FourierMagneticPropagator — WaveBlocksND devel documentation

Navigation

- index
- modules |
- next |
- previous |
- WaveBlocksND devel documentation »

FourierMagneticPropagator¶

About the `FourierMagneticPropagator` class¶

The WaveBlocks Project

@author: R. Bourquin @copyright: Copyright (C) 2010, 2011, 2012, 2013, 2014, 2015, 2016 R. Bourquin @license: Modified BSD License

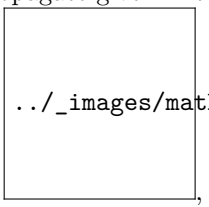
Inheritance diagram¶

digraph inheritance0585146a3b { rankdir=LR; size="8.0, 12.0"; "WaveBlocksND.FourierMagneticPropagator.FourierMagneticPropagator" [fontname="Vera Sans, DejaVu Sans, Liberation Sans, Arial, Helvetica, sans",fontsize=10,height=0.25,shape=box,style="setlinewidth(0.5)",tooltip="This class can numerically propagate given initial values :math:'\Psi(x_0, t_0)' on"]; "WaveBlocksND.Propagator.Propagator" -> "WaveBlocksND.FourierMagneticPropagator.FourierMagneticPropagator" [arrowsize=0.5,style="setlinewidth(0.5)"]; "WaveBlocksND.SplittingParameters.SplittingParameters" -> "WaveBlocksND.FourierMagneticPropagator.FourierMagneticPropagator" [arrowsize=0.5,style="setlinewidth(0.5)"]; "WaveBlocksND.Propagator.Propagator" [fontname="Vera Sans, DejaVu Sans, Liberation Sans, Arial, Helvetica, sans",fontsize=10,height=0.25,shape=box,style="setlinewidth(0.5)",tooltip="Propagators can numerically simulate the time evolution of quantum states"]; "WaveBlocksND.SplittingParameters.SplittingParameters" [fontname="Vera Sans, DejaVu Sans, Liberation Sans, Arial, Helvetica, sans",fontsize=10,height=0.25,shape=box,style="setlinewidth(0.5)"]; }

Class documentation¶

class WaveBlocksND.FourierMagneticPropagator(*parameters*, *potential*, *initial_values*)[source]¶

This class can numerically propagate given initial values 

on a potential hyper surface 

, in presence of a magnetic field. The propagation is done with a splitting of the time propagation operator 

. Available splitting schemes are implemented in SplittingParameters.

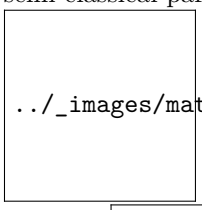
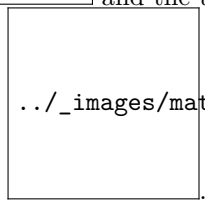
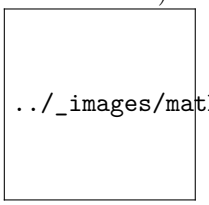
__init__(*parameters*, *potential*, *initial_values*)[source]¶

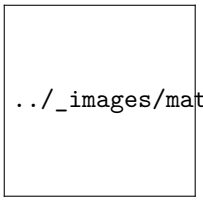
Initialize a new FourierMagneticPropagator instance. Pre-calculate the the kinetic operator 

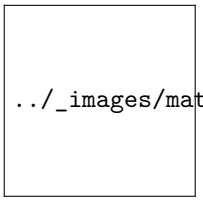
and the potential operator 

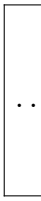
used for time propagation.

Parameters:

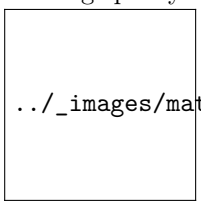
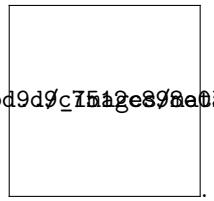
- **parameters** – The set of simulation parameters. It must contain at least the semi-classical parameter  and the time step size  .
- **potential** (A `MatrixPotential` instance.) – The potential  governing the time evolution.
- **initial_values** (A `WaveFunction` instance.) – The initial values  given in the canonical basis.

Raise: `ValueError` If the number of components of  does no

Raise: `ValueError` If the number of components of  does no

Raise: `ValueError` If the dimensions of the splitting scheme parameters 

Parameters: **method** – A string specifying the method for time integration.

Returns: Two arrays  and  .

Method	Order	Authors	Reference
LT	1	Lie/Trotter	[1], [3] page 42, e
S2	2	Strang	[2], [3] page 42, e
SS	2	Strang	[2], [3] page 42, e
PRKS6	4	Blanes/Moan	[4] page 318, tab
BM42	4	Blanes/Moan	[4] page 318, tab
Y4	4	Yoshida	[5], [3] page 40, e
Y61	6	Yoshida	[5], [3] page 144,
BM63	6	Blanes/Moan	[4] page 318, tab
KL6	6	Kahan/Li	[6], [3] page 144,
KL8	8	Kahan/Li	[6], [3] page 145,
KL10	10	Kahan/Li	[6], [3] page 146,

[1] H.F. Trotter, “On the product of semi-groups of operators”, Proc. Am. Math. Soc.1O (1959) 545-551.

[2] *(1, 2)* G. Strang, “On the construction and comparison of difference schemes”, SIAM J. Numer. Anal. 5

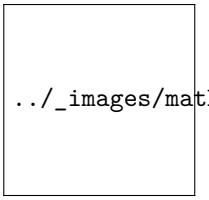
[3] *(1, 2, 3, 4, 5, 6, 7, 8)* E. Hairer, C. Lubich, and G. Wanner, "Geometric Numerical Integration - Structu

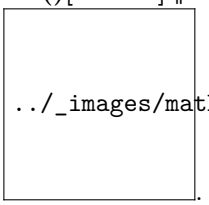
[4] *(1, 2, 3)* S. Blanes and P.C. Moan, "Practical Symplectic Partitioned Runge-Kutta and Runge-Kutta-Ny

[5] *(1, 2)* H. Yoshida, "Construction of higher order symplectic integrators", Phys. Lett. A 150 (1990) 262-2

[6] *(1, 2, 3)* W. Kahan and R.-c. Li, "Composition constants for raising the orders of unconventional schem

build(*method*)¶

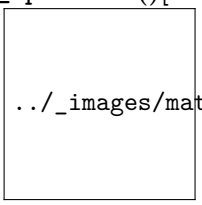


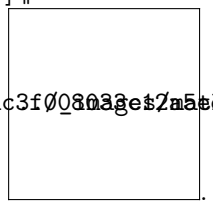
get_number_components()[source]¶ Get the number 

of components of .

Returns: The number .

get_operators()[source]¶ Get the kinetic and potential operators



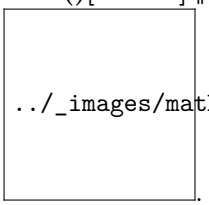


and .

Returns: A tuple containing two `ndarrays`.

`get_potential`()¶ Returns the potential used for time propagation.

Returns: A `MatrixPotential` subclass instance.

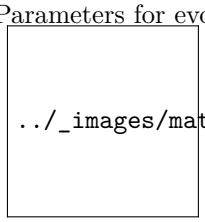
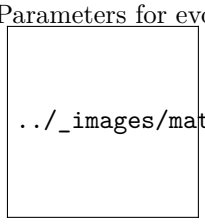
`get_wavefunction`()[source]¶ Get the wavefunction that stores the current data 

.

Returns: The `WaveFunction` instance.

`intsplit`(*psi1*, *psi2*, *a*, *b*, *tspan*, *N*, *args1=[]*, *args2=[]*)¶
Compute a single, full propagation step by operator splitting.

Parameters:

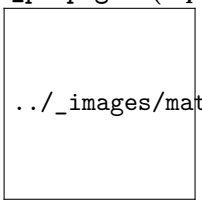
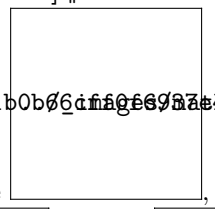
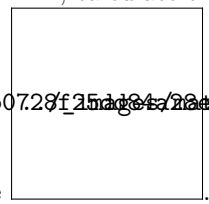
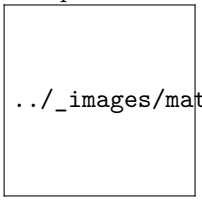
- **psi1** – First evolution operator 
- **psi2** – Second evolution operator 
- **a** – Parameters for evolution with 
- **b** – Parameters for evolution with 
- **tspan** – Timespan  of a single, full splitting step
- **N** – Number of substeps to perform
- **args1** – Additional optional arguments of 
- **args2** – Additional optional arguments of 

Note

The values for `args1` and `args2` have to be of type `list` even in case of single items.

Parameters: **method** – A string specifying the method for time integration.
Returns: The order of this method.

`order`(*method*)¶

`post_propagate`(*tspan*)[**source**]¶ Given an initial wavepacket  at time  , calculate the propagated wavepacket  at time  . We perform  steps of size  .

Parameters: **tspan** – `ndarray` consisting of end time at position 0, other positions are irrelevant.

`pre_propagate`()¶ Given the wavefunction  at initial time  , perform some computations exactly once be-

fore running the ordinary time propagation and after each time simulation data was saved.

This method does not raise an exception but instead just does nothing and returns.

`propagate`(*tspan*)[**source**]¶ This method does nothing.

Table Of Contents

Previous topic

FourierPropagator

Next topic

ChinChenPropagator

This Page

- Show Source

Quick search

Navigation