PEANUTS: AUTOMATIC COMPUTATION OF SOLAR NEUTRINO PROPAGATION

Michele Lucente

INFN TAsP Meeting, Turin
January 19, 2024

Based on:

ML and T. E. Gonzalo, arXiv:**2303.15527** [hep-ph]

https://github.com/michelelucente/PEANUTS

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

**Funded by
the European Union**

Neutrino physics is now in its precision era

Parameter	Ordering	Best fit	1σ range	2σ range	3σ range	"1σ" (%)
$\delta m^2/10^{-5}$ eV^2	NO, IO	7.36	7.21 – 7.52	7.06 – 7.71	6.93 – 7.93	2.3
$\sin^2\theta_{12}/10^{-1}$	NO, IO	3.03	2.90 – 3.16	2.77 – 3.30	2.63 – 3.45	4.5
$\|\Delta m^2\|/10^{-3}$ eV^2	NO	2.485	2.454 – 2.508	2.427 – 2.537	2.401 – 2.565	1.1
	IO	2.455	2.430 – 2.485	2.403 – 2.513	2.376 – 2.541	1.1
$\sin^2\theta_{13}/10^{-2}$	NO	2.23	2.17 – 2.30	2.11 – 2.37	2.04 – 2.44	3.0
	IO	2.23	2.17 – 2.29	2.10 – 2.38	2.03 – 2.45	3.1
$\sin^2\theta_{23}/10^{-1}$	NO	4.55	4.40 – 4.73	4.27 – 5.81	4.16 – 5.99	6.7
	IO	5.69	5.48 – 5.82	4.30 – 5.94	4.17 – 6.06	5.5
δ/π	NO	1.24	1.11 – 1.42	0.94 – 1.74	0.77 – 1.97	16
	IO	1.52	1.37 – 1.66	1.22 – 1.78	1.07 – 1.90	9
$\Delta\chi^2_{\rm IO-NO}$	IO−NO	+6.5				

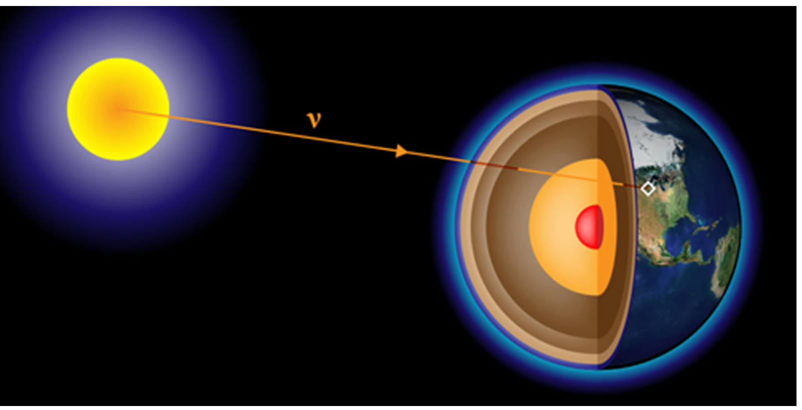
F. Capozzi, E. Di Valentino, E. Lisi, A. Marrone, A. Melchiorri and A. Palazzo, arXiv:2107.00532 [hep-ph]

A wealth of information can be derived from combining independent observations

Need for tools to compute oscillation dynamics in a fast and precise way

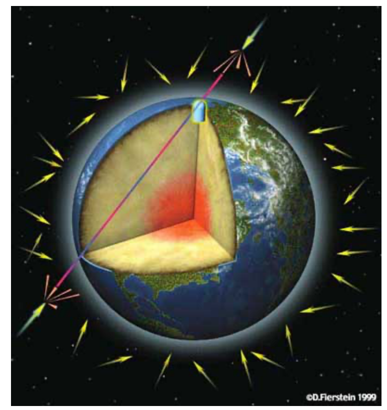
Matter effects relevant for most experiments

Solar



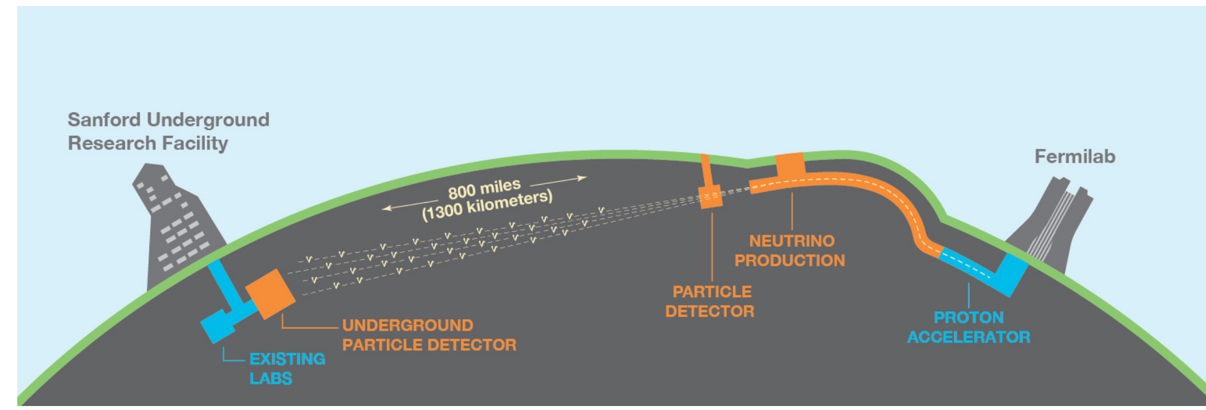
Alan Stonebraker/APS

Atmospheric



David Fierstein, Scientific American

Long-baseline



DUNE collaboration

For propagation within Earth evolution is usually solved numerically

$$i \frac{\mathrm{d}}{\mathrm{d}t} c_\alpha(t) = H_{\alpha\beta}(t) c_\beta(t) \qquad c_\alpha(t) = \langle \nu_\alpha | \nu(t) \rangle$$

This can be a very time consuming task

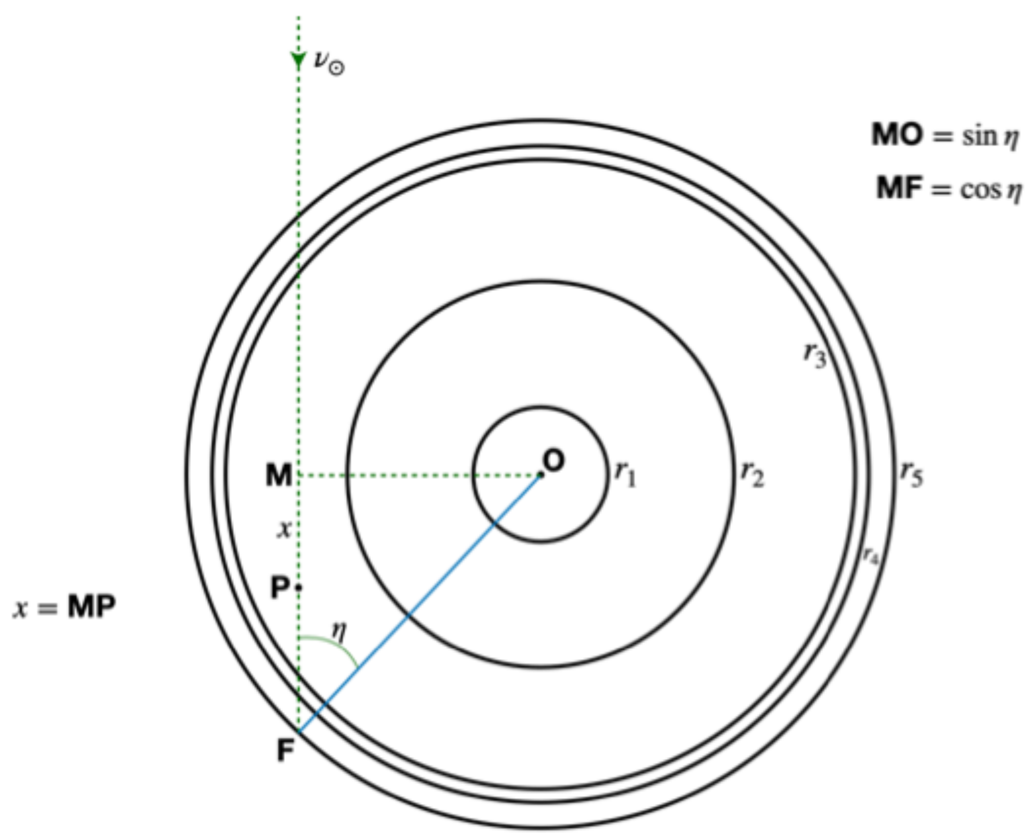
Alternative solution: time-dependent perturbation theory

