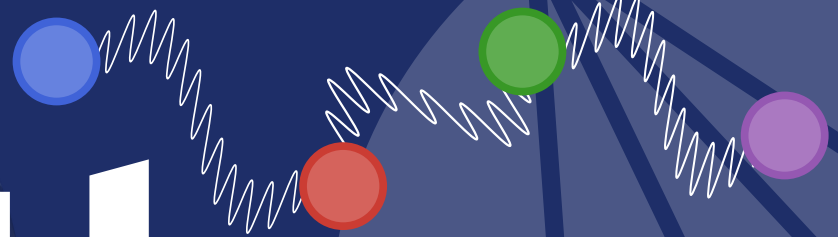
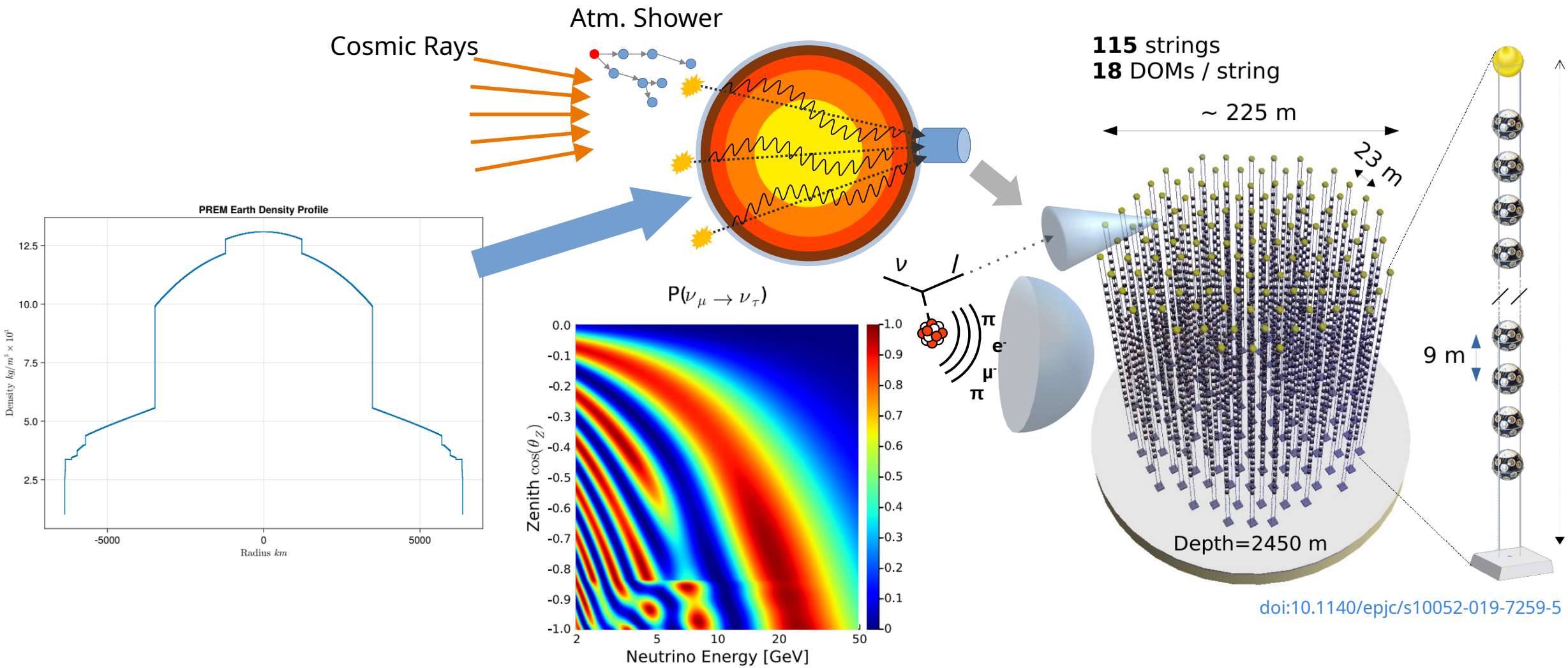
Friedrich-Alexander-Universität
Erlangen-Nürnberg



