The file format is the following:

```
#TDS <Nx> <Ny>
# ... put any comment in a line beginning with '#'

<X coord 1> <Y coord 1> <pot val 1>
<X coord 2> <Y coord 2> <pot val 2>
<X coord 3> <Y coord 3> <pot val 3>
...
```

N_x and N_y are the number of discretization points along the x and y axis. Any following line represent the function value on one point of the grid: you must put exactly $N_x \cdot N_y$ points, which represent the internal grid points. The potential on the border is considered to be infinite. The points must be on a cartesian grid: if not the software will detect it and terminate with an error. The potential grid can be different from the problem grid described above. In this case the software automatically resamples the potential using a bicubic interpolation scheme to represent the potential on the same grid of the wave function. To avoid a not wanted interpolation, TDS Tools has the parameter **allow_interpolation** which can be 0 or 1. If it is set to 0 and an interpolation is required, TDS Tools exits with an error message.

Example:

```
#TDS 50 50
0          0          1.0e-4
0.2e-6     0          0.26e-4
0.4e-6     0          0.89e-5
...
0          0.15e-6     0.77e-5
0.2e-6     0.15e-6     0.34e-4
0.4e-6     0.15e-6     0.47e-4
...
0          0.3e-6      -0.18e-4
0.2e-6     0.3e-6      0.42e-5
0.4e-6     0.3e-6      -0.94e-4
...
```

- **Description:** TDS Tools also provides a basic scripting language for describing a class of both separable and non separable potentials. The potential is defined by two 1-D functions V_x :

$[x_1, x_{N_x}] \rightarrow R$ and $V_y : [y_1, y_{N_y}] \rightarrow R$ and a 2-D function $V_{xy} : [x_1, x_{N_x}] \times [y_1, y_{N_y}] \rightarrow R$. The potential function is

$$V(x, y) : [x_1, x_{N_x}] \times [y_1, y_{N_y}] \rightarrow R, \quad V(x, y) = V_x(x) + V_y(y) + V_{xy}(x, y)$$

Each 1-D function V_x and V_y is defined by a string, which is a sequence of one or more of the following commands, separated by a semicolon:

- CONSTANT <a> <val>
Sets the value val in all the interval $[a, b]$.
- LINEAR <a> <v1> <v2>
The interval $[a, b]$ is filled with a linear function from (a, v_1) to (b, v_2) .
- POLY3 <a> <v1> <v2>
The interval $[a, b]$ is filled with a third order polynomial connecting the points (a, v_1) and (b, v_2) . It is defined as: $V_x(x) = v_1 + (v_2 - v_1)(3t^2 - 2t^3)$ with $t = \frac{x-a}{b-a}$
- POLY5 <a> <v1> <v2>
The interval $[a, b]$ is filled with a fifth order polynomial connecting the points (a, v_1) and (b, v_2) . It is defined as: $V_x(x) = v_1 + (v_2 - v_1)(10t^3 - 15t^4 + 6t^5)$ with $t = \frac{x-a}{b-a}$
- HARMONIC <a> <Pmax> <P0>
Fills the interval $[a, b]$ with a parabolic potential having the minimum point in the centre of the interval. P_0 is the value in the minimum point and $P_0 + P_{max}$ is the value in a and b . P_0 and P_{max} are expressed in eV. It is defined as $V_x(x) = P_0 + m^* \left(\frac{P_{max}}{\hbar} t \right)^2$ with $t = \frac{x-a}{b-a}$

The 2-D function V_{xy} can only be expressed by means of constant valued boxes. There is a unique command:

BOX <Xul> <Yul> <Xlr> <Ylr> <val>

It fills a box with upper-left vertex (x_{ul}, y_{ul}) and lower-right vertex (x_{lr}, y_{lr}) with value val .

2.4 Output

A set of user parameters can be used for describing how the result of the simulation is output.

- **write_folder** Specify the name of the folder in which the output files are written. If the folder does not exists, TDS Toold automatically creates it.
- **write_grid** can be 0 or 1, and it is a flag which specify if the application have to write into the output folder the **grid.dat** file the size and the x and y nodes of the grid. In case the downsampling is enabled (see below), the downsampled grid is written. The file format is the following:

```

<Nx> <Ny>
<X node 1>
<X node 2>
...
<X node Nx>
<Y node 1>
<Y node 2>
...
<Y node Ny>

```

where Nx and Ny are the number of internal nodes.

- **write_pot** indicates if the potential (eventually interpolated) has to be written in an output file. It can assume the following values: **none** if no potential file has to be written, **txt** if the text version is written in the pot0000.dat file, **bin** if the binary version is written in the pot0000.bin file and **both** if the potential is written in both the text and binary formats. Enabling the binary format is mandatory for using the visualization window in TDS Tools.

The text file has one row for each internal point of the grid and the format is the following:

```

<X coord 1> <Y coord 1> <pot val 1>
<X coord 2> <Y coord 2> <pot val 2>
<X coord 3> <Y coord 3> <pot val 3>

```

The binary file has a little-endian format and has the following data:

```

Bytes 0-3: 32 bit integer containing 8*Nx*Ny
Bytes from 4: the column-wise representation of the potential matrix
               whose elements are double precision reals.

```

- **write_psi** indicates if the wave function has to be written in an output file. It can assume the following values: **none** if no wave function file has to be written, **txt** if the text version is written in the psinnnn.dat file, **bin** if the binary version is written in the psinnnn.bin file and **both** if the potential is written in both the text and binary formats, where nnnn is the timestep number. Enabling the binary format is mandatory for using the visualization window in TDS Tools.

The text file has one row for each internal point of the grid and the format is the following:

```

<X coord 1> <Y coord 1> (<real val 1>, <imag val 1>)
<X coord 2> <Y coord 2> (<real val 2>, <imag val 2>)
<X coord 3> <Y coord 3> (<real val 3>, <imag val 3>)

```

The binary file has a little-endian format and has the following data:

Bytes 0-3: 32 bit integer containing $16 \times N_x \times N_y$

Bytes from 4: the column-wise representation of the wave function matrix whose elements are pairs of double precision reals with the real and imaginary part on each node.

- **write_timestep.** TDS Tools can downsample in time the output. **write_timestep** is the time interval between two successive psi files.
- **write_downsample_x** is an integer downscaling factor along the x axis.
- **write_downsample_y** is an integer downscaling factor along the y axis. Together with the previous parameter, it allows to use a finer grid for a more precise simulation and a coarser file output not to excessively waste disk space or visualization time.