

# QuDynamics : Julia package for solving Dynamical Equations in Quantum Mechanics

Amit

BITS-Pilani and Fudan University

Julia Users Meetup,  
University of Science and Technology of China

25-12-2015

## Introduction

What is Julia and Why ?!

Hello World! Julia

## Introduction to Julia

Functions, Methods, Types

Packaging in Julia

## JuliaQuantum : QuBase, QuDynamics

QuBase and QuDynamics

Defining a Quantum System

Examples in QuDynamics

## QuDynamics and FOSS

What is Git ?!

Contributing to FOSS and QuDynamics

# What is Julia and Why ?!

# What is Julia and Why ?!

- ▶ Julia is a high-level, high-performance programming language for technical computing.

# What is Julia and Why ?!

- ▶ Julia is a high-level, high-performance programming language for technical computing.
- ▶ Julia is written in Julia (Base).

# What is Julia and Why ?!

- ▶ Julia is a high-level, high-performance programming language for technical computing.
- ▶ Julia is written in Julia (Base).
- ▶ Performance as fast as compared to C (Every language has a tradeoff but Julia aims to be optimal for scientific computing).

# What is Julia and Why ?!

- ▶ Julia is a high-level, high-performance programming language for technical computing.
- ▶ Julia is written in Julia (Base).
- ▶ Performance as fast as compared to C (Every language has a tradeoff but Julia aims to be optimal for scientific computing).
- ▶ Parallelization techniques and distributed computing come along with the base distribution.

# What is Julia and Why ?!

- ▶ Julia is a high-level, high-performance programming language for technical computing.
- ▶ Julia is written in Julia (Base).
- ▶ Performance as fast as compared to C (Every language has a tradeoff but Julia aims to be optimal for scientific computing).
- ▶ Parallelization techniques and distributed computing come along with the base distribution.
- ▶ Julia is FOSS allowing discussion of internals and design.



# Hello World! Julia

# Hello World! Julia

- ▶ Use either SageMathCloud ([cloud.sagemath.com](http://cloud.sagemath.com)) (or) JuliaBox ([juliabox.org](http://juliabox.org)).

# Hello World! Julia

- ▶ Use either SageMathCloud ([cloud.sagemath.com](http://cloud.sagemath.com)) (or) JuliaBox ([juliabox.org](http://juliabox.org)).
- ▶ REPL - Read - Evaluate - Print - Loop

# Hello World! Julia

- ▶ Use either SageMathCloud ([cloud.sagemath.com](http://cloud.sagemath.com)) (or) JuliaBox ([juliabox.org](http://juliabox.org)).
- ▶ REPL - Read - Evaluate - Print - Loop
- ▶ Hello World ! — `> println("Hello World !")`

# Functions, Methods, Types

# Functions, Methods, Types

## ► Functions

# Functions, Methods, Types

- ▶ Functions
- ▶ Structure which takes a tuple and returns a value

# Functions, Methods, Types

- ▶ Functions
- ▶ Structure which takes a tuple and returns a value
- ▶ Methods (OOP reference – > functional programming)



# Functions, Methods, Types

- ▶ Functions
- ▶ Structure which takes a tuple and returns a value
- ▶ Methods (OOP reference – > functional programming)
- ▶ One possible behavior for a function is called a method (Multiple Dispatch)

# Functions, Methods, Types

- ▶ Functions
- ▶ Structure which takes a tuple and returns a value
- ▶ Methods (OOP reference – > functional programming)
- ▶ One possible behavior for a function is called a method (Multiple Dispatch)
- ▶ Types

# Functions, Methods, Types

- ▶ Functions
- ▶ Structure which takes a tuple and returns a value
- ▶ Methods (OOP reference – > functional programming)
- ▶ One possible behavior for a function is called a method (Multiple Dispatch)
- ▶ Types
- ▶ Examples !

# Packaging in Julia !

# Packaging in Julia !

- ▶ REPL *Pkg* usage

# Packaging in Julia !

- ▶ REPL *Pkg* usage
- ▶ Using *Pkg.add()*, *Pkg.clone()*, *Pkg.rm()*, *Pkg.installed()*, *Pkg.status()*

# QuBase & QuDynamics

# QuBase & QuDynamics

- ▶ QuBase aims to provide all the basic machinery used in QM.



# QuBase & QuDynamics

- ▶ QuBase aims to provide all the basic machinery used in QM.
- ▶ QuDynamics uses the constructs of QuBase along with the Base machinery.

# Examples in QuDynamics

# Examples in QuDynamics

- ▶ QuStateEvolution (alias QuPropagator) construct

## Examples in QuDynamics

- ▶ QuStateEvolution (alias QuPropagator) construct
- ▶ Hamiltonian, initial state, time steps, Methods

## Examples in QuDynamics

- ▶ QuStateEvolution (alias QuPropagator) construct
- ▶ Hamiltonian, initial state, time steps, Methods
- ▶ Methods : QuEuler, QuCrankNicolson, QuKrylov, QuODE45, QuODE78, QuODE23s, QuExpV, QuExpokit, QuMCWF

## Examples in QuDynamics

- ▶ QuStateEvolution (alias QuPropagator) construct
- ▶ Hamiltonian, initial state, time steps, Methods
- ▶ Methods : QuEuler, QuCrankNicolson, QuKrylov, QuODE45, QuODE78, QuODE23s, QuExpV, QuExpokit, QuMCWF
- ▶ System input include – > Schrodinger Eq, Liouville von Neumann Eq, Lindblad Master Eq.