Neurthino

Johannes Schumann
JuliaHEP, 09 Nov 2023

Neutrino Oscillations



doi:10.1140/epjc/s10052-019-7259-5

Neutrino Oscillations – The math fun

How it always starts: $i\partial_t |\psi\rangle = H |\psi\rangle$

PMNS matrix: mass basis ↔ flavour basis

$$U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{23} & \sin\theta_{23} \\ 0 & -\sin\theta_{23} & \cos\theta_{23} \end{pmatrix} \begin{pmatrix} \cos\theta_{13} & 0 & e^{i\delta}\sin\theta_{13} \\ 0 & 1 & 0 \\ -e^{i\delta}\sin\theta_{13} & 0 & \cos\theta_{13} \end{pmatrix} \begin{pmatrix} \cos\theta_{12} & \sin\theta_{12} & 0 \\ -\sin\theta_{12} & \cos\theta_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Transition probability:

$$P_{\nu_\alpha \to \nu_\beta}(L) = \delta_{\alpha\beta} - 4\sum_{k,l} \mathrm{Re}\left[U^*_{\alpha k} U_{\beta k} U_{\alpha l} U^*_{\beta l}\right] \sin^2\left(\frac{\Delta m^2_{kl} L}{4E_\nu}\right)$$

$$\pm 2\sum_{k,l} \mathrm{Im}\left[U^*_{\alpha k} U_{\beta k} U_{\alpha l} U^*\right]_{\beta l} \sin^2\left(\frac{\Delta m^2_{kl} L}{2E_\nu}\right)$$

α,β → Flavour basis
K,l → Mass basis

Neutrino Oscillations – The math fun

Ok and what's the problem now … seem pretty straight forward

Neutrino Oscillations – The math fun

Matter Hamiltonian:

$$H^m_{\alpha\beta} = H^0_{\alpha\beta} + \delta_{ee} A$$

Mhmmmmmmm

... which leads to new PMNS for every electron density

$$H^m_{ij} = \tilde{U} H^m_{\alpha\beta} \tilde{U}^\dagger$$

so depending on the resolution of matter composition:
→ many eigenvalues to find
→ ... and matrix multiplications to do

Feasible problem, but annoying problem:
You want a package to do it for you

What you need to know about

Neurthino

- Neutrino oscillation probability calculator
  → n-Flavour modelling
- Includes **P**reliminary **R**eference **E**arth **M**odel

GitHub

src	Correct matter effects for sterile flavours; handling of anti neutrinos	2 years ago
test	Adding 4 flavour matter effects unit tests	2 years ago
.gitignore	File setup by DrWatson	4 years ago
.gitlab-ci.yml	Use the proper Julia image for GitLab CI	2 years ago
.travis.yml	Update versions for Travis CI	2 years ago
LICENSE	Add license	2 years ago
Project.toml	Use 0.8 & 0.9 from DocStringExtensions	7 months ago
README.md	Fix README for GitLab	2 years ago
codemeta.json	Add reference publication	2 years ago

MIT license
Activity
7 stars
6 watching
3 forks
Report repository

Releases 2
v1.0.1 Latest
on Sep 16, 2021
+ 1 release

Packages
No packages published

Contributors 3
8me Johannes Schumann
tamasgal Tamas Gal
philippeller Philipp Eller

Languages
Julia 90.0% Jupyter Notebook 10.0%

README.md

Neurthino

docs stable | docs dev | CI passing | codecov 89% | DOI 10.5281/zenodo.5512622

Neurthino.jl

Neurthino.jl is a package for calculating neutrino oscillation probabilities. The main focus of the package lies on atmospheric neutrino flux and the neutrino propagation through earth.

Basic Usage

First of all the basic vacuum properties have to be defined by creating a `OscillationParameters` struct with fixed number of neutrino flavours of the considered model:

```
julia> using Neurthino

julia> osc = OscillationParameters(3);
```

Example

Set Oscillation Parameters (e.g. from NuFIT):

```
julia> using Neurthino

julia> osc = OscillationParameters(3);

julia> setθ!(osc, 1⇒2, 0.59);

julia> setθ!(osc, 1⇒3, 0.15);

julia> setθ!(osc, 2⇒3, 0.84);

julia> setδ!(osc, 1⇒3, 3.86);

julia> setΔm²!(osc, 2⇒3, -2.511e-3);

julia> setΔm²!(osc, 1⇒2, -7.41e-5);
```

NuFIT 5.2 (2022)

