

# $\nu$ -sim

Mateus Marques

February 5, 2022

## 1 Numerical description

In general:

$$i \frac{d\psi}{dt} = \mathcal{H}\psi \Rightarrow \frac{d}{dt} \begin{bmatrix} \Re \\ \Im \end{bmatrix} = \begin{bmatrix} \mathcal{H}_{\Im} & \mathcal{H}_{\Re} \\ -\mathcal{H}_{\Re} & \mathcal{H}_{\Im} \end{bmatrix} \begin{bmatrix} \Re \\ \Im \end{bmatrix},$$

where  $\psi = \Re + i\Im = (\Re_1 + i\Im_1, \dots, \Re_n + i\Im_n)$ ,  $\mathcal{H} = \mathcal{H}_{\Re} + i\mathcal{H}_{\Im}$  being  $\Re, \Im, \mathcal{H}_{\Re}, \mathcal{H}_{\Im}$  real.

The original complex system of  $n$  equations becomes a real system with  $2n$  equations.

In the case of Neutrino Oscillations in matter, our hamiltonian is always of the form:

$$\mathcal{H} = \mathcal{H}^0 + \text{diag}(V(L), 0, \dots, 0),$$

where  $\mathcal{H}^0$  is constant and  $V(L) = \sqrt{2}G_F N_e(L)$  is the only parameter that depends on the traveled distance  $L$  (time also, because  $L = ct = t$ ).  $G_F$  is the Fermi constant and  $N_e(L)$  is the solar electron density. The hamiltonian  $\mathcal{H}$  is so simple because the only neutrino that interacts with solar matter is the neutrino  $\nu_e$  of the electron.

The matrix  $\mathcal{H}^0$  is simply given by the sandwich:

$$\mathcal{H}^0 = U M U^\dagger,$$

where  $U$  is the neutrino mixing matrix (it's only a unitary matrix, and it has standard parametrization), and  $M$  is the diagonal matrix corresponding to the mass eigenvalues of the neutrinos in vacuum (it can be simplified, making one of its entries zero).

Now, the algorithm is very simple:

1. Assume  $U, M$  and a table  $L \times N_e(L)$  of data as input.
2. Using the following structures of the [GSL Library](#):

```
gsl_complex, gsl_matrix_complex, gsl_matrix;
```

we easily calculate  $\mathcal{H}^0 = \mathcal{H}_{\Re}^0 + i\mathcal{H}_{\Im}^0$  by complex matrix operations and then obtain  $\mathcal{H}_{\Re}^0, \mathcal{H}_{\Im}^0$  by taking the real and imaginary parts with the C macros:

```
GSL_REAL, GSL_IMAG.
```

3. Interpolate  $N_e(L)$  using [Bahcall's data](#) and compute  $V(L) = \sqrt{2}G_F N_e(L)$  for  $-R_\odot \leq L \leq R_\odot$ , where  $R_\odot$  is the solar radius. Here we use the following header and type from GSL:

```
#include <gsl/gsl_spline.h>
gsl_interp_steffen,
```

which by the last [1D Interpolation Example](#) seems to be the best spline for the case.

4. Now we simply solve the following ODE numerically and print the results.

$$\frac{d}{dt} \begin{bmatrix} \Re \\ \Im \end{bmatrix} = \begin{bmatrix} \mathcal{H}_{\Im} & \mathcal{H}_{\Re} \\ -\mathcal{H}_{\Re} & \mathcal{H}_{\Im} \end{bmatrix} \begin{bmatrix} \Re \\ \Im \end{bmatrix},$$

where  $\mathcal{H}_{\Re} = \mathcal{H}_{\Re}^0 + \text{diag}(V(L), 0, \dots, 0)$  and  $\mathcal{H}_{\Im} = \mathcal{H}_{\Im}^0$ . For this we use the header

```
#include <gsl/gsl_odeiv2.h>.
```