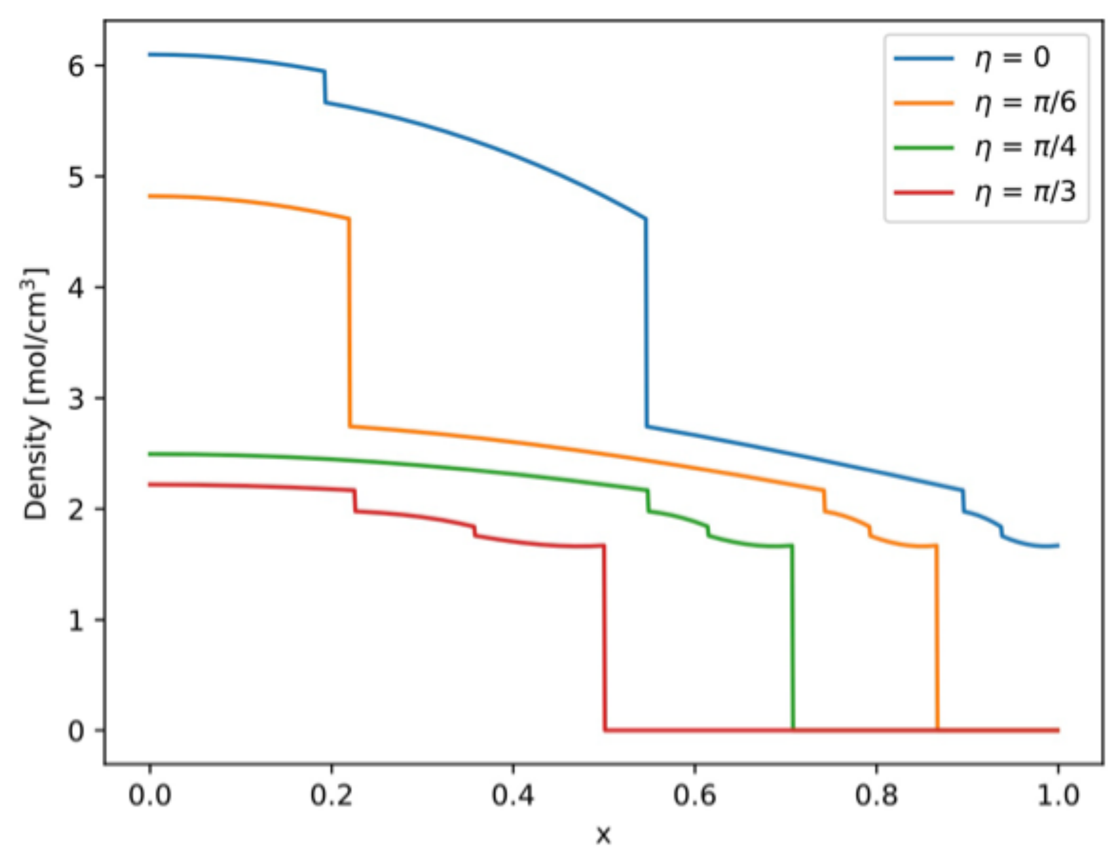
The evolution of active neutrinos within Earth can be solved analytically

E. Lisi and D. Montanino, hep-ph/9702343

$$H_\nu = \underbrace{U^* \mathrm{diag}(k) U^T}_{H_\nu^0} + \underbrace{V(x)\ \mathrm{diag}(1,0,0)}_{V_\nu}$$

$$k = \frac{\Delta m^2}{2E} = \frac{2.533}{m} \times \frac{\Delta m^2}{\mathrm{eV}^2} \times \frac{\mathrm{MeV}}{E} \qquad V = \sqrt{2} G_F n_e = \frac{3.868 \times 10^{-7}}{m} \times \frac{n_e}{\mathrm{mol/cm}^3}$$





Parametrisation of Earth density

Earth interior can be parametrised as a 5-shells structure, within which density varies smoothly

E. Lisi and D. Montanino, hep-ph/9702343

$$N_j(r) = \alpha_j + \beta_j r^2 + \gamma_j r^4$$

$$[N] = \mathrm{mol/cm^3}$$

j	Shell	$[r_{j-1}, r_j]$	α_j	β_j	γ_j
1	Inner core	[0, 0.192]	6.099	−4.119	0.000
2	Outer core	[0.192, 0.546]	5.803	−3.653	−1.086
3	Lower mantle	[0.546, 0.895]	3.156	−1.459	0.280
4	Transition Zone	[0.895, 0.937]	−5.376	19.210	−12.520
5	Upper mantle	[0.937, 1]	11.540	−20.280	10.410

For non-radial paths, the parametrisation is functionally invariant

$$N_j(x) = \alpha'_j + \beta'_j x^2 + \gamma'_j x^4$$

$$\begin{aligned} \alpha'_j &= \alpha_j + \beta_j \sin^2\eta + \gamma_j \sin^4\eta, \\ \beta'_j &= \beta_j + 2\gamma_j \sin^2\eta, \\ \gamma'_j &= \gamma_j, \end{aligned}$$

The precision of the expansion can be improved by considering the variable δn(x)

$$n_e(x) = \bar{n}_e + \delta n(x) \qquad \bar{n}_e = \frac{1}{x_2 - x_1}\int_{x_1}^{x_2} \mathrm{d}x \; n_e(x)$$

$$\tilde{H}(x) = \underbrace{\tilde{H}_k + \sqrt{2}G_F \bar{n}_e \mathrm{diag}(1,0,0)}_{\tilde{H}_0} + \underbrace{\sqrt{2}G_F \delta n(x) \mathrm{diag}(1,0,0)}_{\delta\tilde{H}(x)}$$

Perturbative solution

If a perturbative expansion exist

$$\tilde{H}(x) = \underbrace{\tilde{H}_k + \sqrt{2}G_F\bar{n}_e\mathrm{diag}(1,0,0)}_{\tilde{H}_0} + \underbrace{\sqrt{2}G_F\delta n(x)\mathrm{diag}(1,0,0)}_{\delta\tilde{H}(x)}$$

The evolutor operator can be expanded as

E. Lisi and D. Montanino, hep-ph/9702343

$$\mathcal{U}(x_2,x_1) = \bar{\mathcal{U}}(x_2,x_1) - i\int_{x_1}^{x_2} \mathrm{d}x\, \bar{\mathcal{U}}(x_2,x)\ \delta\tilde{H}(x)\ \bar{\mathcal{U}}(x,x_1) + \mathcal{O}(\delta\tilde{H}^2)$$

Evolutor for constant density ($H = H_0$)

We want to solve this integral for the general 3-flavour case

3-flavour perturbed evolutor

The evolutor for a constant Hamiltonian can generally be expressed in a closed form

$$e^{-i\bar{H}x} = \phi \sum_{a=1}^{3} e^{-ix\lambda_a} \frac{1}{3\lambda_a^2 + c_1} \left[\left(\lambda_a^2 + c_1 \right) \mathbb{1} + \lambda_a T + T^2 \right] \equiv \phi \sum_{a=1}^{3} e^{-ix\lambda_a} M_a$$

T. Ohlsson and H. Snellman, arXiv:hep-ph/9910546 [hep-ph]

The full position dependence is contained within a scalar function

$$T = \bar{H} - \mathrm{Tr}(\bar{H})\mathbb{1}/3 \qquad \phi = e^{-ix\frac{\mathrm{Tr}(\bar{H})}{3}}$$

λ are the roots of the characteristic equation

$$\lambda^3 + c_1 \lambda + c_0 = 0$$

$$\begin{aligned} c_1 &= T_{11}T_{22} - T_{12}T_{21} + T_{11}T_{33} - T_{13}T_{31} + T_{22}T_{33} - T_{23}T_{32} \\ c_0 &= -\det T. \end{aligned}$$

Analytic evolutor expression

$$\tilde{\lambda}_a = \lambda_a + \mathrm{Tr}(\bar{H})/3$$

$$\begin{aligned}
\mathcal{U}^{(1)}(x_2,x_1) &= -i\int_{x_1}^{x_2} \mathrm{d}x\ \bar{\mathcal{U}}(x_2,x)\ \delta\tilde{H}(x)\ \bar{\mathcal{U}}(x,x_1) \\
&= -i\sum_{a,b=1}^{3}\int_{x_1}^{x_2} \mathrm{d}x e^{-i\tilde{\lambda}_a(x_2-x)} M_a \mathrm{diag}\left(\sqrt{2}G_F\delta n(x),0,0\right) M_b e^{-i\tilde{\lambda}_b(x-x_1)} \\
&= -i\sum_{a,b=1}^{3} M_a \mathrm{diag}\left(\sqrt{2}G_F I_{ab}(x_2,x_1),0,0\right) M_b,
\end{aligned}$$

$$\boxed{I_{ab}(x_2,x_1) = \int_{x_1}^{x_2} \mathrm{d}x\ e^{-i\tilde{\lambda}_a(x_2-x)}\ \delta n(x)\ e^{-i\tilde{\lambda}_b(x-x_1)}}$$

$$\delta n(x) = \tilde{\alpha}' + \beta' x^2 + \gamma' x^4$$

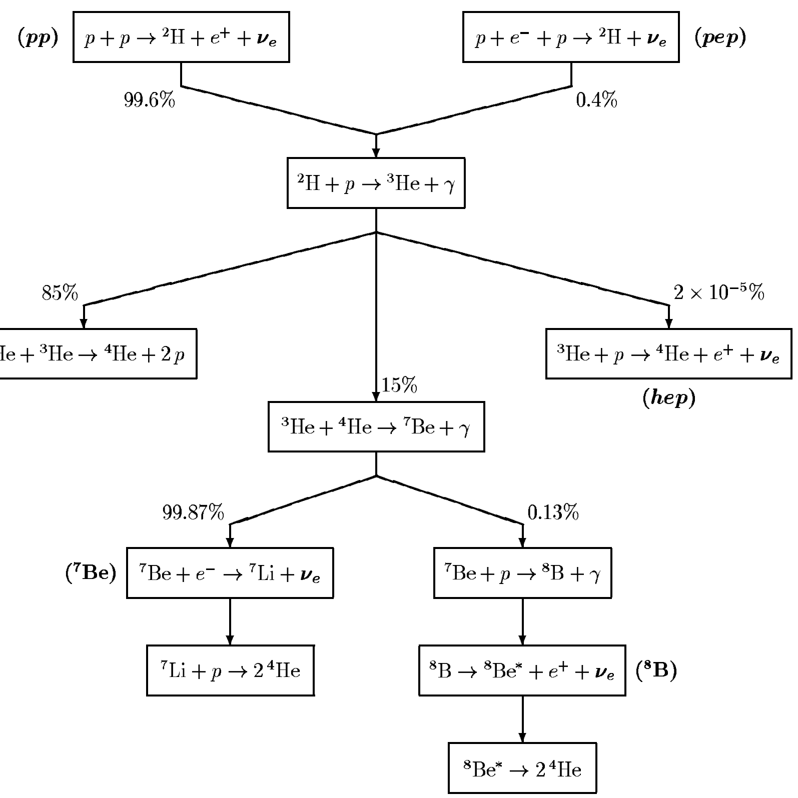
These integrals can be solved analytically within each shell