		Normal Ordering (best fit)		Inverted Ordering ($\Delta\chi^2 = 2.3$)	
		bfp $\pm 1\sigma$	3σ range	bfp $\pm 1\sigma$	3σ range
without SK atmospheric data	$\sin^2\theta_{12}$	$0.303^{+0.012}_{-0.011}$	$0.270 \to 0.341$	$0.303^{+0.012}_{-0.011}$	$0.270 \to 0.341$
	$\theta_{12}/^\circ$	$33.41^{+0.75}_{-0.72}$	$31.31 \to 35.74$	$33.41^{+0.75}_{-0.72}$	$31.31 \to 35.74$
	$\sin^2\theta_{23}$	$0.572^{+0.018}_{-0.023}$	$0.406 \to 0.620$	$0.578^{+0.016}_{-0.021}$	$0.412 \to 0.623$
	$\theta_{23}/^\circ$	$49.1^{+1.0}_{-1.3}$	$39.6 \to 51.9$	$49.5^{+0.9}_{-1.2}$	$39.9 \to 52.1$
	$\sin^2\theta_{13}$	$0.02203^{+0.00056}_{-0.00059}$	$0.02029 \to 0.02391$	$0.02219^{+0.00060}_{-0.00057}$	$0.02047 \to 0.02396$
	$\theta_{13}/^\circ$	$8.54^{+0.11}_{-0.12}$	$8.19 \to 8.89$	$8.57^{+0.12}_{-0.11}$	$8.23 \to 8.90$
	$\delta_{CP}/^\circ$	197^{+42}_{-25}	$108 \to 404$	286^{+27}_{-32}	$192 \to 360$
	$\frac{\Delta m^2_{21}}{10^{-5}\ \mathrm{eV}^2}$	$7.41^{+0.21}_{-0.20}$	$6.82 \to 8.03$	$7.41^{+0.21}_{-0.20}$	$6.82 \to 8.03$
	$\frac{\Delta m^2_{3\ell}}{10^{-3}\ \mathrm{eV}^2}$	$+2.511^{+0.028}_{-0.027}$	$+2.428 \to +2.597$	$-2.498^{+0.032}_{-0.025}$	$-2.581 \to -2.408$
		Normal Ordering (best fit)		Inverted Ordering ($\Delta\chi^2 = 6.4$)	
		bfp $\pm 1\sigma$	3σ range	bfp $\pm 1\sigma$	3σ range
with SK atmospheric data	$\sin^2\theta_{12}$	$0.303^{+0.012}_{-0.012}$	$0.270 \to 0.341$	$0.303^{+0.012}_{-0.011}$	$0.270 \to 0.341$
	$\theta_{12}/^\circ$	$33.41^{+0.75}_{-0.72}$	$31.31 \to 35.74$	$33.41^{+0.75}_{-0.72}$	$31.31 \to 35.74$
	$\sin^2\theta_{23}$	$0.451^{+0.019}_{-0.016}$	$0.408 \to 0.603$	$0.569^{+0.016}_{-0.021}$	$0.412 \to 0.613$
	$\theta_{23}/^\circ$	$42.2^{+1.1}_{-0.9}$	$39.7 \to 51.0$	$49.0^{+1.0}_{-1.2}$	$39.9 \to 51.5$
	$\sin^2\theta_{13}$	$0.02225^{+0.00056}_{-0.00059}$	$0.02052 \to 0.02398$	$0.02223^{+0.00058}_{-0.00058}$	$0.02048 \to 0.02416$
	$\theta_{13}/^\circ$	$8.58^{+0.11}_{-0.11}$	$8.23 \to 8.91$	$8.57^{+0.11}_{-0.11}$	$8.23 \to 8.94$
	$\delta_{CP}/^\circ$	232^{+36}_{-26}	$144 \to 350$	276^{+22}_{-29}	$194 \to 344$
	$\frac{\Delta m^2_{21}}{10^{-5}\ \mathrm{eV}^2}$	$7.41^{+0.21}_{-0.20}$	$6.82 \to 8.03$	$7.41^{+0.21}_{-0.20}$	$6.82 \to 8.03$
	$\frac{\Delta m^2_{3\ell}}{10^{-3}\ \mathrm{eV}^2}$	$+2.507^{+0.026}_{-0.027}$	$+2.427 \to +2.590$	$-2.486^{+0.025}_{-0.028}$	$-2.570 \to -2.406$

Simple Example

For a single case you are pretty much done:

```
julia> p = Pνν(osc, 1, 10000)
4-dimensional AxisArray{Float64,4,...} with axes:
    :Energy, [1.0]
    :Baseline, [10000.0]
    :InitFlav, NeutrinoFlavour[Electron, Muon, Tau]
    :FinalFlav, NeutrinoFlavour[Electron, Muon, Tau]
…
julia> p[Energy=1, Baseline=1, InitFlav=Muon, FinalFlav=Tau]
0.2592655682779296
```