## Examples in QuDynamics

- ▶ QuStateEvolution (alias QuPropagator) construct
- ▶ Hamiltonian, initial state, time steps, Methods
- ▶ Methods : QuEuler, QuCrankNicolson, QuKrylov, QuODE45, QuODE78, QuODE23s, QuExpMV, QuExpokit, QuMCWF
- ▶ System input include – > Schrodinger Eq, Liouville von Neumann Eq, Lindblad Master Eq.
- ▶ Configurations allowed ! – > Operator matrix with state vector, Operator matrix with density matrix.

## Examples in QuDynamics

- ▶ QuStateEvolution (alias QuPropagator) construct
- ▶ Hamiltonian, initial state, time steps, Methods
- ▶ Methods : QuEuler, QuCrankNicolson, QuKrylov, QuODE45, QuODE78, QuODE23s, QuExpMV, QuExpokit, QuMCWF
- ▶ System input include – > Schrodinger Eq, Liouville von Neumann Eq, Lindblad Master Eq.
- ▶ Configurations allowed ! – > Operator matrix with state vector, Operator matrix with density matrix.
- ▶ Other ways of specifying ! – > QuSchrodingerEq(pass the operator) Internally passing a operator matrix creates an equation.



## Examples in QuDynamics

- ▶ QuStateEvolution (alias QuPropagator) construct
- ▶ Hamiltonian, initial state, time steps, Methods
- ▶ Methods : QuEuler, QuCrankNicolson, QuKrylov, QuODE45, QuODE78, QuODE23s, QuExpMV, QuExpokit, QuMCWF
- ▶ System input include – > Schrodinger Eq, Liouville von Neumann Eq, Lindblad Master Eq.
- ▶ Configurations allowed ! – > Operator matrix with state vector, Operator matrix with density matrix.
- ▶ Other ways of specifying ! – > QuSchrodingerEq(pass the operator) Internally passing a operator matrix creates an equation.
- ▶ Quantum Monte Carlo Wave Function Method (Test case example)

# What is Git ?!

# What is Git ?!

- ▶ Git is a version control system.

# What is Git ?!

- ▶ Git is a version control system.
- ▶ Clones, forks, branches ?!

# What is Git ?!

- ▶ Git is a version control system.
- ▶ Clones, forks, branches ?!
- ▶ Push, pull, remotes !

# Contributing to FOSS and QuDynamics

# Contributing to FOSS and QuDynamics

- ▶ QuDynamics is FOSS. (MIT license)

# Contributing to FOSS and QuDynamics

- ▶ QuDynamics is FOSS. (MIT license)
- ▶ Fork the repo, Clone locally, New branch



# Contributing to FOSS and QuDynamics

- ▶ QuDynamics is FOSS. (MIT license)
- ▶ Fork the repo, Clone locally, New branch
- ▶ Test locally, include the tests.

## Contributing to FOSS and QuDynamics

- ▶ QuDynamics is FOSS. (MIT license)
- ▶ Fork the repo, Clone locally, New branch
- ▶ Test locally, include the tests.
- ▶ Send in a PR (Pull Request) against the master !

## Contributing to FOSS and QuDynamics

- ▶ QuDynamics is FOSS. (MIT license)
- ▶ Fork the repo, Clone locally, New branch
- ▶ Test locally, include the tests.
- ▶ Send in a PR (Pull Request) against the master !
- ▶ Some of the WIP and future tasks can be found on the issue tracker

## Contributing to FOSS and QuDynamics

- ▶ QuDynamics is FOSS. (MIT license)
- ▶ Fork the repo, Clone locally, New branch
- ▶ Test locally, include the tests.
- ▶ Send in a PR (Pull Request) against the master !
- ▶ Some of the WIP and future tasks can be found on the issue tracker
- ▶ Time dependency, QuTiP interface, Docs, QuBase independency, parallelization !

## Contributing to FOSS and QuDynamics

- ▶ QuDynamics is FOSS. (MIT license)
- ▶ Fork the repo, Clone locally, New branch
- ▶ Test locally, include the tests.
- ▶ Send in a PR (Pull Request) against the master !
- ▶ Some of the WIP and future tasks can be found on the issue tracker
- ▶ Time dependency, QuTiP interface, Docs, QuBase independency, parallelization !
- ▶ To add more solvers refer to the Wiki page !

## Contributing to FOSS and QuDynamics

- ▶ QuDynamics is FOSS. (MIT license)
- ▶ Fork the repo, Clone locally, New branch
- ▶ Test locally, include the tests.
- ▶ Send in a PR (Pull Request) against the master !
- ▶ Some of the WIP and future tasks can be found on the issue tracker
- ▶ Time dependency, QuTiP interface, Docs, QuBase independency, parallelization !
- ▶ To add more solvers refer to the Wiki page !
- ▶ More development notes on the news page of JuliaQuantum tagged as GSoC.

## Contributing to FOSS and QuDynamics

- ▶ QuDynamics is FOSS. (MIT license)
- ▶ Fork the repo, Clone locally, New branch
- ▶ Test locally, include the tests.
- ▶ Send in a PR (Pull Request) against the master !
- ▶ Some of the WIP and future tasks can be found on the issue tracker
- ▶ Time dependency, QuTiP interface, Docs, QuBase independency, parallelization !
- ▶ To add more solvers refer to the Wiki page !
- ▶ More development notes on the news page of JuliaQuantum tagged as GSoC.
- ▶ Example

# Thank you



- ▶ Done as a part of Google Summer of Code 2015
- ▶ Mentored by Dr. Alexander Croy, Max Planck Institute for the Physics of Complex Systems, Germany.
- ▶ Repo : <https://github.com/JuliaQuantum/QuDynamics.jl>