Wissenschaftliches Programmieren
Modul, Abschlussprojekt

Jan Wangerin, Emanuel Schlake

Sep 16, 2019

CONTENTS:

CHAPTER

ONE

ABSCHLUSSPROJEKT API

Automatic tests using pytest. Compares results of the modules to reference data.

test_schroedinger.**test_potential**(*testname*)
Tests whether the potential and eigenvalues match the reference data

Parameters **testname** – Name of the directory the reference data is in.

Reading input and saving output.

modules.in_out.**output**(*potential*, *energies*, *wavefuncs*, *x_exp*, *sigma_x*, *xnew*, *first*, *last*, *path*)
Save potential, eigenvalues, eigenfunctions, expactation value and uncertainty of x into the path folder.

Parameters

- **potential** – Potential of the problem
- **energies** – Energy eigenvalues
- **wavefuncs** – Eigenfunctions
- **x_exp** – The expectation value of x
- **sigma_x** – Uncertainty of x
- **xnew** – x-axis
- **first** – Index of the first eigenvalue/eigenvector to include
- **last** – Index of the last eigenvalue/eigenvector to include
- **path** – Where the output is stored

modules.in_out.**params**(*file*)
Reads the inputfile.

Parameters **file** – The path to the input file. Usually 'schroedinger.inp'

Returns A dictionary containing input information.

Creates a plot of the previously calculated results of 'schroedinger.py'

modules.plot.**visualizer**(*path*, *scale*)
script to plot wavefunctinos, energies, expectations values and uncertainty

Parameters

- **path** – path to the output of a previous calculation
- **scale** – Scales the wavefunctions

Returns matplotlib subplot containing two plots of given output.

Script to find eigenvalues/eigenvectors of the 1-dim schroedinger equation and calculation expectation values of x for a given problem.

modules.solver.**exp_values**(*wavefunc*, *x_min*, *x_max*, *n_point*)

Calculating the expectation values and uncertainty of x. Returns $<x>$ and sigma_x

Parameters

- **wavefunc** – Previously calculated wavefunctions to the problem
- **x_min** – Left end of the x-axis
- **x_max** – Right end of the x-axis
- **n_point** – Number of points the x-axis contains

modules.solver.**solver**(*potential*, *mass*, *x_min*, *x_max*, *n_point*)

Script to solve the 1-dimensional, stationary schroedinger equation for a given potential. Returns eigenvalues (energielevels) and normalised wavefunctions.

Parameters

- **potential** – Potential of the problem
- **mass** – Mass of particle
- **x_min** – Left end of the x-axis
- **x_max** – Right end of the x-axis
- **n_point** – Number of points the x-axis contains

Script to find the intepolating function to a given set of support (x,y).

modules.interpolator.**interpolator**(*x_sup*, *y_sup*, *method*)

Used to interpolate the Potential from a given set of points using a given method. The method can either be polynomial, linear or cspline (natural cubic spline)).

Parameters

- **x_sup** – Supporting x coordinates of the potential
- **y_sup** – Supporting y coordinates of the potential
- **method** – Either polynomial, linear or a cubi spline

Returns The interpolation function of the potential.

PYTHON MODULE INDEX

m

t

INDEX