The "Full" Example

First: Create PMNS matrix and Hamiltonian

```
julia> U = PMNSMatrix(osc)
3×3 Array{Complex{Float64},2}:
   0.82161+0.0im          0.550114+0.0im         -0.112505+0.0983582im
 -0.301737+0.0608595im   0.601232+0.0407488im    0.736282+0.0im
  0.476688+0.0545516im  -0.576975+0.0365253im    0.659968+0.0im

julia> H = Hamiltonian(osc)
3-element Array{Complex{Float64},1}:
 -0.0008902666666666667 + 0.0im
 -0.0008163666666666667 + 0.0im
  0.0017066333333333333 + 0.0im
```

The "Full" Example

Now one can determine the U and H for a density and energy:

```
julia> U_mat, H_mat = MatterOscillationMatrices(U, H, 1, 13);
julia> H_mat
3-element Array{Complex{Float64},1}:
 -0.00084049013185075 02 - 2.5459232191294903e-20im
   9.07812614939963 5e-5 - 1.7515135102794 3e-20im
  0.0017419062876598283 - 1.8741859435908039e-19im
julia> U_mat
3×3 Array{Complex{Float64},2}:
  0.0358018-0.000158113im  0.970863+0.0im       -0.178275+0.156083im
 -0.662778+0.00661213im    0.157174+0.116074im   0.722845+0.0im
  0.74793+0.0im            0.0917808+0.104043im  0.649115-0.00104331im
```

The "Full" Example

Second: create paths

```
julia> zenith = acos.(range(-1,stop=0,length=200));


julia> paths = Neurthino.prempath(zenith, 2.5, samples=100,
discrete_densities=0:0.1:14);
```

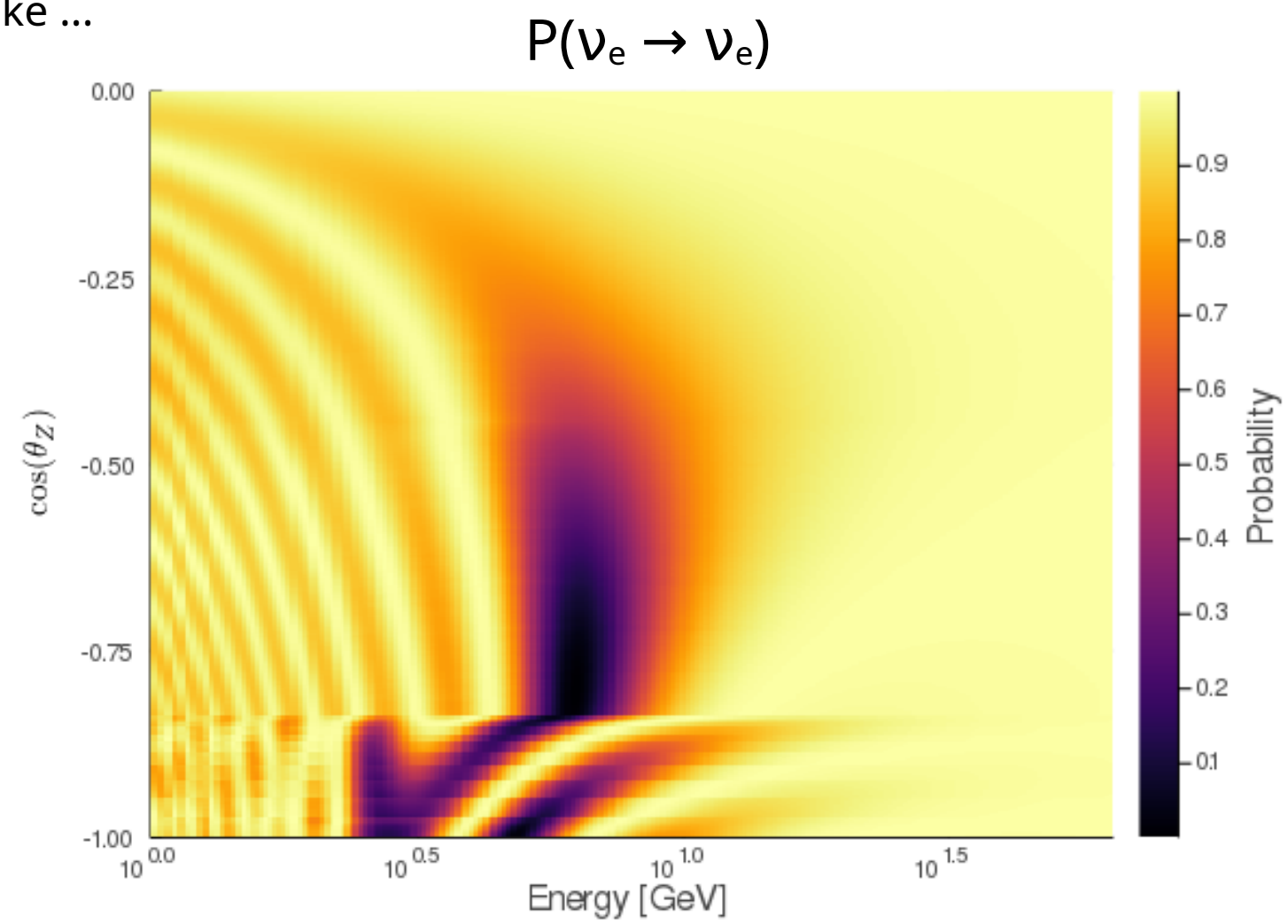
Third: Calculate probabilities

```
julia> energies = 10 .^ range(0, stop=2, length=200);


julia> prob = Pνν(U, H, energies, paths);
```

