

# SQuIDS: A Tool to Solve Time Evolution in finite dimensional (open) Quantum Systems

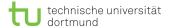
An Application to Neutrino Oscillations

arxiv:1412.3832

Dominik Hellmann

TU Dortmund WG Päs

May 3, 2023



Introduction

2 SQuIDS

3 Exercises

#### Motivation

Task: Solve time evolution of finite dimensional quantum (sub-)systems:

- ► Flavor oscillations
- Quantum computation
- ► Systems with finitely many energy levels
- Spins

Time evolution of closed quantum system: Schrödinger equation

$$i\frac{\partial}{\partial t}|\psi\rangle = \hat{H}|\psi\rangle \qquad (\hbar = 1)$$
 (1)



# Density matrices instead of state vectors

Often: finite dimensional system S coupled to a complicated (but uninteresting) environment E

- → Get rid of Environment (keyword: partial trace)
- → Just consider degrees of freedom of interest

#### Consequence: Decoherence

- ightharpoonup Subsystem cannot be described by pure state  $|\psi\rangle$
- ▶ Mixed state: Described by density matrix  $\varrho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$



# Example: Neutrino oscillations in matter

$\mathcal{S}\simeq\mathbb{C}^3$	E
flavor degrees of freedom	all remaining d.o.f (momenta, spins,)
$ \psi angle = \sum_{lpha=\mathbf{e}}^{ au} \psi_{lpha}   u_{lpha} angle$	ightarrow infinite dimensional

- ▶ We are not at all interested in E
- Only the  $\nu$  flavor composition is interesting to us
- **But**: E significantly influences flavor d.o.f.
- ⇒ Need effective description!



# Time evolution of the density matrix

Master equation(s): (multiple density matrices possible)

$$\frac{\mathrm{d}\varrho_j}{\mathrm{d}t} = -i[\hat{H}_j(t),\varrho_j(t)] + \{\Gamma_j(t),\varrho_j(t)\} + F_j[\{\varrho_k\}_k,t]$$

- ▶ Why multiple  $\varrho_i$ ? E.g.: One per energy bin!
- $\hat{H} = \hat{H}_0 + \hat{H}_1(t)$ : Unitary evolution
- Γ: Decoherence
- ▶ F: Other non-linear effects (coupling between  $\varrho_i$ )



# Simple Example: $\nu$ Oscillations in Vacuum

#### Neutrino Experiment:

- ► Fixed baseline *L*
- $\triangleright$  N energy bins  $\{E_i\}_i$

$$\hat{H}^{j} = \hat{H}_0^{j} = E_j \cdot \mathbb{I} + \frac{1}{2E_i} \mathbb{M}^2$$

- Γ ≡ 0
- $F \equiv 0$
- $\blacktriangleright \ \varrho_j(t) = \sum_{\alpha=e}^{\tau} \phi_{\alpha}^j(t) |\nu_{\alpha}\rangle\langle\nu_{\alpha}|$

$$\begin{split} \mathbb{M} &= \sum_{j,k=1}^{3} (\mathbb{M}_{0})_{jk} |\nu_{j}\rangle \langle \nu_{k}| \\ &= \sum_{\alpha,\beta=e}^{\tau} (\mathbb{M}_{1})_{\alpha\beta} |\nu_{\alpha}\rangle \langle \nu_{\beta}| \\ \mathbb{M}_{0} &= \begin{pmatrix} m_{1} & 0 & 0 \\ 0 & m_{2} & 0 \\ 0 & 0 & m_{3} \end{pmatrix} \\ \mathbb{M}_{1} &= U_{\mathrm{PMNS}}^{\dagger} \mathbb{M}_{1} U_{\mathrm{PMNS}} \end{split}$$



# Some Important Considerations

The master equation simplifies to

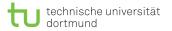
$$\frac{\mathrm{d}\varrho_j}{\mathrm{d}t} = -i[\hat{H}_0^j, \varrho_j(t)]$$

Further simplifications

- ightharpoonup Consider in mass basis:  $\hat{H}_0^j$  is diagonal
- ▶ Depends only on commutator!
  - $[A, \mathbb{I}] = 0 \Rightarrow [\hat{H}_0^j, \varrho_i(t)] = [\hat{H}_0^j \epsilon_i \mathbb{I}, \varrho_i(t)]$
  - $\epsilon_j := E_j + m_1^2/2E_j$

$$ilde{H}^j = rac{1}{2 E_j} egin{pmatrix} 0 & 0 & 0 & 0 \ 0 & \Delta m_{21}^2 & 0 \ 0 & 0 & \Delta m_{31}^2 \end{pmatrix}$$





# Some Important Considerations (ctd.)



# Summary

What did we learn so far (in general)

- 1. We passed to density matrix formulation (allows for mixed states)
- 2. Formulated master equation
- 3. Can subtract a part prop. to identity from  $\hat{H}$  (only energy diff. important)
- 4. Can solve  $\hat{H}_0$  exactly (interaction picture)  $\varrho \to e^{-i\hat{H}_0\Delta T}\varrho e^{i\hat{H}_0\Delta T}$



## Overview



## Const



#### SU Vector



# **SQuIDS**



# Clone the Repo!

Git Repository includes all needed files (slides, code templates)

#### Instructions

cd to the location where you want to place the repo
git clone https://github.com/BObsen/sm\_to\_bsm\_neutrino.git
cd sm\_to\_bsm\_neutrino



#### Const class exercise

- 1. Declare a default constructed const class object
- 2. Answer the following questions:
  - 2.1 How many  ${\rm eV}^{-1}$  correspond to  $300\,{\rm km}$
  - 2.2 How many radians correspond to 25°
  - 2.3 If you are 24 years old, how many  $eV^{-1}$  are you old?
- 3. Set the mixing parameters for three neutrino generations to:
  - $\theta_{12} = 33.48^{\circ}$
  - $\theta_{13} = 8.55^{\circ}$
  - $\theta_{23} = 42.3^{\circ}$
- 4. Set the energy differences to:
  - $\Delta m_{21}^2 = 7.5 \cdot 10^{-5} \, \text{eV}^2$
  - $\Delta m_{31}^{21} = 2.45 \cdot 10^{-3} \,\mathrm{eV}^2$



#### SU vector exercise

- 1. Declare an empty SU vector corresponding to a 3D Hilbert space
- 2. Initialize an array of projectors for the three mass eigenstates  $(B_0)$
- 3. Rotate them to the flavor basis  $(B_1)$
- 4. Initialize a SU vector corresponding to the matrix  $(B_0)$

$$\Delta \mathbb{M}^2 := egin{pmatrix} 0 & 0 & 0 \ 0 & \Delta m_{21}^2 & 0 \ 0 & 0 & m_{31}^2 \end{pmatrix}$$



#### **Exercises**

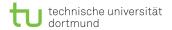
SQuIDS application: Neutrino oscillations in vacuum



#### **Exercises**



#### **Exercises**



# **BACK UP**



#### Installation

What do we need for this tutorial?

- ► A unix-like (sub-)system
  - ► Linux
  - ► Mac (+ Xcode developer tools!)
  - On Windows: WSL
- ► A C++ compiler
- ► Make, wget, Git

Use scripts install\_gsl.sh and install\_SQuIDS.sh from the repo!



# Installation (GSL)

```
cd $HOME
mkdir -p smToBsmLibs/gsl
wget ftp://ftp.gnu.org/gnu/gsl/gsl-latest.tar.gz
tar -zxvf gsl-latest.tar.gz
rm gsl-latest.tar.gz
cd $ffind gsl-* | head -n 1)
./configure --prefix=$HOME/smToBsmLibs/gsl
make
make check
make install
LD_LIBRARY_PATH=$HOME/smToBsmLibs/gsl/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
cd $HOME
rm -rf $(find gsl-* | head -n 1)
```



# Installation (SQuIDS)

```
cd $HOME
mkdir -p smToBsmLibs/SQuIDS
git clone https://github.com/jsalvado/SQuIDS.git
cd $(find SQuIDS* | head -n 1)
./configure --with-gsl-incdir=$HOME/smToBsmLibs/gsl/include \
--with-gsl-libdir=$HOME/smToBsmLibs/gsl/lib \
--prefix=$HOME/smToBsmLibs/SQuIDS
make
make test
make install
LD_LIBRARY_PATH=$HOME/smToBsmLibs/SQuIDS/lib:$LD_LIBRARY_PATH # linux only
export LD_LIBRARY_PATH # linux only
cd $HOME
rm -rf $(find SQuIDS* | head -n 1)
```