The final evolutor along a neutrino path is the time-ordered product of single-shell evolutors

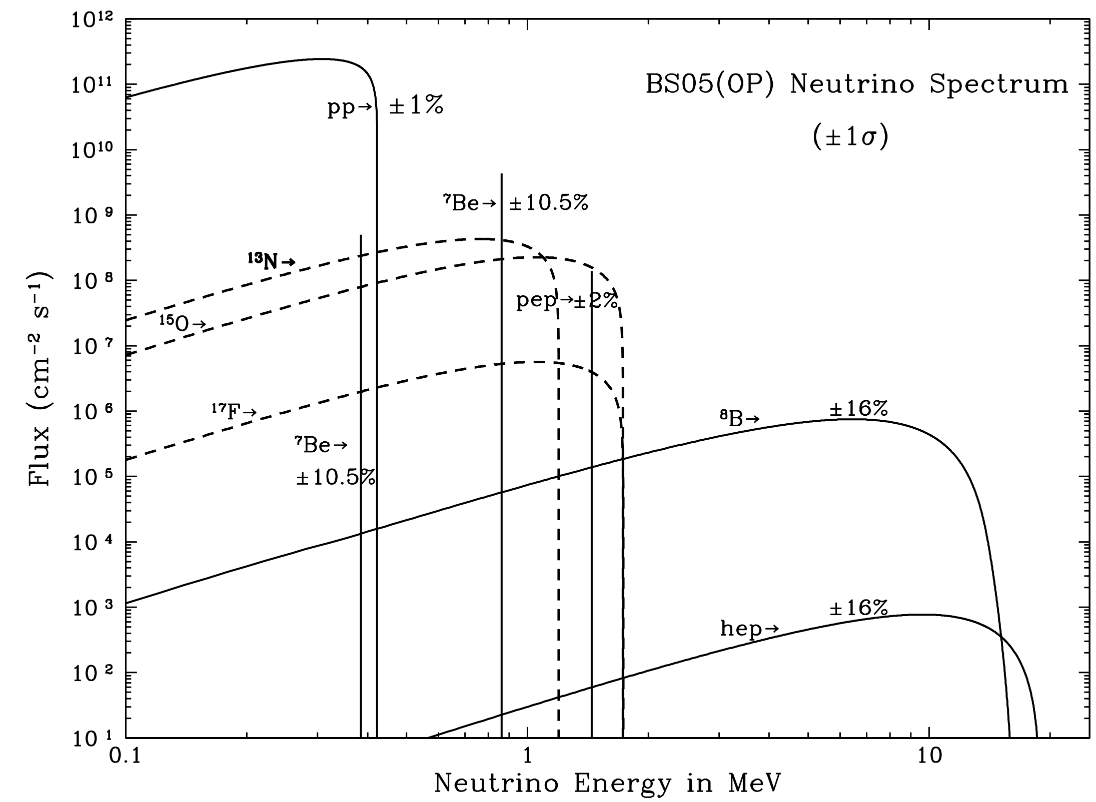
$$\mathcal{U}(x_2,x_1) = \mathcal{U}(x_2,x_i)\mathcal{U}(x_i,x_1)$$

Solar neutrinos

Solar neutrinos are produced via different reactions

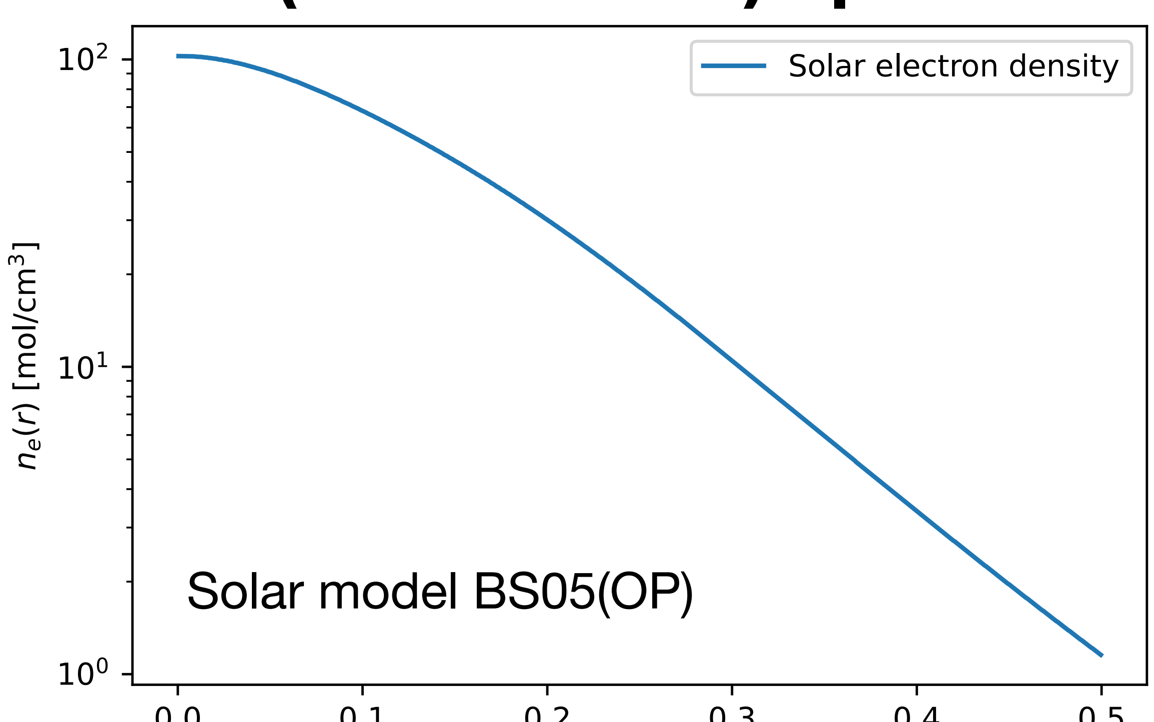


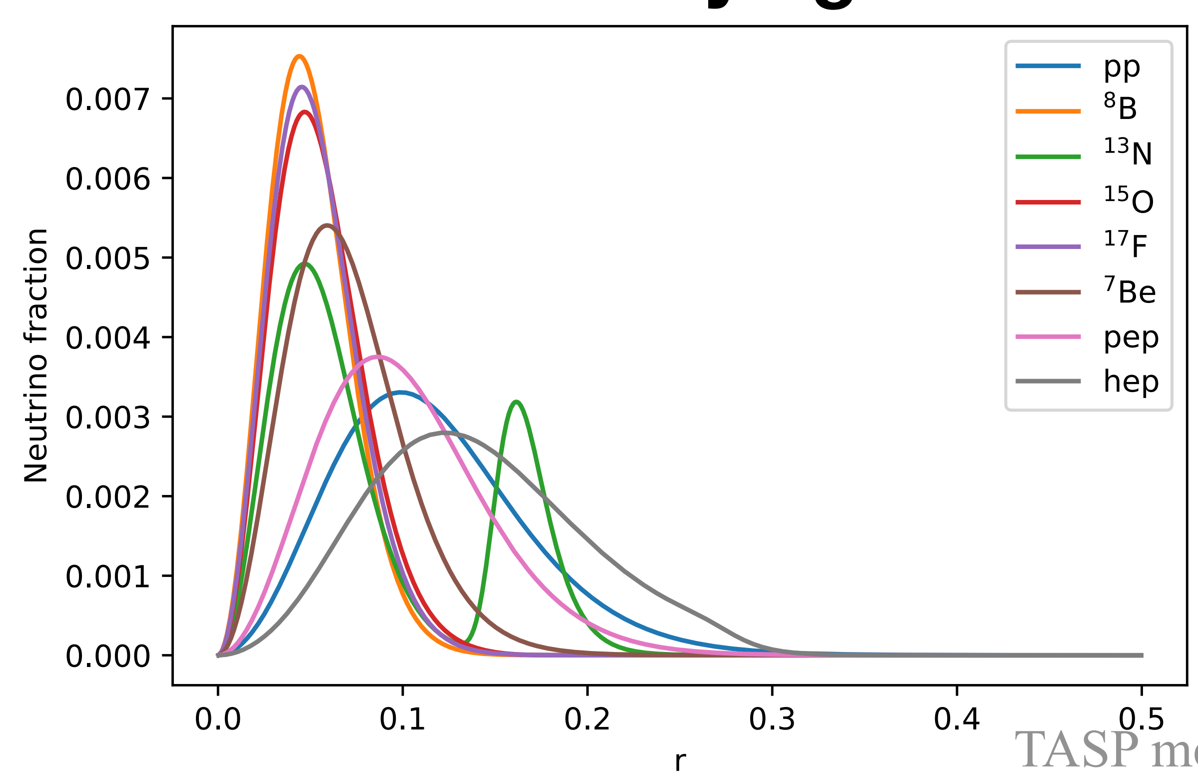
P. Lipari, 2001 CERN-CLAF School proceedings



J. N. Bahcall, A. M. Serenelli and S. Basu, arXiv:astro-ph/0412440 [astro-ph]

The exact (unoscillated) spectrum depends on the underlying solar model





Time integration and exposure

Solar neutrino experiments typically collect data over a finite interval of time

$$\langle P_E \rangle = \frac{\int_{\tau_{d_1}}^{\tau_{d_2}} \mathrm{d}\tau_d \int_{\tau_{h_1}(\tau_d)}^{\tau_{h_2}(\tau_d)} \mathrm{d}\tau_h P_E\left(\eta(\tau_d, \tau_h)\right)}{\int_{\tau_{d_1}}^{\tau_{d_2}} \mathrm{d}\tau_d \int_{\tau_{h_1}(\tau_d)}^{\tau_{h_2}(\tau_d)} \mathrm{d}\tau_h}$$

It is more effective to transform the double integral into a single one over η

$$\begin{aligned} \int_{\tau_{d_1}}^{\tau_{d_2}} \mathrm{d}\tau_d \int_{\tau_{h_1}(\tau_d)}^{\tau_{h_2}(\tau_d)} \mathrm{d}\tau_h P_E\left(\eta(\tau_d, \tau_h)\right) &= \int_{\tau_{d_1}}^{\tau_{d_2}} \mathrm{d}\tau_d \int_0^{\pi} \mathrm{d}\eta \frac{\mathrm{d}\tau_h(\tau_d, \eta)}{\mathrm{d}\eta} P_E\left(\eta\right) \\ = \int_0^{\pi} \mathrm{d}\eta P_E\left(\eta\right) \int_{\tau_{d_1}}^{\tau_{d_2}} \mathrm{d}\tau_d \frac{\mathrm{d}\tau_h(\tau_d, \eta)}{\mathrm{d}\eta} &= \int_0^{\pi} \mathrm{d}\eta P_E(\eta) W(\eta). \end{aligned}$$

E. Lisi and D. Montanino, hep-ph/9702343

$$\boxed{\langle P_E \rangle = \int_0^{\pi} \mathrm{d}\eta W(\eta) P_E(\eta)}$$

PEANUTS: Propagation and Evolution of Active NeUTrinoS

PEANUTS is an open-source Python package for the automatic computation of solar neutrino spectra and active neutrino propagation through Earth

Designed to be:

FAST: it employs analytic formulae for the neutrino propagation through varying matter density

FLEXIBLE: user can freely specify

- **Solar model**
- **Earth density profile**
- **Detector location and underground depth**

It provides functionalities for a fully automated simulation of solar neutrino fluxes at detector, and access to individual routines for more specialised computations

Example of use: simple mode

To rapidly evaluate a single point in the terminal

```
run_prob_earth.py [options] -f/-m <state> <energy> <eta> <depth> [<th12> <th13> <th23> <delta> <dm21> <dm31>]
```

```
(peanuts) ➜  PEANUTS git:(main) python run_prob_earth.py -f 0.1+0.03j,0.6+0.05j,0.09-0.9j 10 0 2e3 5.83638e-01 1.49575e-01 8.5521e-01 3.40339 7.42e-05 2.51e-03

Propagation and Evolution of Active NeUTrinoS (PEANUTS)
================================

Created by:
   Michele Lucente (michele.lucente@unibo.it)
   Tomas Gonzalo   (tomas.gonzalo@kit.edu)

PEANUTS v1.2 is open source and under the terms of the GPL-3 license.

Documentation and details for PEANUTS can be found at
T. Gonzalo and M. Lucente, arXiv:2303.15527

==========================================



Computing the probability on Earth with values

Neutrino state           : [(0.1+0.03j), (0.6+0.05j), (0.09-0.9j)]
Basis                    : flavour
theta_{12}               : 0.583638
theta_{13}               : 0.149575
theta_{23}               : 0.85521
delta_CP                 : 3.40339
Delta m_{21}^2           : 7.42e-05 eV^2
Delta m_{31}^2           : 0.00251 eV^2
Energy                   : 10.0 MeV
Nadir angle              : 0.0 rad
Depth                    : 2000.0 m
Evolution method         : analytical

Running PEANUTS...

Probability to oscillate to an electron neutrino :  0.09678794095824175
Probability to oscillate to a muon neutrino      :  0.8415536323168107
Probability to oscillate to a tau neutrino       :  0.25316790164687186
```

Example of use: expert mode

To perform scripted computations using YAML configuration files

```
run_peanuts.py -f <my_yaml_file>
```

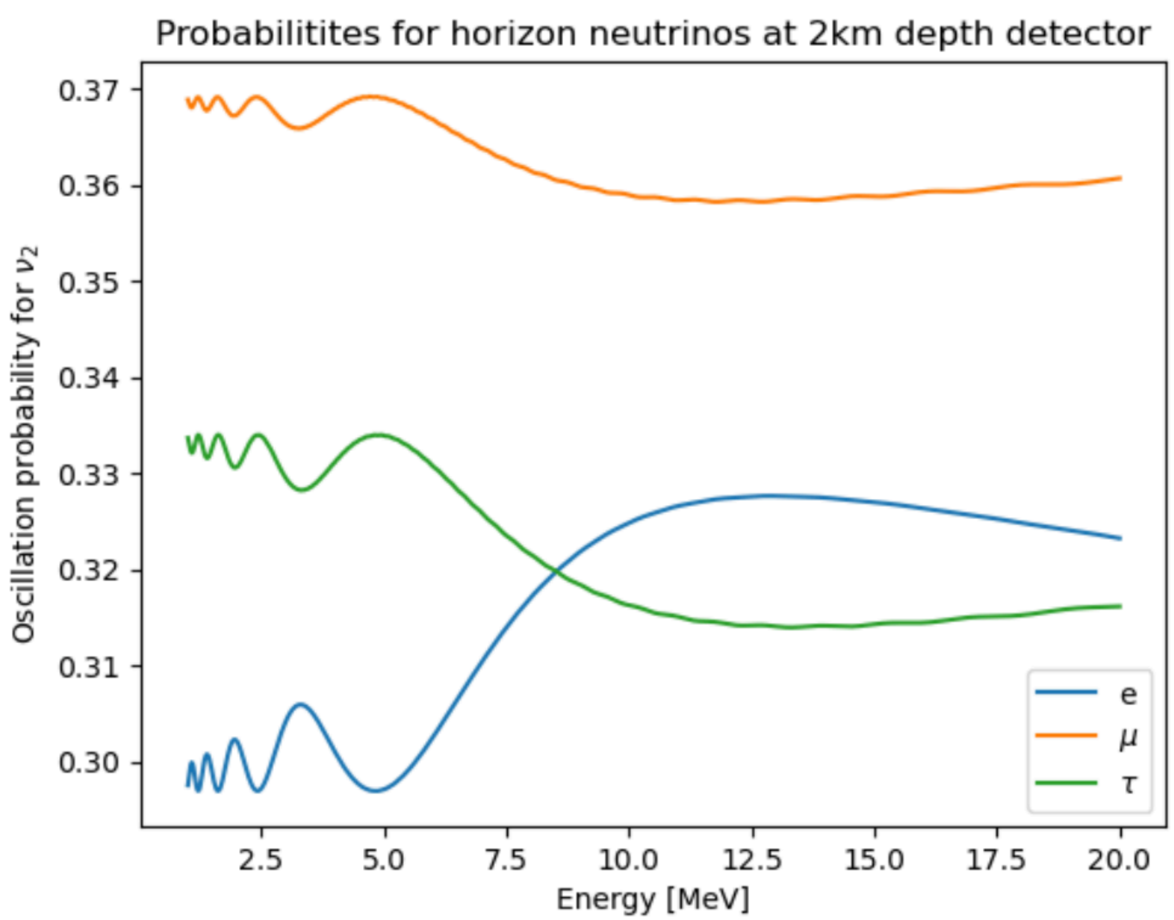
```yaml
Energy: 10

Neutrinos:
  dm21: 7.42e-05
  dm3l: 2.51e-03
  theta12: 5.83638e-01
  theta23: 8.5521e-01
  theta13: 1.49575e-01
  delta: 3.40339

Solar:

  fraction: "8B"

Earth:

  eta: [1.5708, 3.14159, 0.01]
  depth: 2000

Output: "Output/horizon_underground.dat"
```

Example of use: flexible mode

The user can also access individual methods, to employ in a personal code

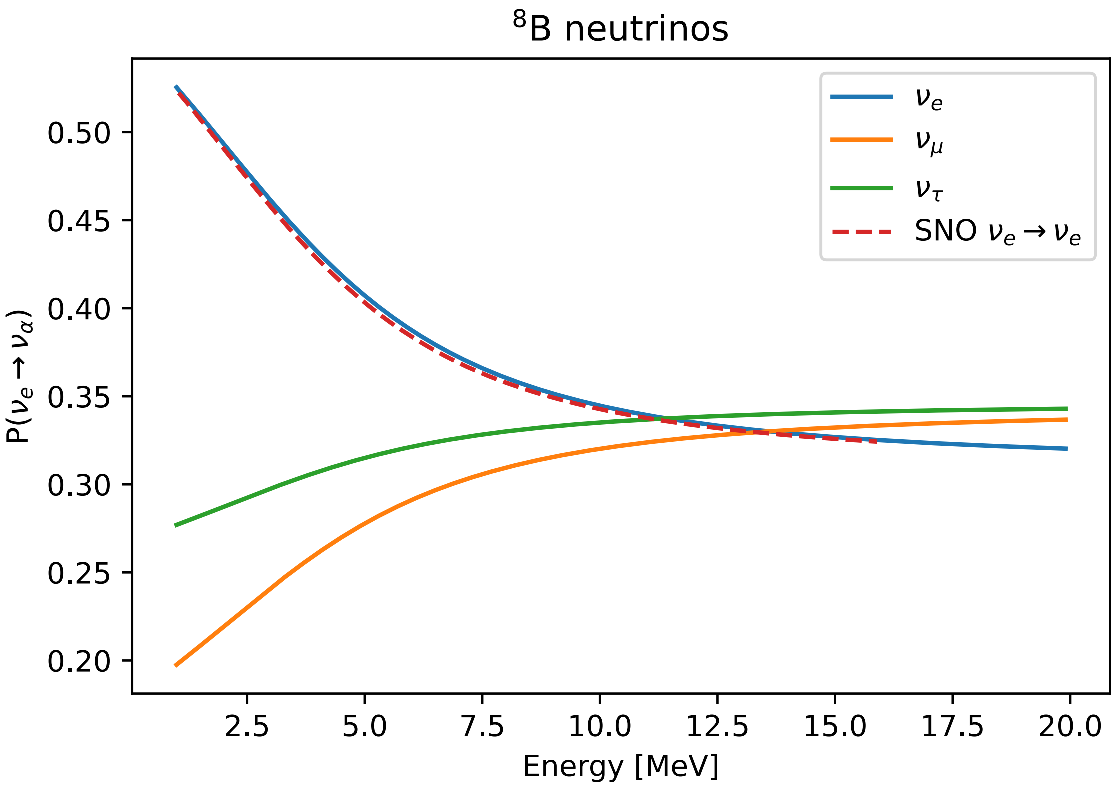
```
from peanuts.earth import Pearth

x = np.linspace(1, 20, 1000)

evolved_state = np.array([Pearth(nustate, earth_density, pmns, DeltamSq21, DeltamSq3l, E, eta, depth, massbasis=False, antinu=False) for E in
```

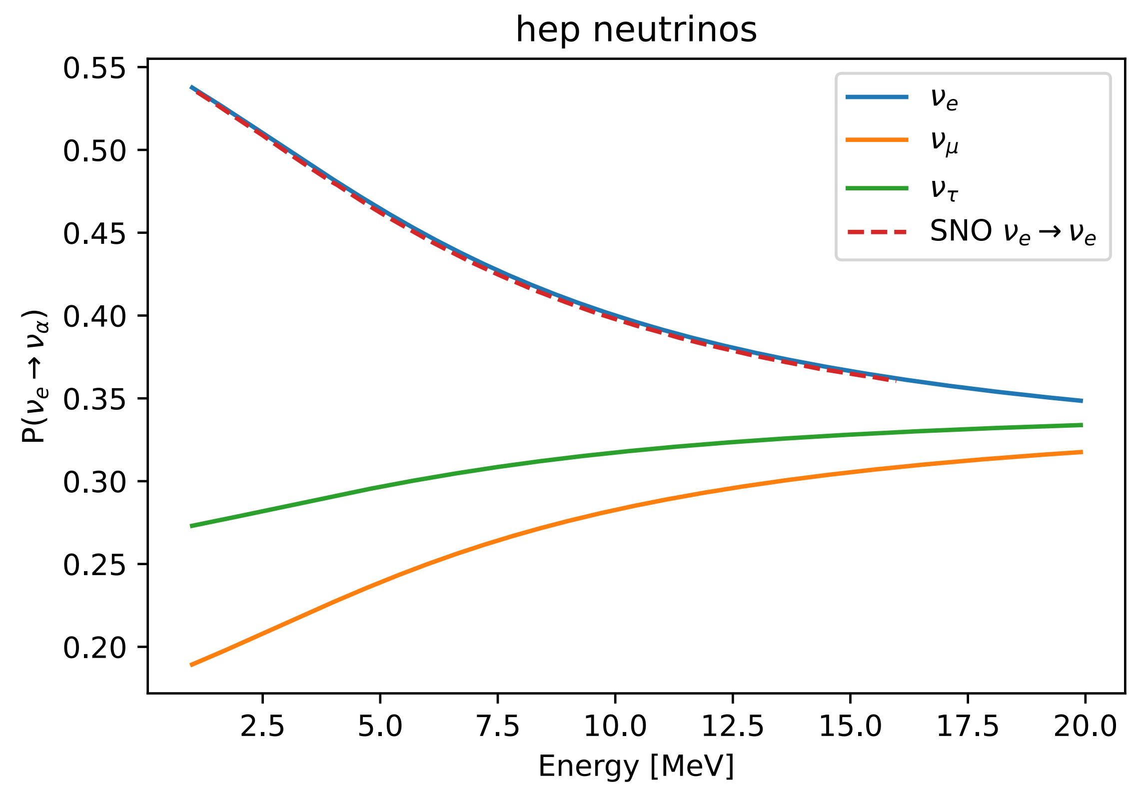
```
import matplotlib.pyplot as plt

plt.plot(x, evolved_state)

plt.xlabel("Energy [MeV]")
plt.ylabel(r"Oscillation probability for $\nu_2$")

plt.title("Probabilitites for horizon neutrinos at 2km depth detector")

plt.legend([r'e', r'$\mu$', r'$\tau$'])

plt.show()
```



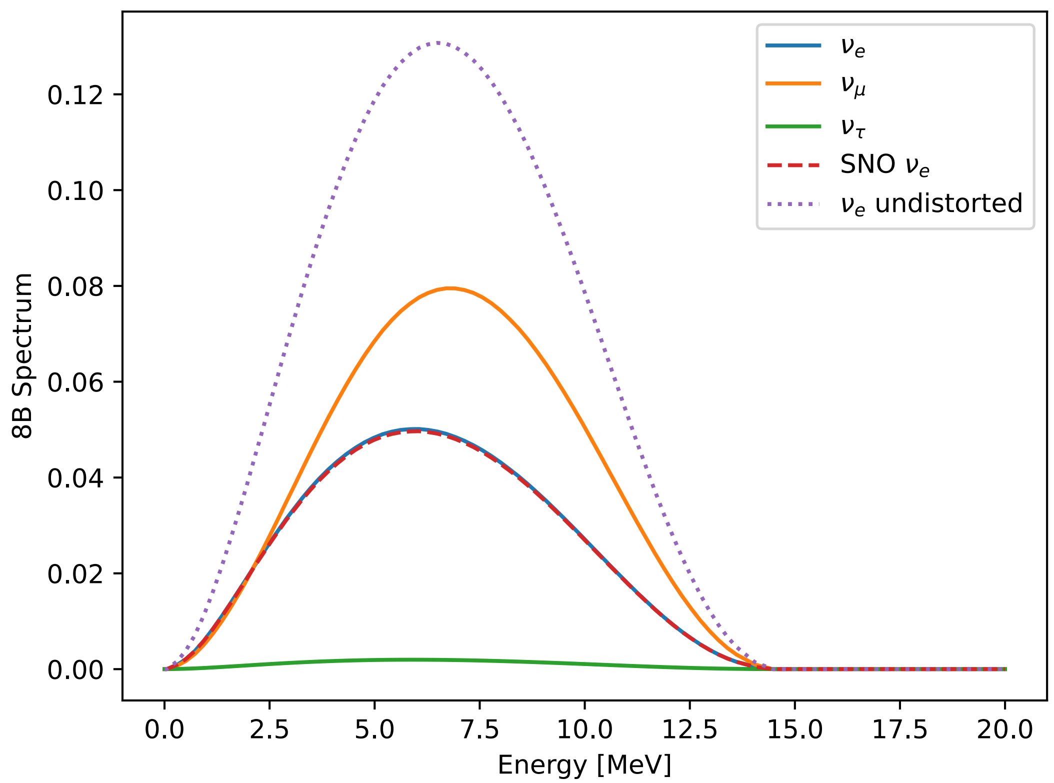
Validation: solar spectra

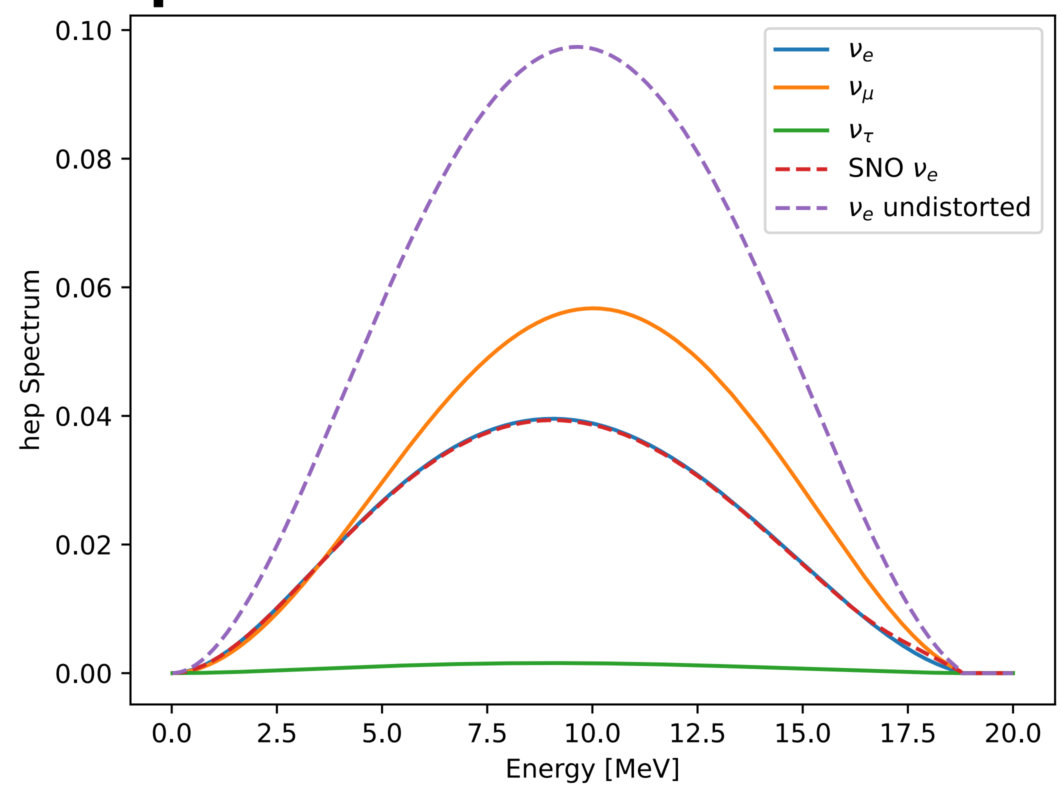
Survival probability at the surface of the Sun





Distorted neutrino spectra

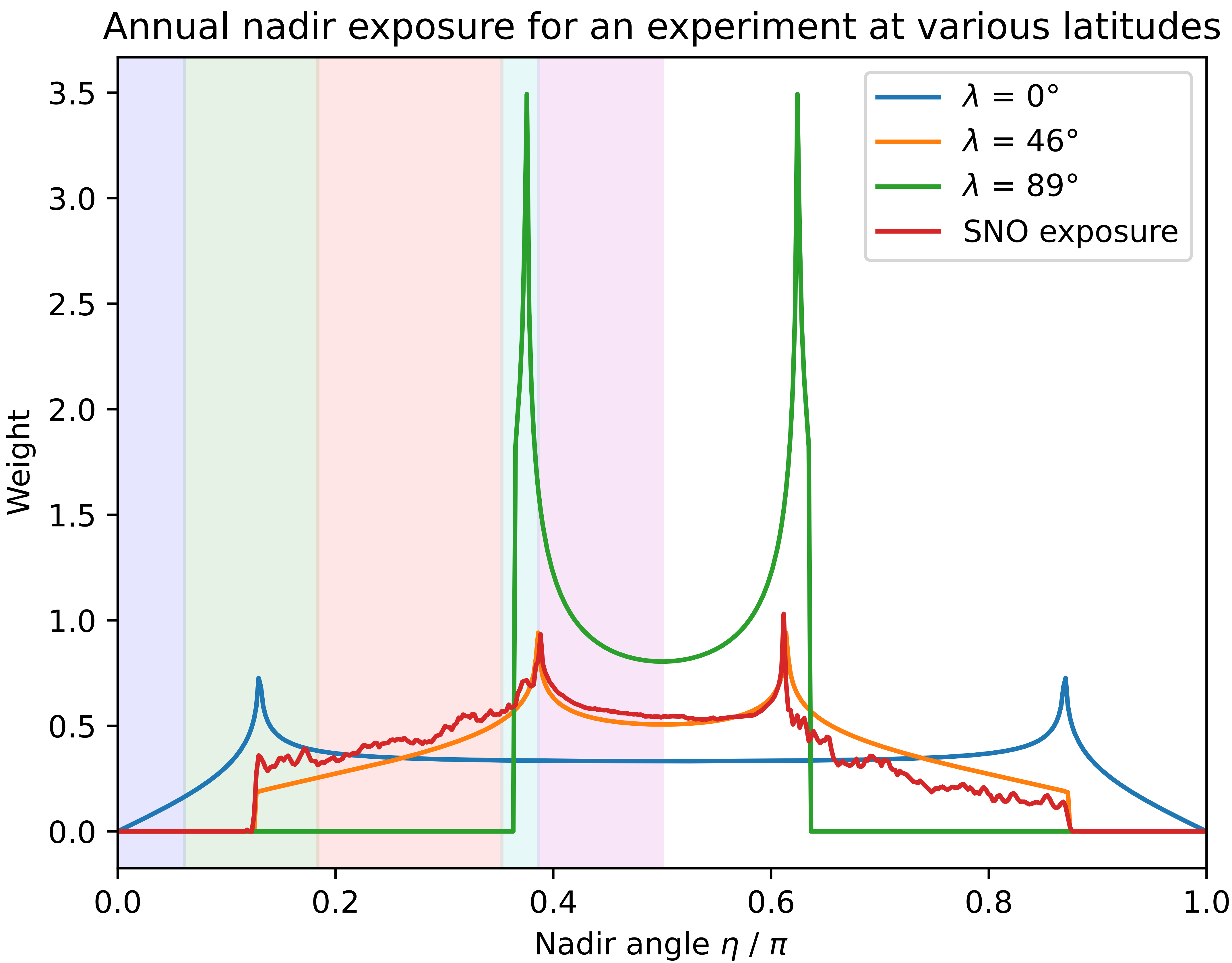




Comparison with SNO predictions

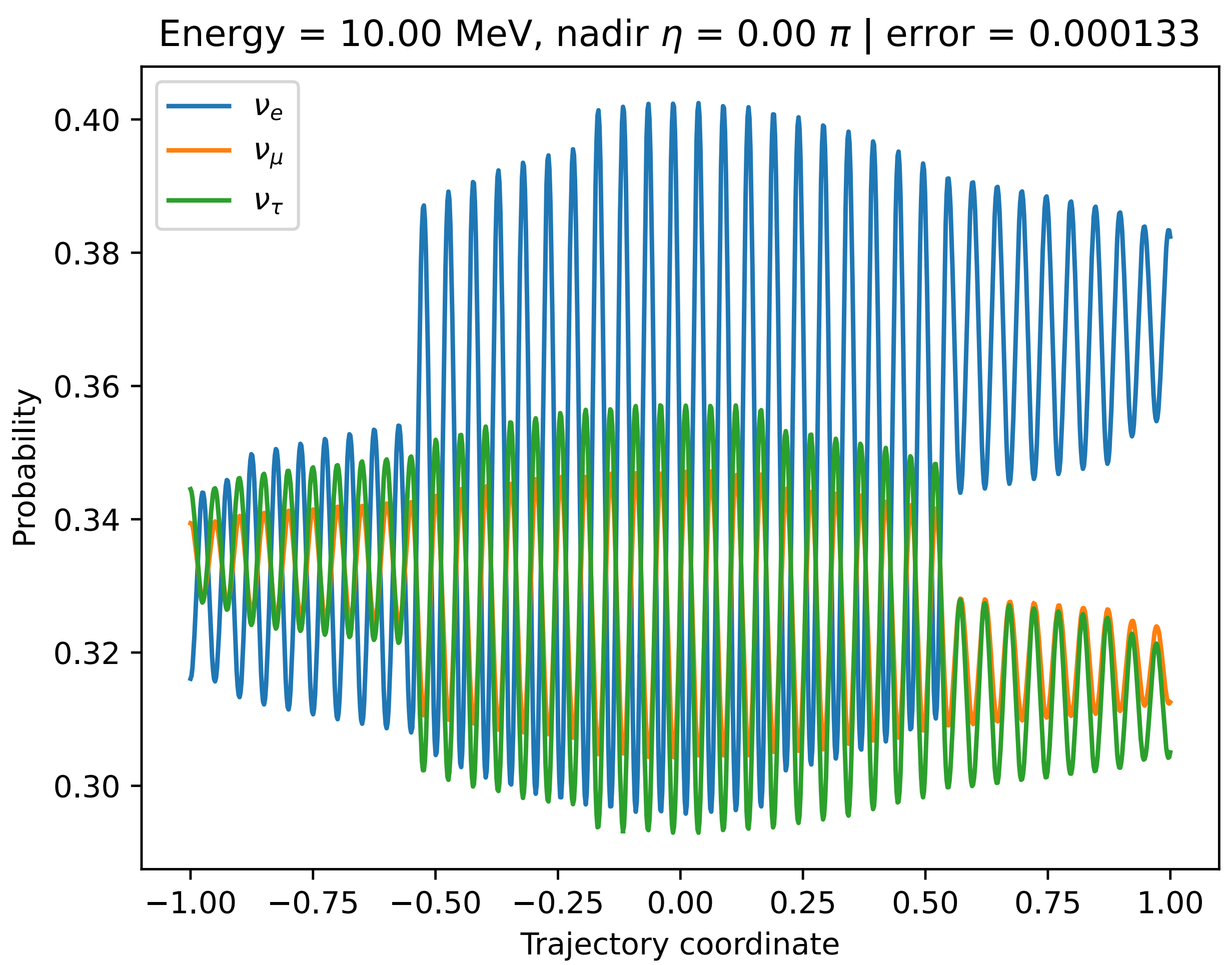
N.F. Fiúza de Barros, Ph.D. thesis

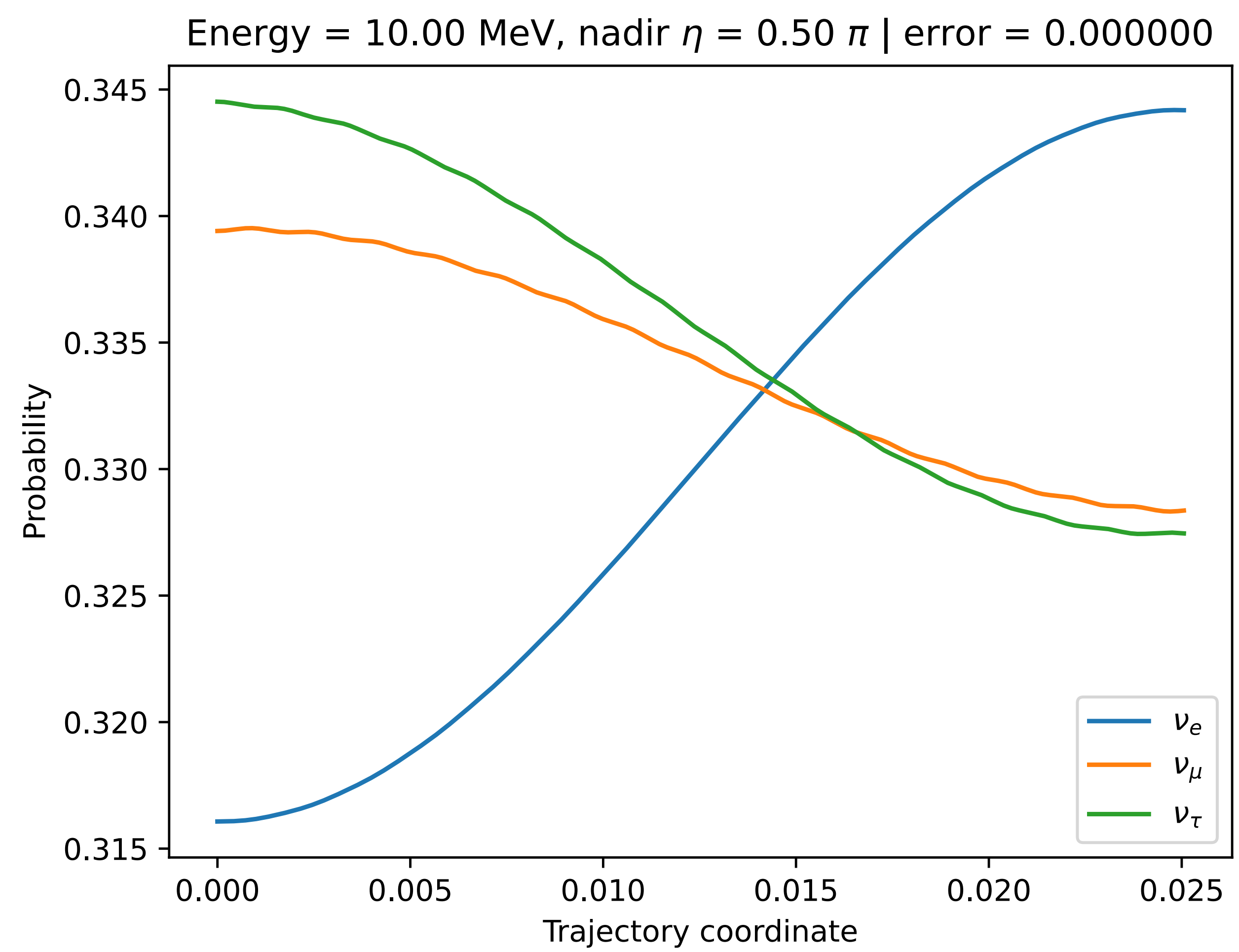
Validation: nadir angle exposure



Validation: numerical vs analytical

We also provide a numerical routine, to cross-check analytical results and thoroughly explore single-point dynamics



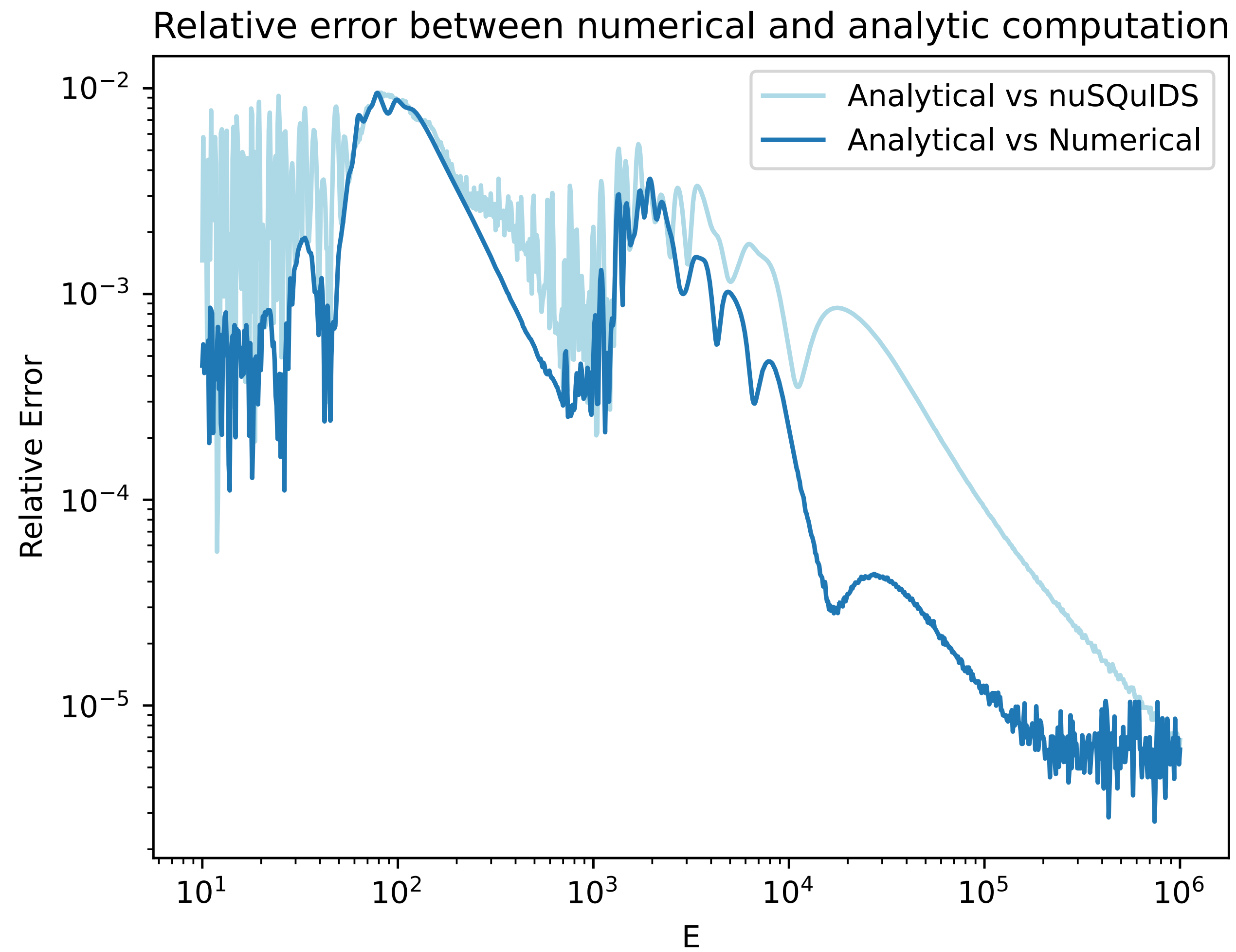


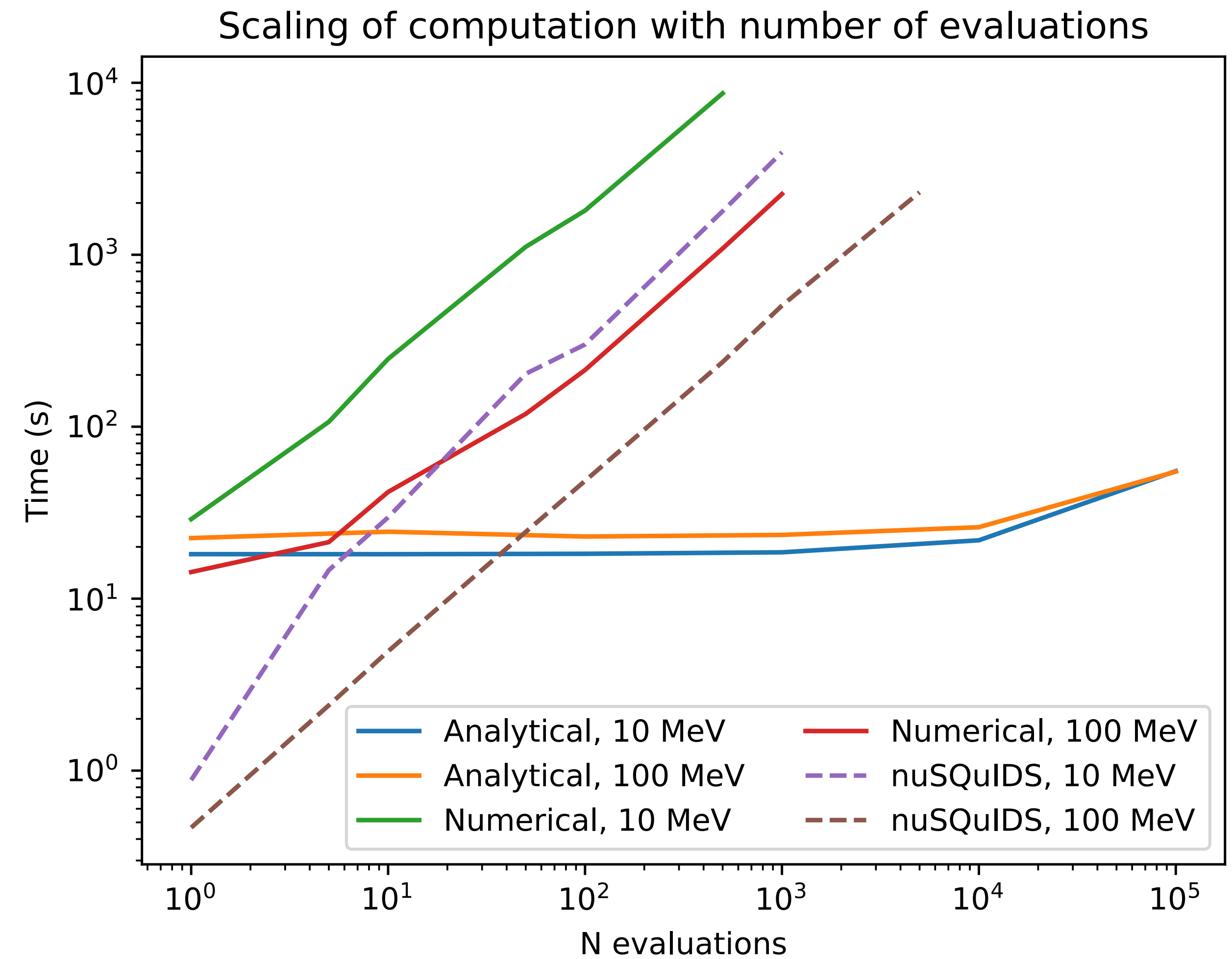
The relative error on the complex coefficients is always smaller than 10^{-2}

Comparison with other public codes

We compare PEANUTS with nuSQuIDS, a state-of-the-art C++ software to compute neutrino oscillations

C. A. Argüelles, J. Salvado and C. N. Weaver, arXiv:2112.13804 [hep-ph]





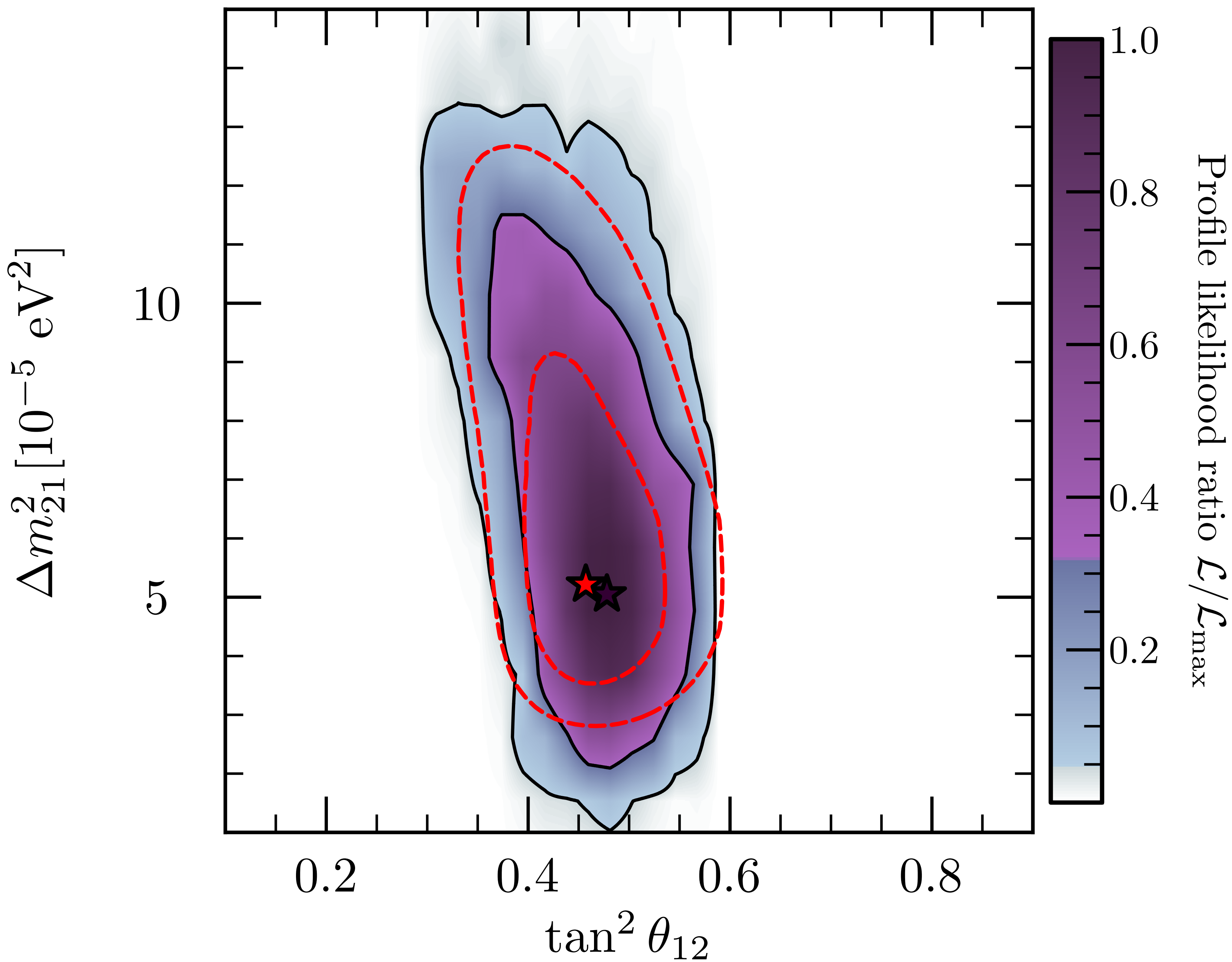
We observe excellent agreement of results over a wide range of energies

PEANUTS is orders of magnitude faster than numerical algorithms for more than ~ 10 evaluations

Validation: solar neutrino fit

SNO Collaboration, arXiv:nucl-ex/0610020 [nucl-ex]

We employ PEANUTS to analyse data from Phase I of the SNO experiment



The scan sampled around 380k parameter points, each of which performed, on average, around 5k evaluations of the probability

Conclusion

We released **PEANUTS**, an **open-source Python package** for the **automatic computation** of **solar neutrino spectra** and **propagation within Earth**

PEANUTS is designed to be **fast**, by employing **analytic** formulae for the neutrino propagation through varying matter density

And **flexible**, by allowing the user to input **arbitrary solar models, custom Earth density profiles and general detector locations**

It provides **functionalities** for a **fully automated simulation** of solar neutrino fluxes at detector, and access to **individual routines** to perform more specialised computations

Extensively validated against the results of the **SNO experiment** and **nuSQuIDS public code**, providing **excellent agreement** with **significantly reduced computational time**

Currently working on including routines for **atmospheric** and **long-baseline** experiments

Backup

Quick installation guide

PEANUTS is designed to be platform-agnostic, but the user should ensure that dependencies are installed and compatible

A simple way to do this is using the **conda** and **pip** package managers

Create a virtual environment and install dependencies

```
conda create -n peanuts python numpy numba scipy pandas mpmath pyyaml matplotlib gitpython
conda activate peanuts
pip install pyinterval pyslha
```

Download PEANUTS

```
git clone https://github.com/michelelucente/PEANUTS
cd PEANUTS
```

Run it!

```
python run_peanuts.py -f examples/solar_earth_test.yaml
```

We also provide a YAML file with tested working environment

```
conda env create -f peanuts_env.yml
```

Evolutor equations

The evolutor operator is defined by the equations

$$|\nu, t\rangle = \hat{\mathcal{U}}(t, t_0)\,|\nu, t_0\rangle \qquad \hat{\mathcal{U}}(t_0, t_0) = \hat{\mathbb{1}}$$

From the Schrödinger equation

$$i\frac{\mathrm{d}}{\mathrm{d}t}\,|\nu, t\rangle = \hat{H}(t)\,|\nu, t\rangle \Rightarrow i\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathcal{U}}(t, t_0)\,|\nu, t_0\rangle = \hat{H}(t)\hat{\mathcal{U}}(t, t_0)\,|\nu, t_0\rangle$$

$$i\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathcal{U}}(t, t_0) = \hat{H}(t)\hat{\mathcal{U}}(t, t_0)$$

The formal solution is

$$\boxed{\mathcal{U}(t, t_0) = \mathcal{T}\left[e^{-i\int_{t_0}^{t}\mathrm{d}t'\,H(t')}\right]}$$

An exact closed form is only known in very specific scenarios (e.g. if [H(t), H(t')] = 0)

Solar neutrino flux

Solar neutrinos form an incoherent flux of mass (propagation) eigenstates

In the adiabatic regime

$$P^{\odot}_{\nu_e \to \nu_i}(E) = \int_0^1 \mathrm{d}r \, |T_{ei}(E, n_e(r))|^2 \, f(r) \qquad \int_0^1 \mathrm{d}r f(r) = 1$$

Probability for electron neutrino of energy E produced at point with matter density $n_e(r)$ to propagate as mass eigenstate i

Production point distribution function

Propagation is coherent when a neutrino crosses the Earth

$$P^{SE}_{\alpha}(t, E) = |\mathcal{U}_{\alpha\beta}(t, t_0) U_{\beta i}|^2 \, P^{\odot}_{\nu_e \to \nu_i}(E)$$

weight of the mass eigenstate i in the incoherent solar flux

$$\mathcal{U}_{\alpha\beta}(t, t_0) U_{\beta i} = \langle \nu_\alpha | \nu_i, t \rangle$$

transition amplitude from the evolved mass eigenstate i to interaction eigenstate α