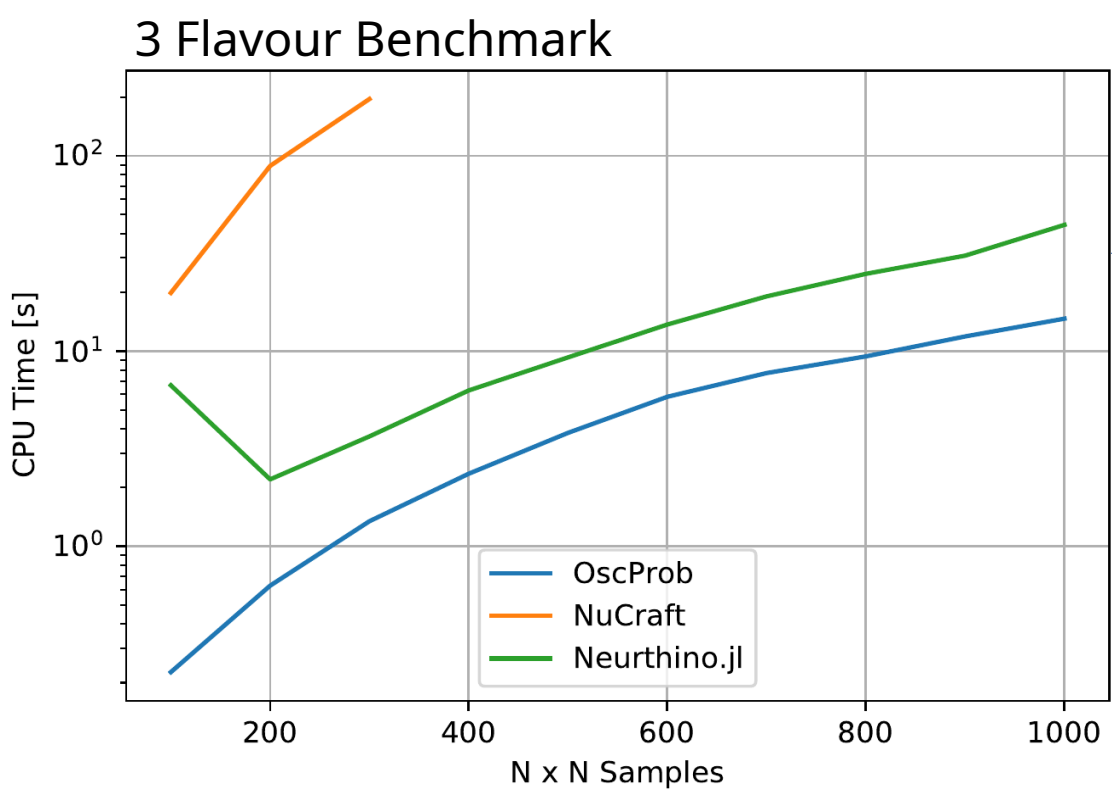
The "Full" Example

What does it look like ...



Summary

- Working package for neutrino oscillations
  → out of the box n-Flavour without any optimizations yet
  → Includes earth matter density model (PREM)
  … open to new ones if you have one ;)
  →
- Outlook / ToDo:
  → Use Liouville-von Neumann formulation
  some effects (e.g. decoherence not fully covered)
  → Interpolation of oscillation parameters in matter
  → GPU option for determining eigenvalues



Friedrich-Alexander-Universität
Erlangen-Nürnberg

ERLANGEN CENTRE
FOR ASTROPARTICLE
PHYSICS

FAU

Thank You

Backup

- Matter Oscillation Parameters Interpolation: