Neutrino Oscillation Computational Aspects

João Coelho

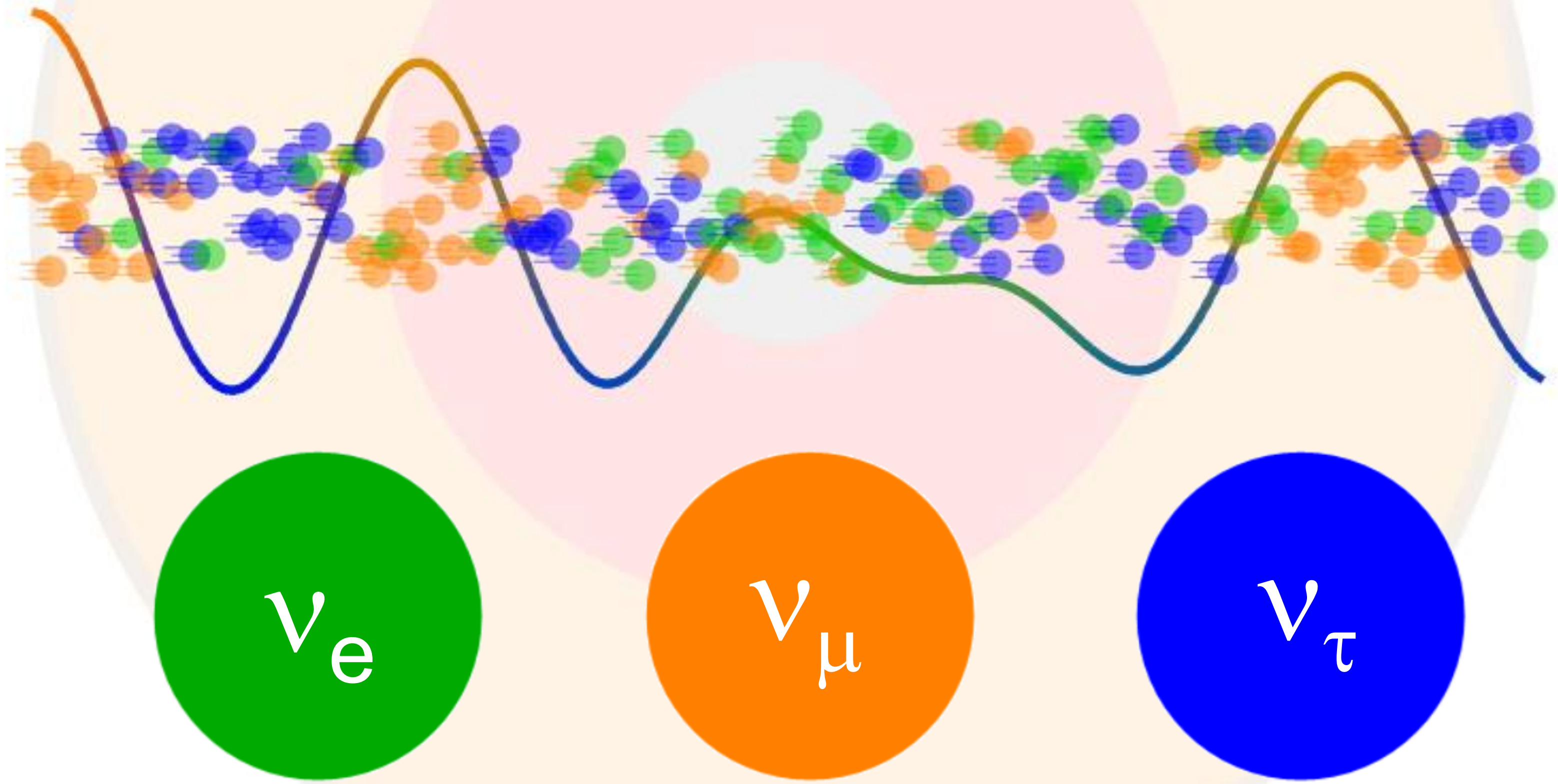
APC Laboratory

15 November 2023

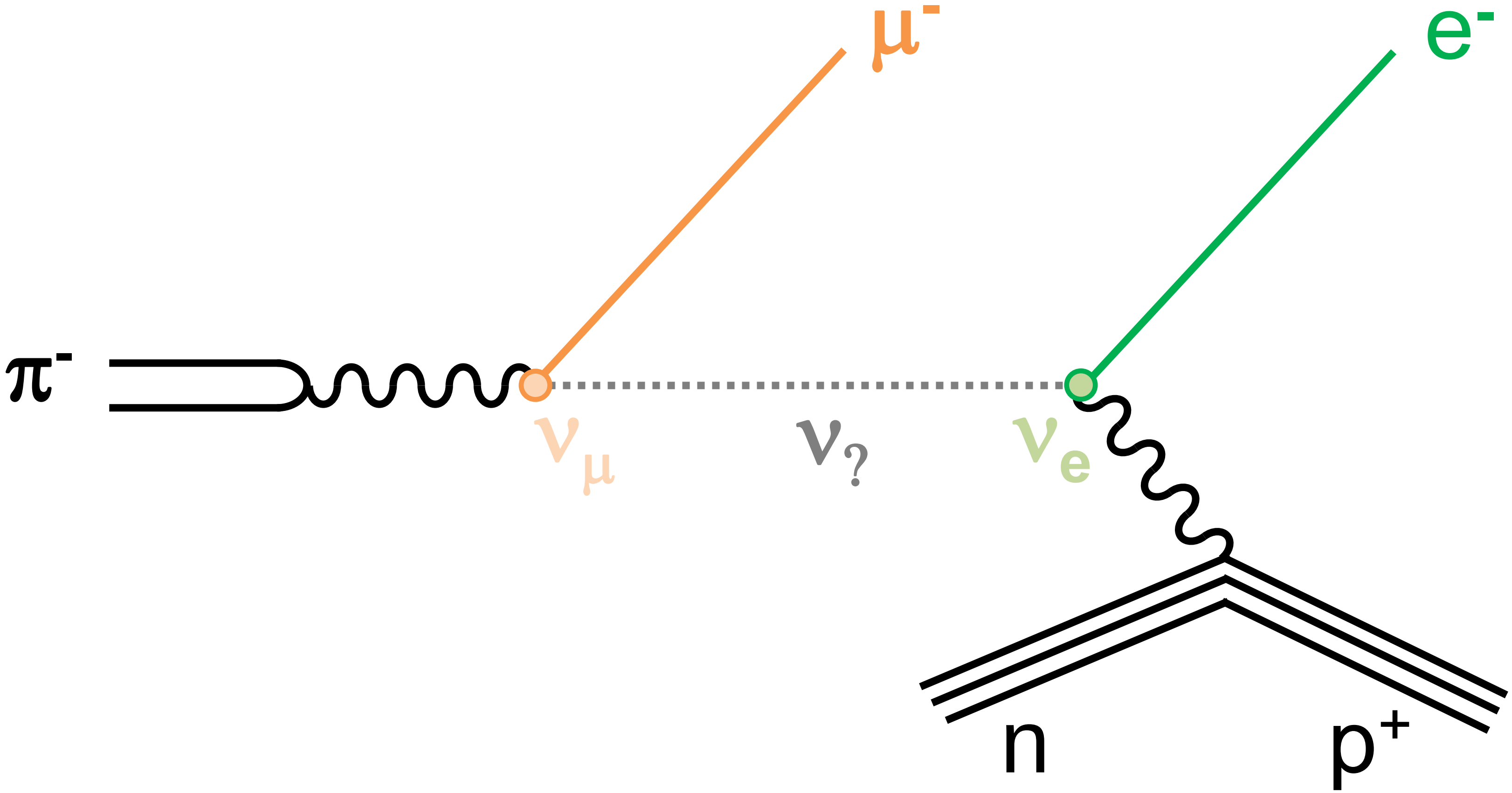
Neutrino Oscillations

- Neutrinos are created in a superposition of mass states
- Time evolution generates flavour oscillations

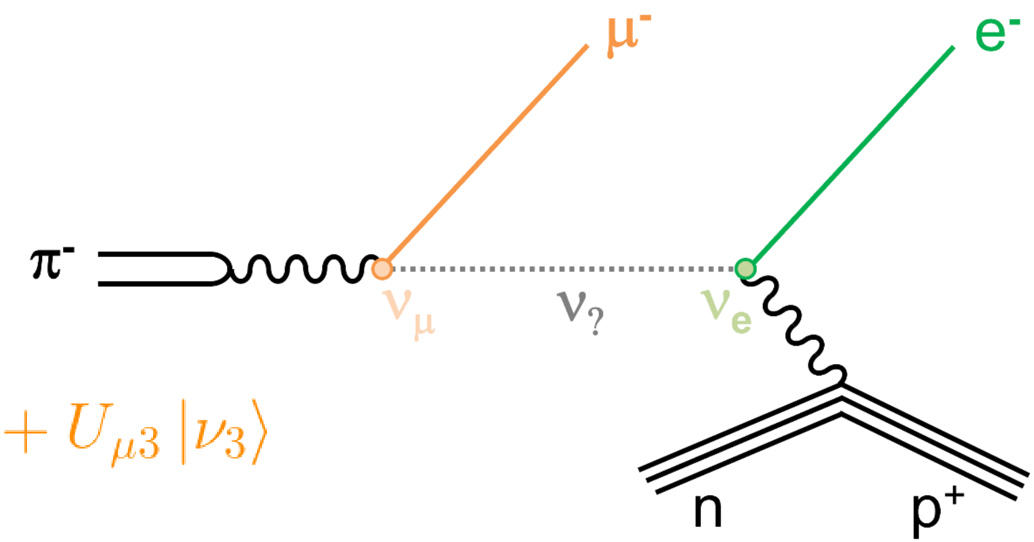
Quantum Mechanics



Neutrino Oscillations



Neutrino Oscillations



$$|\nu_\mu\rangle = U_{\mu 1}|\nu_1\rangle + U_{\mu 2}|\nu_2\rangle + U_{\mu 3}|\nu_3\rangle$$

$$|\nu(t)\rangle = U_{\mu 1}e^{-iE_1 t}|\nu_1\rangle + U_{\mu 2}e^{-iE_2 t}|\nu_2\rangle + U_{\mu 3}e^{-iE_3 t}|\nu_3\rangle$$

$$P_{\mu e} = |\langle\nu_e|\nu(t)\rangle|^2 = \left|U^*_{e1}U_{\mu 1}e^{-iE_1 t} + U^*_{e2}U_{\mu 2}e^{-iE_2 t} + U^*_{e3}U_{\mu 3}e^{-iE_3 t}\right|^2$$

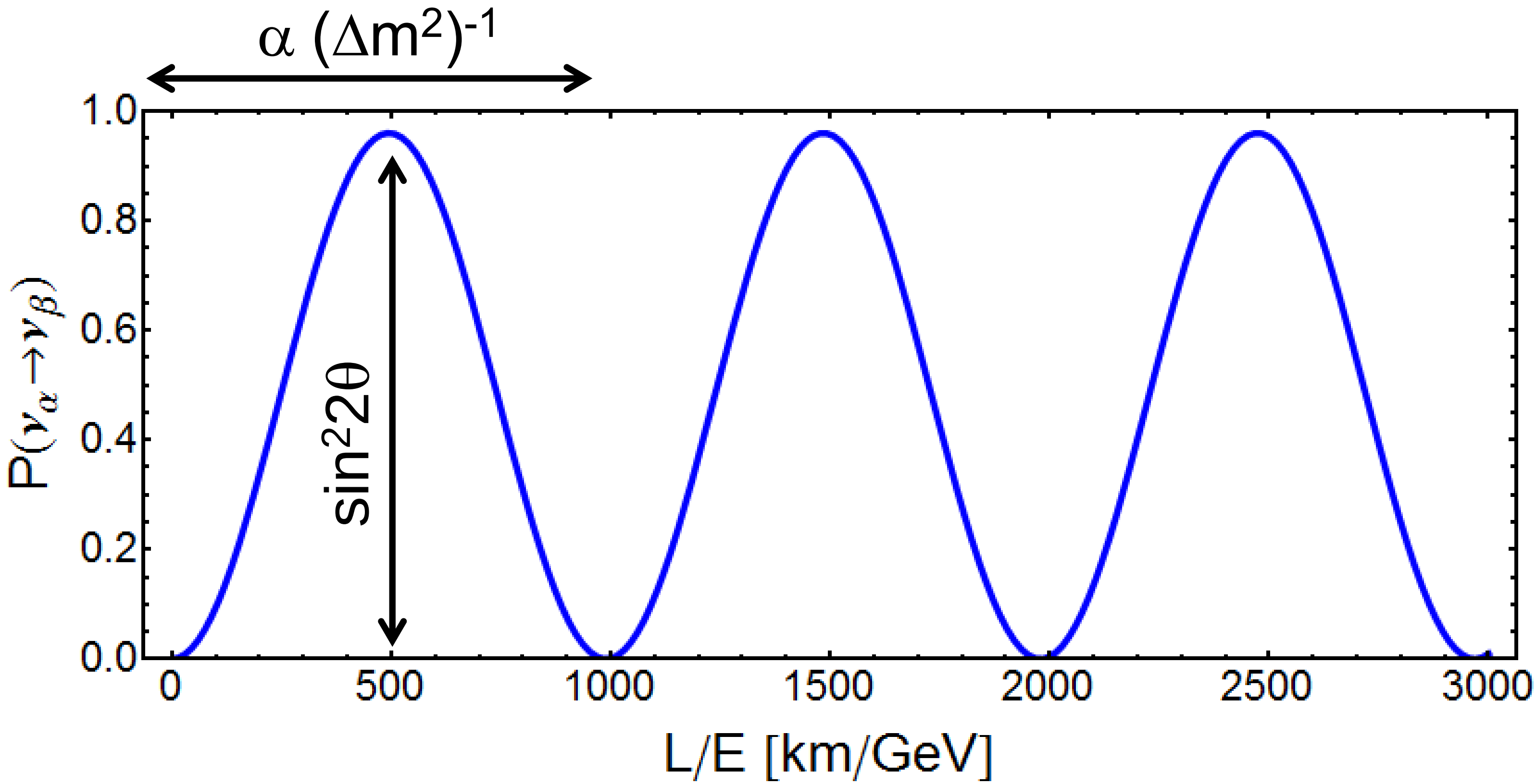
$$E_i \approx E + \frac{m_i^2}{2E}$$

$$t \approx L$$

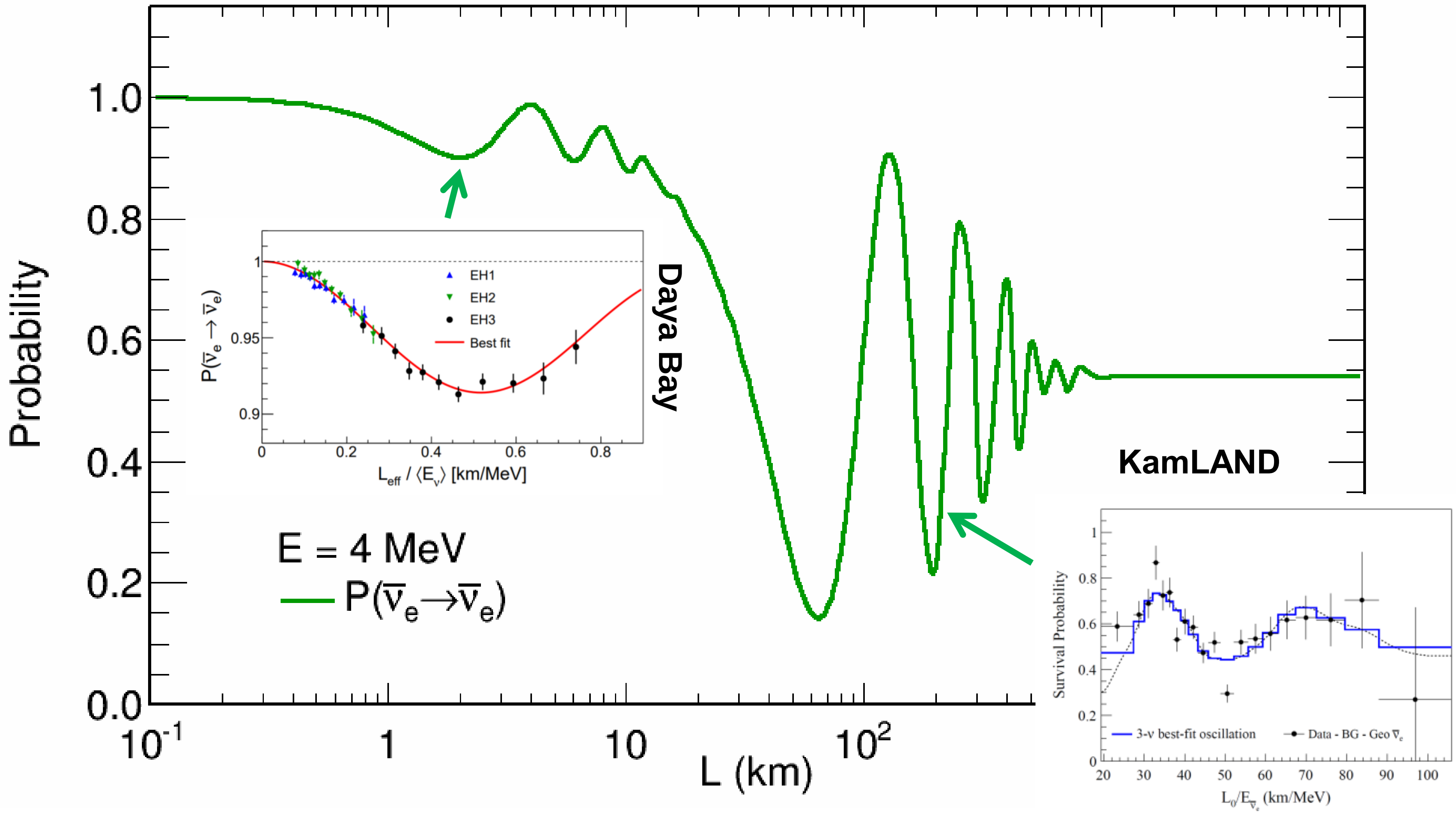
$$P_{\alpha\to\beta} = \delta_{\alpha\beta} - 4\sum_{j>k}\mathcal{R}_e\left\{U^*_{\alpha j}U_{\beta j}U_{\alpha k}U^*_{\beta k}\right\}\sin^2\left(\frac{\Delta_{jk}m^2 L}{4E}\right) + 2\sum_{j>k}\mathcal{I}_m\left\{U^*_{\alpha j}U_{\beta j}U_{\alpha k}U^*_{\beta k}\right\}\sin\left(\frac{\Delta_{jk}m^2 L}{2E}\right),$$

Neutrino Oscillations

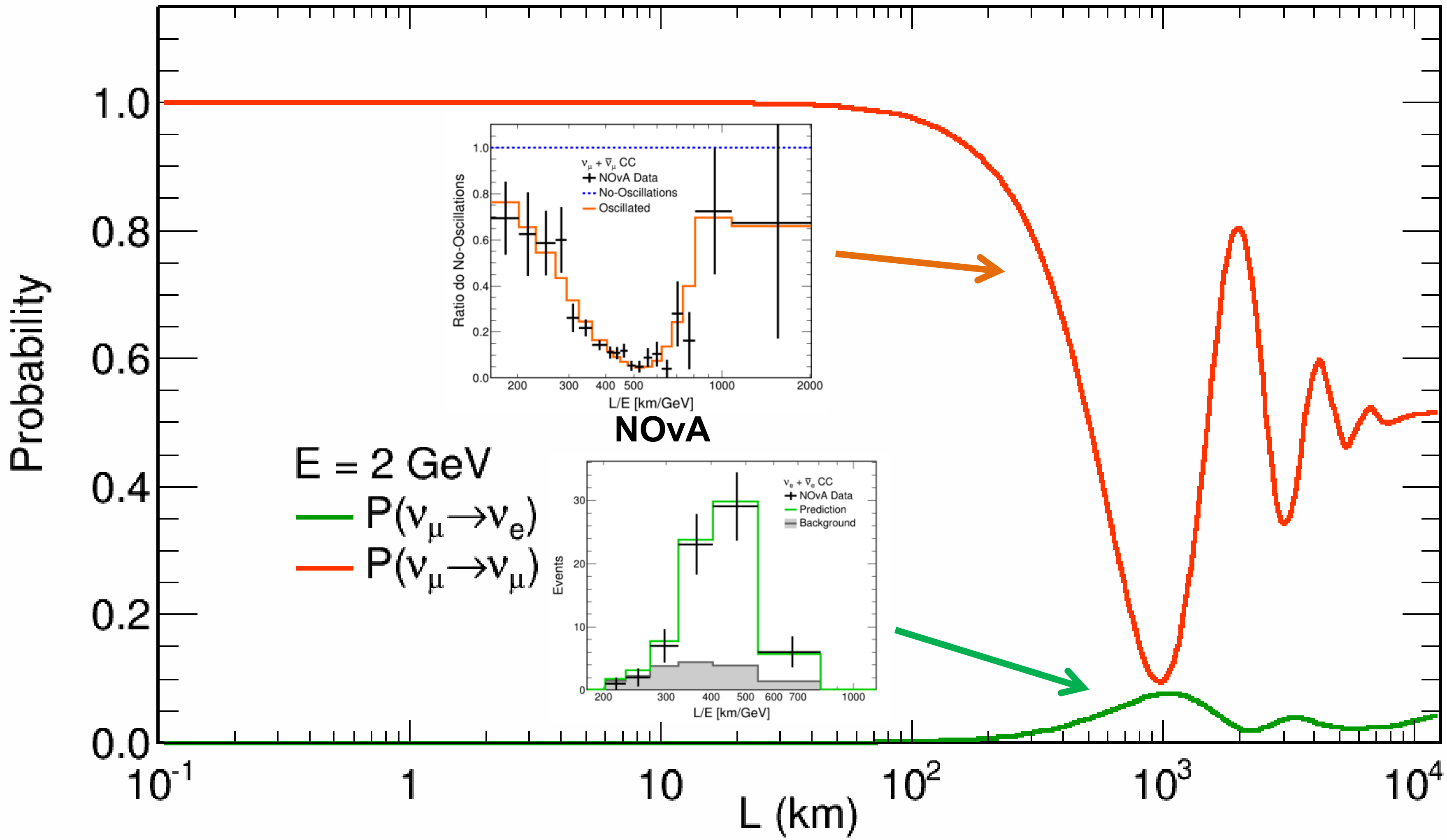
$$P(\nu_\alpha \to \nu_\beta) \approx \sin^2 2\theta \times \sin^2\left(1.27 \times \Delta m^2\ [\mathrm{eV}^2] \times L/E\ [\mathrm{km/GeV}]\right)$$



The Data: Reactor Neutrinos

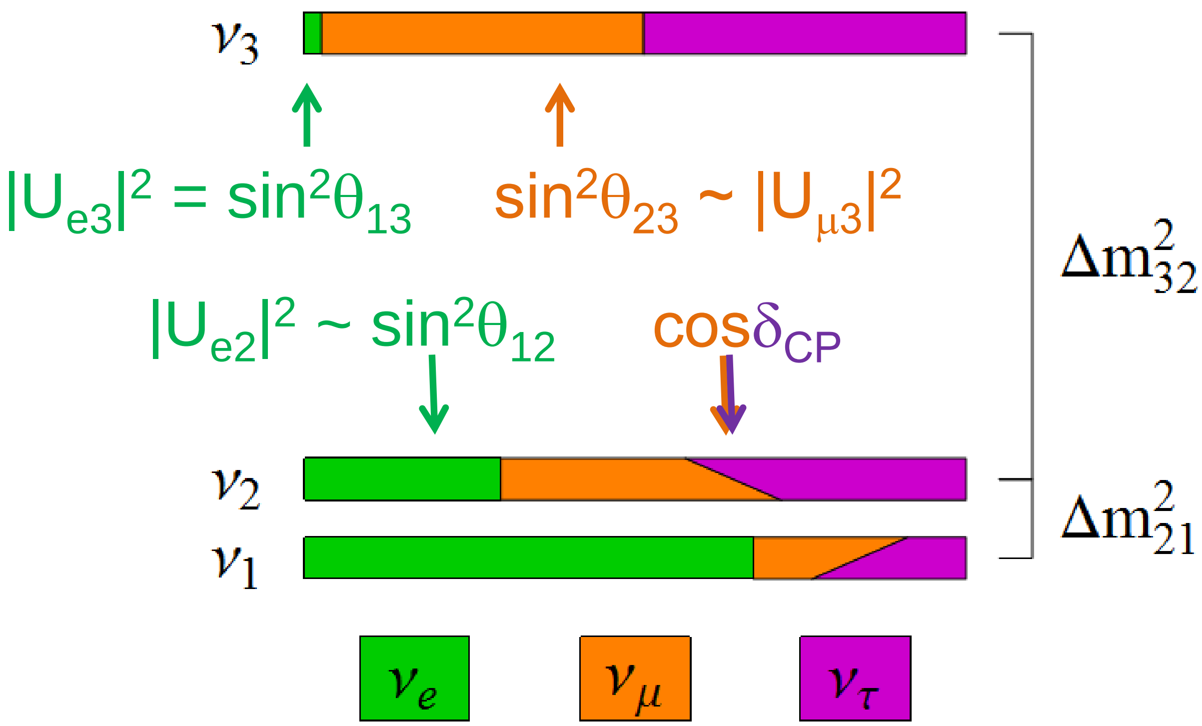


The Data: Accelerator Neutrinos



Neutrino Oscillations

- There are 3 neutrinos, so things are a bit more complicated
- Two independent differences in mass-squared (Δm^2_{21}, Δm^2_{32})
- 3 mixing angles (θ_{12}, θ_{13}, θ_{23}) and 1 CPV phase δ_{CP}



Quantum Evolution

Schrödinger: $i\frac{\partial}{\partial t}\mathcal{U} = H\,\mathcal{U}$

$\mathcal{P}_{\alpha\to\beta} = |\langle\beta|\,\mathcal{U}(t)\,|\alpha\rangle|^2$

Time-independent H:

$\mathcal{U}(t) = e^{-iHt}$

$H = V H_D V^\dagger$ ← **Main problem**

$\mathcal{U}(t) = V e^{-iH_D t} V^\dagger$ ← **Easy to compute**

$V H_D V^\dagger$

Eigenvectors (→ V) **Eigenvalues** (→ H_D)

Neutrino Hamiltonian in Vacuum

PMNS Matrix = Vacuum Eigenvectors

$$U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{bmatrix} \begin{bmatrix} c_{13} & 0 & s_{13}e^{-i\delta_{CP}} \\ 0 & 1 & 0 \\ -s_{13}e^{i\delta_{CP}} & 0 & c_{23} \end{bmatrix} \begin{bmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Mass basis

$$H_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\Delta m^2_{21}}{2E} & 0 \\ 0 & 0 & \frac{\Delta m^2_{31}}{2E} \end{bmatrix}$$

Flavour basis

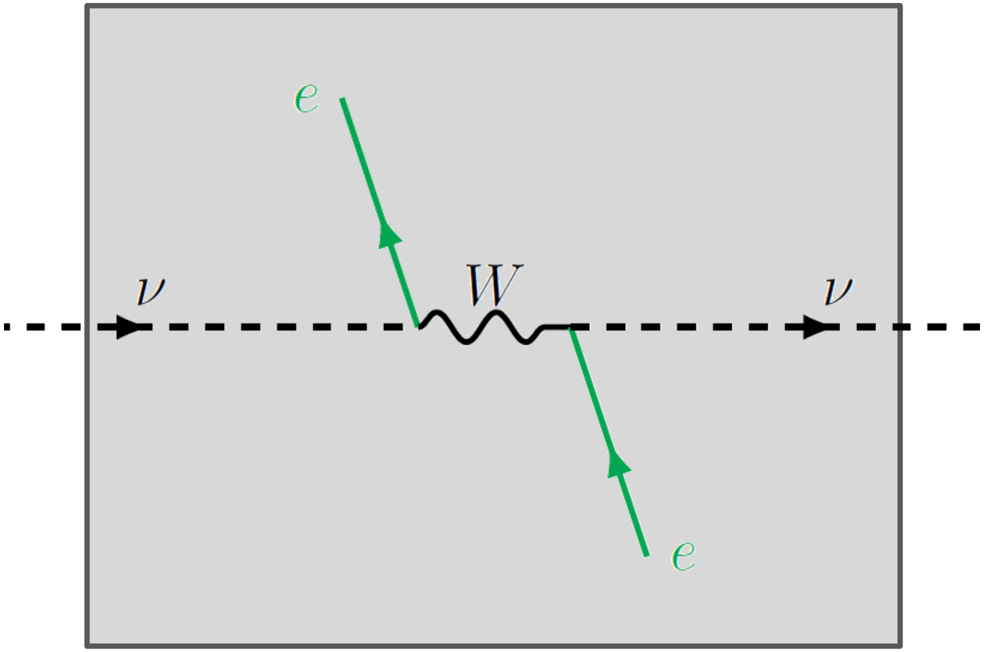
$$H = U H_0 U^\dagger$$

Easy to solve

Already know diagonalisation

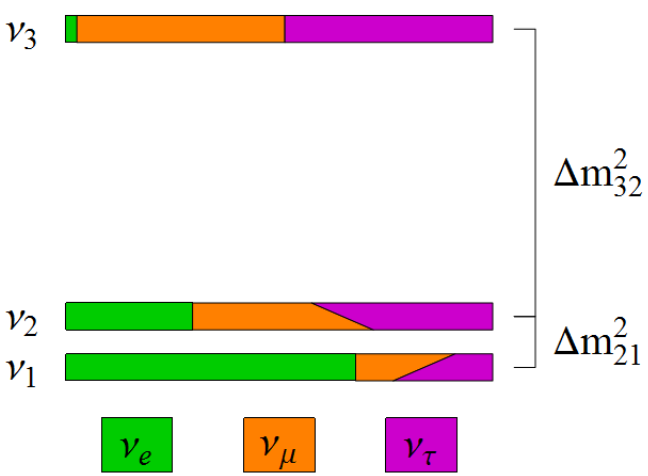
(Eigenvectors and Eigenvalues)

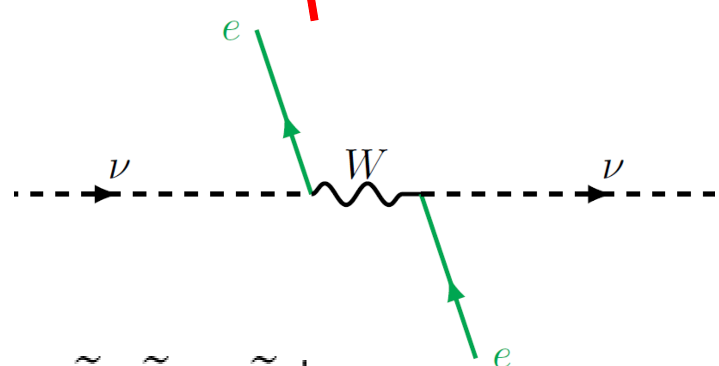
Neutrinos in Matter



Matter Effects

$$\overbrace{H_{eff} = U \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\Delta m^2_{21}}{2E} & 0 \\ 0 & 0 & \frac{\Delta m^2_{31}}{2E} \end{bmatrix} U^\dagger}^{H_0} + \overbrace{V_e \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}^{V}$$



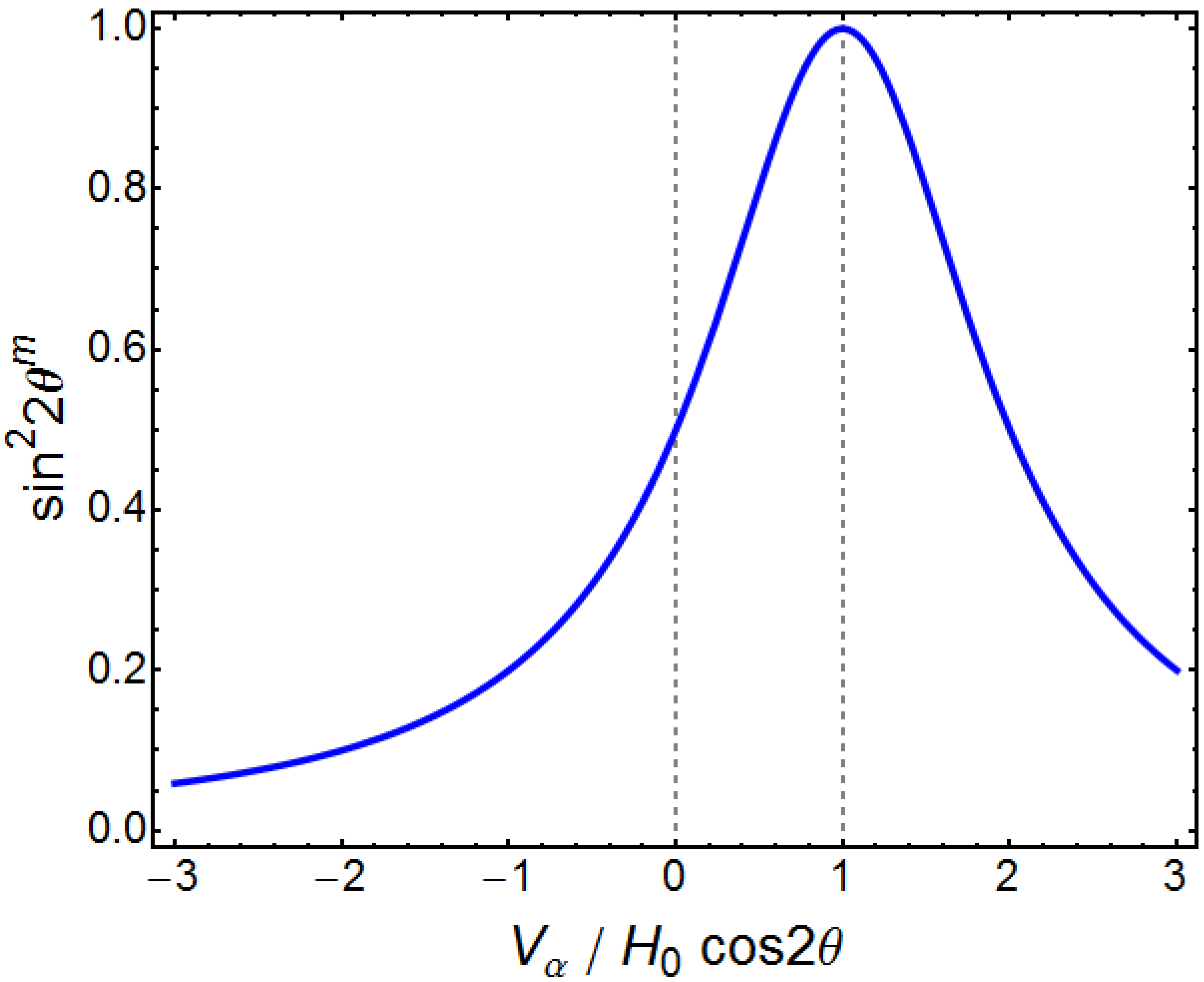


$$H_{eff} = \tilde{U}\,\tilde{H}_D \tilde{U}^\dagger$$

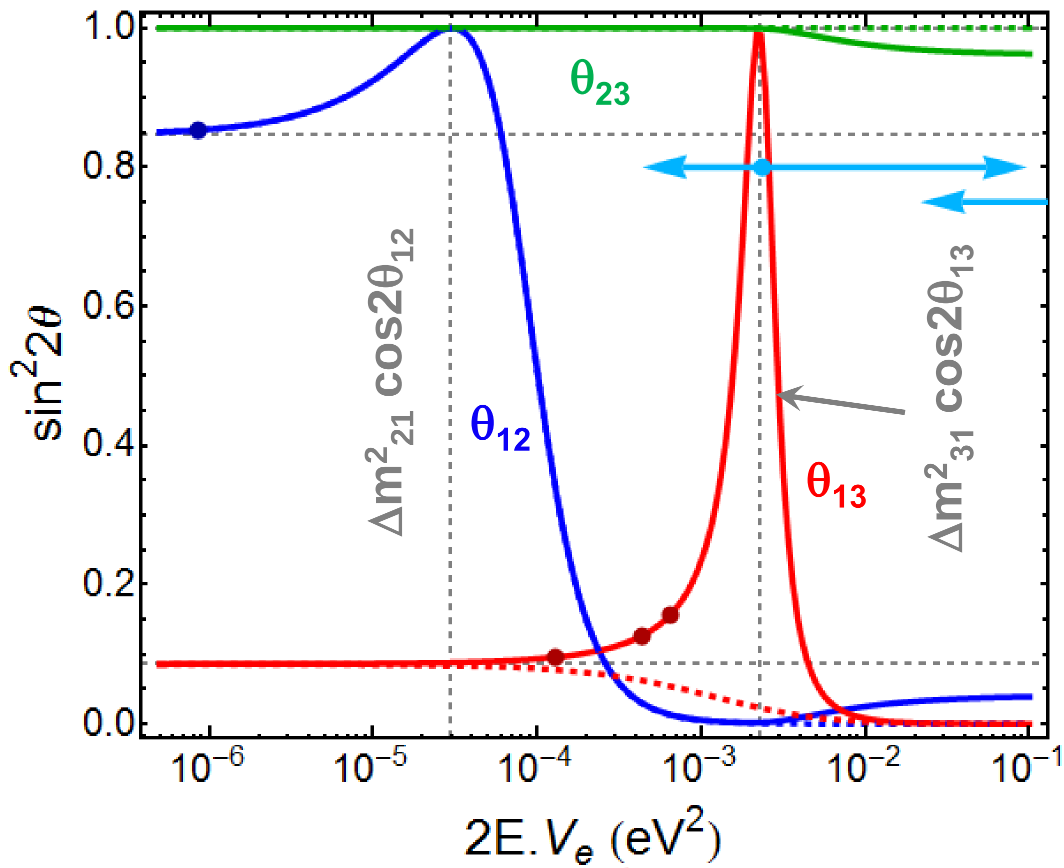
Effective Mixing

Effective Masses

Resonance

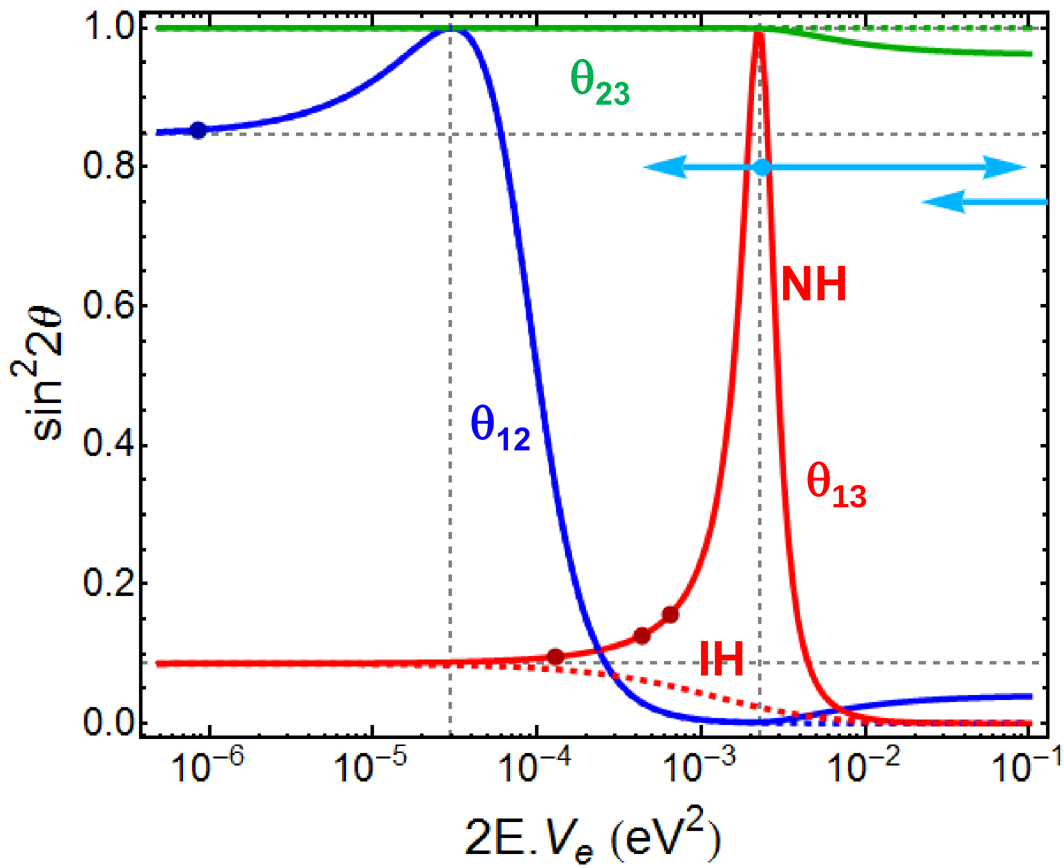


Resonances



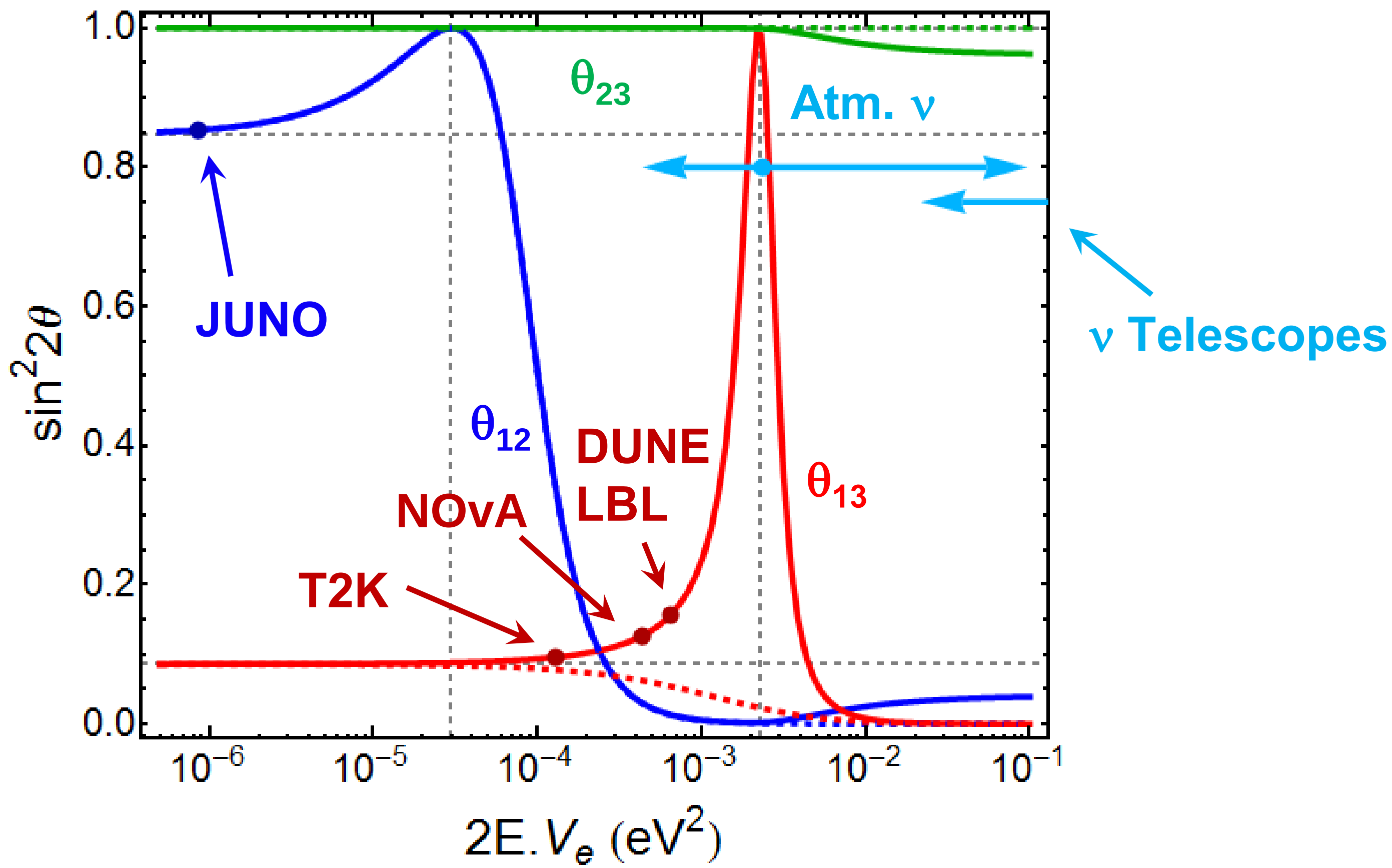
$$H_{eff} = U \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\Delta m^2_{21}}{2E} & 0 \\ 0 & 0 & \frac{\Delta m^2_{31}}{2E} \end{bmatrix} U^\dagger + V_e \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Resonances



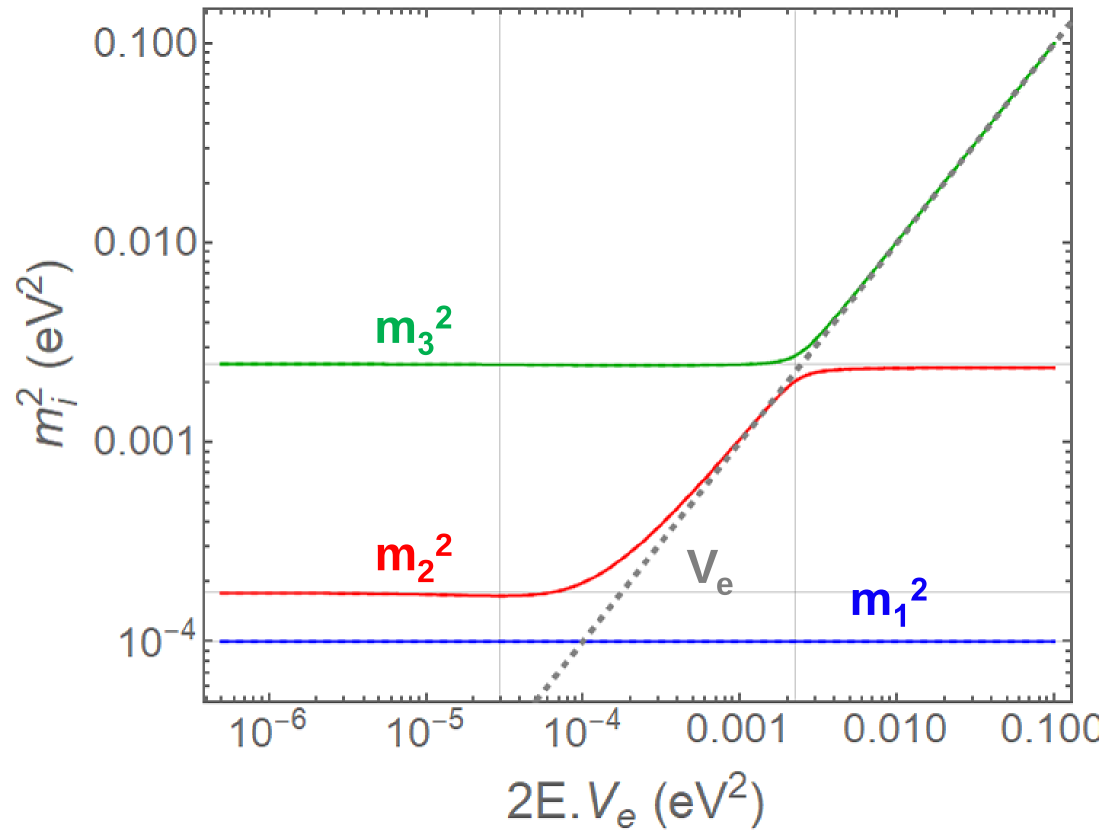
$$H_{eff} = U \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\Delta m^2_{21}}{2E} & 0 \\ 0 & 0 & \frac{\Delta m^2_{31}}{2E} \end{bmatrix} U^\dagger + V_e \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Resonances



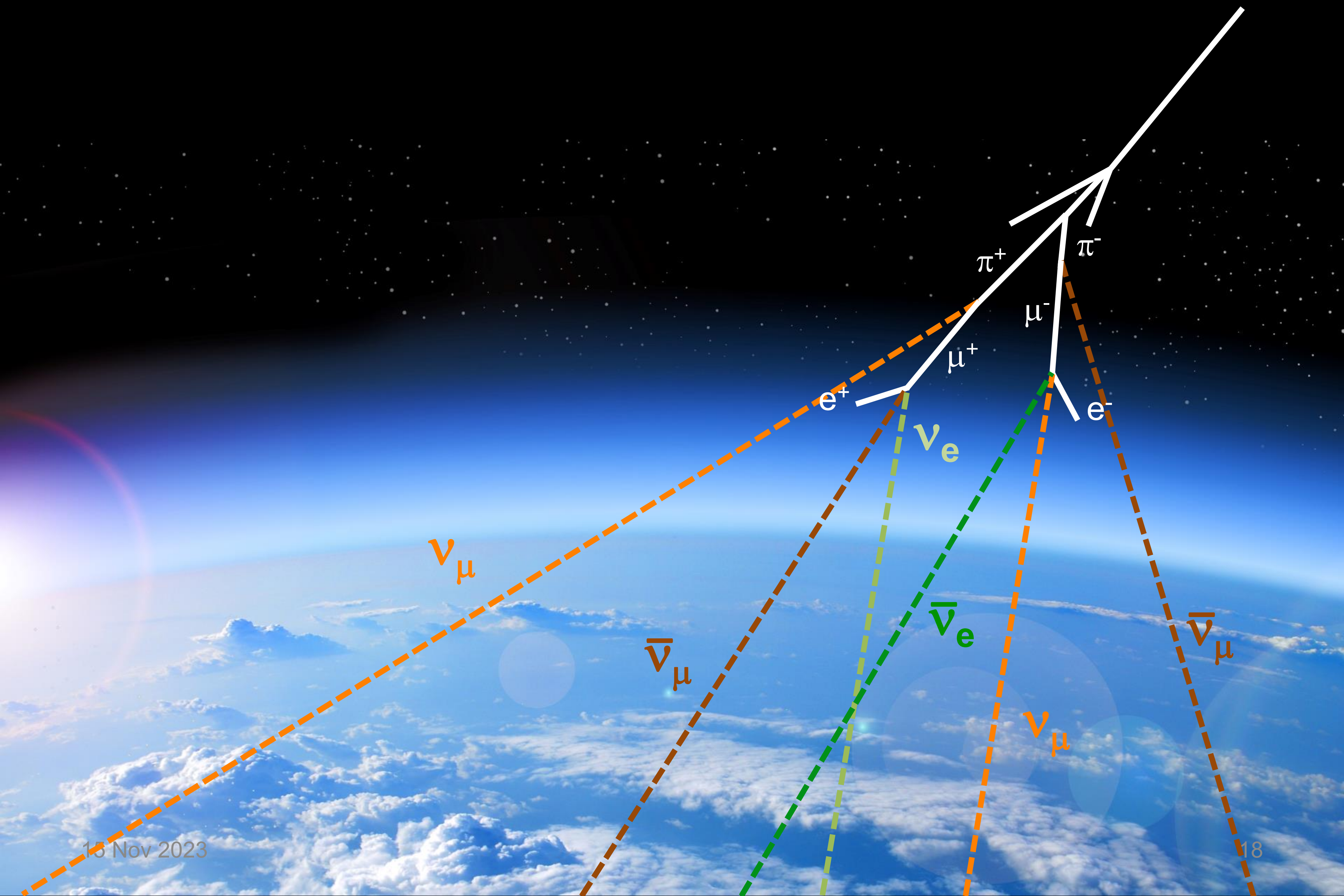
$$H_{eff} = U \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\Delta m^2_{21}}{2E} & 0 \\ 0 & 0 & \frac{\Delta m^2_{31}}{2E} \end{bmatrix} U^\dagger + V_e \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Resonances

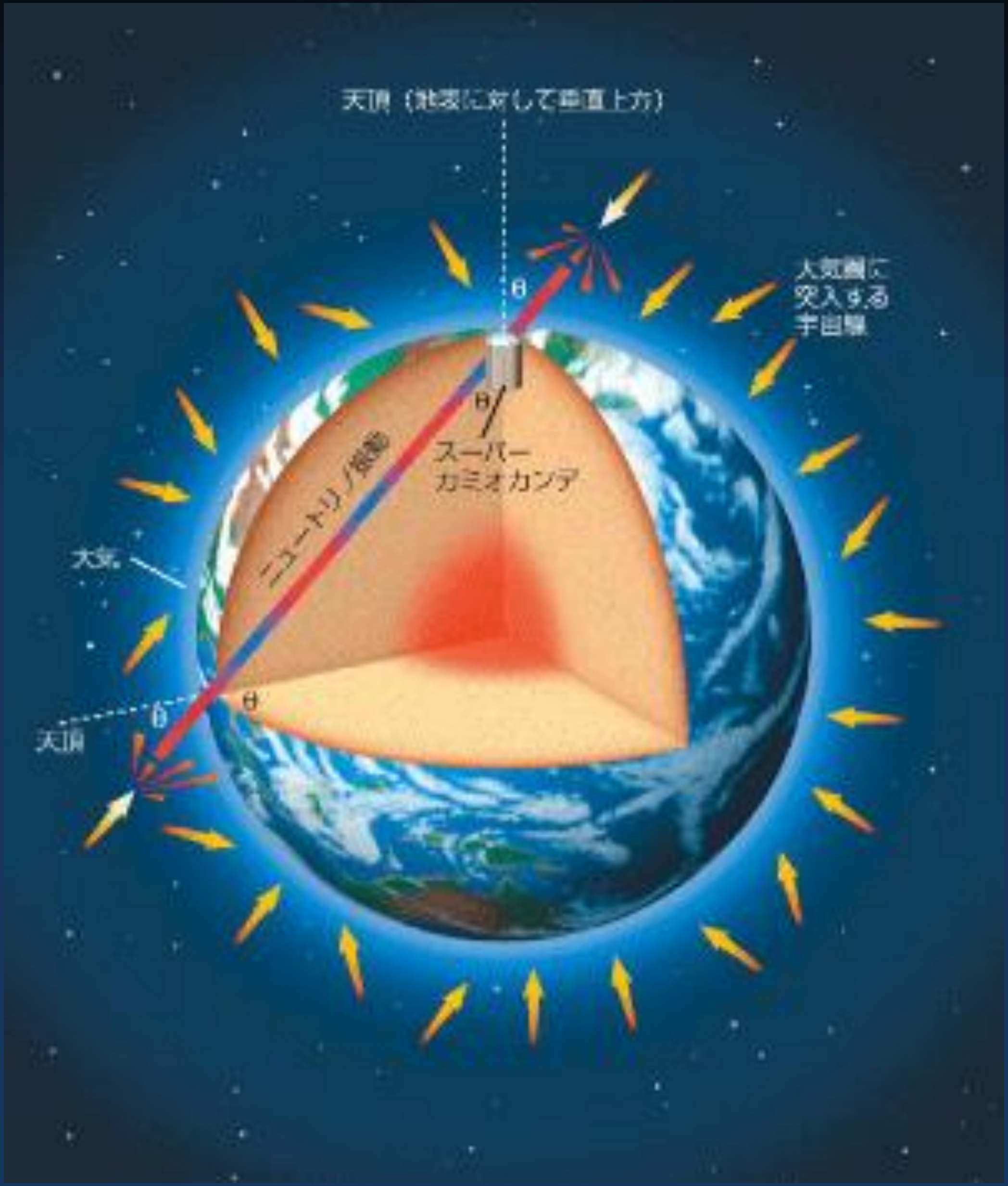


$$H_{eff} = U \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\Delta m_{21}^2}{2E} & 0 \\ 0 & 0 & \frac{\Delta m_{31}^2}{2E} \end{bmatrix} U^\dagger + V_e \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Atmospheric Neutrinos



Atmospheric Neutrinos



Neutrino Hamiltonian in Matter

$$H_{eff} = U \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\Delta m^2_{21}}{2E} & 0 \\ 0 & 0 & \frac{\Delta m^2_{31}}{2E} \end{bmatrix} U^\dagger + V_e \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

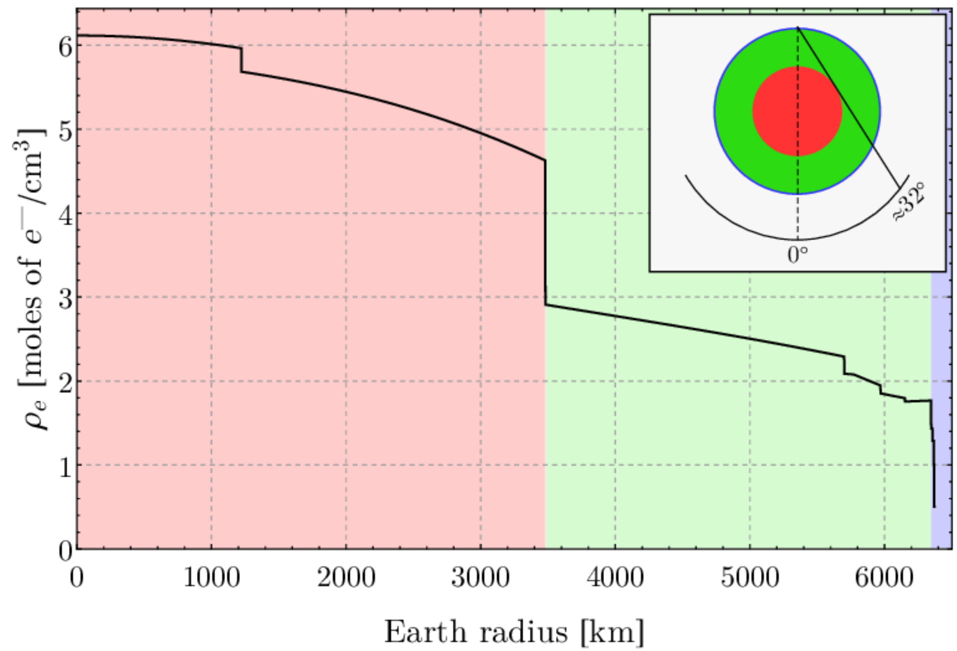
Time Dependent

$V_e \equiv \pm\sqrt{2} G_F n_e$ (+ for ν, − for $\overline{\nu}$)

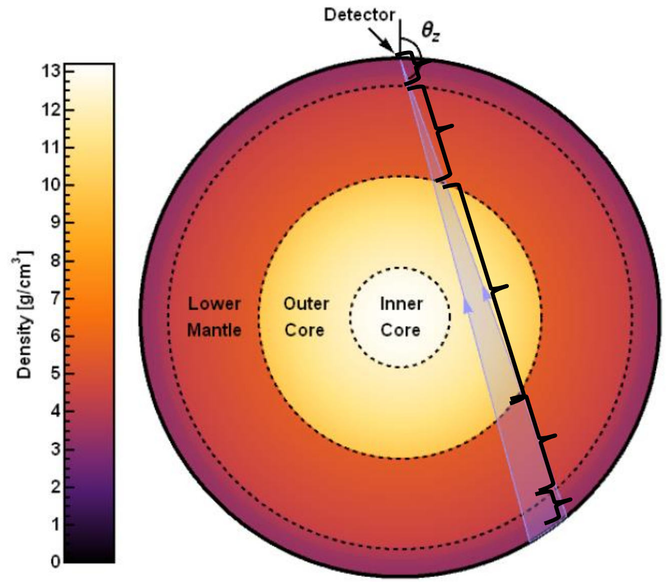
Relevant when $V_e \gtrsim \frac{\Delta m^2}{2E} \sim 1/L$

In general, $1/V_e \sim 1700$ km $\times \left(\frac{3 \text{ g/cm}^3}{\rho}\right)$

Comparable to $\frac{\Delta m^2_{31}}{2E}$ for $E \sim 10$ GeV



Quantum Evolution



Schrödinger: $i\frac{\partial}{\partial t}\mathcal{U} = H\,\mathcal{U}$

$$\mathcal{P}_{\alpha\to\beta} = |\langle\beta|\,\mathcal{U}(t)\,|\alpha\rangle|^2$$

Time-independent H:

$$\mathcal{U}(t) = e^{-iHt}$$

$$H = V H_D V^\dagger$$

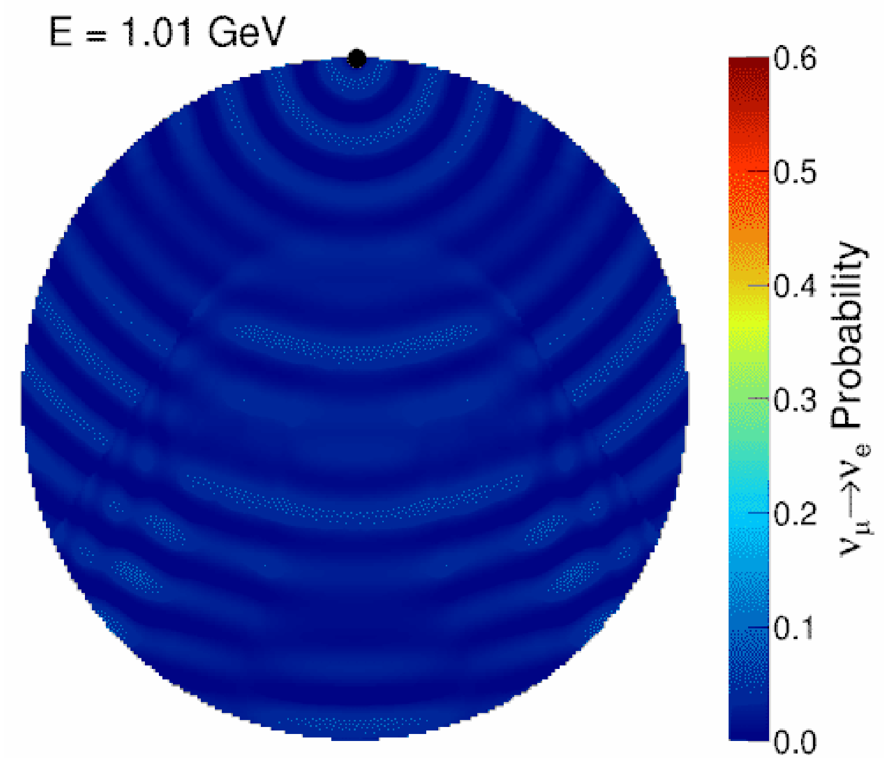
$$\mathcal{U}(t) = V e^{-iH_D t} V^\dagger$$

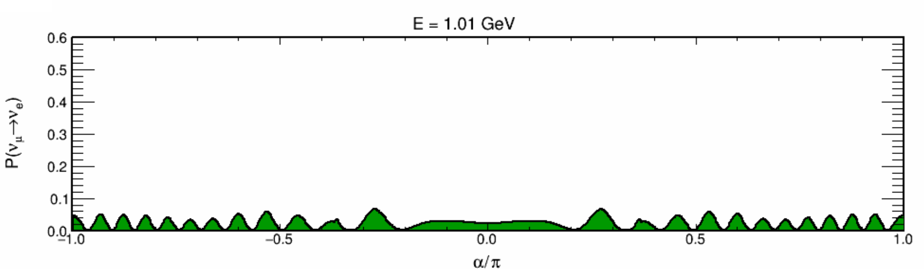
Time-dependent H:

$$\mathcal{U}(t) = \mathcal{T} e^{-i\int_0^t H(t')dt'} \approx \prod_k e^{-iH(t_k)\Delta t}$$

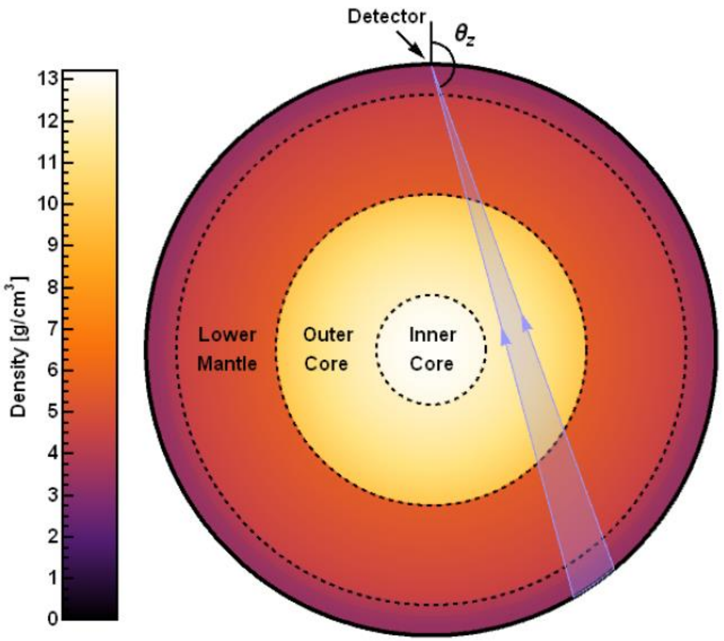
- Trace neutrino path through the Earth
- Break path into N segments of similar electron density
- Compute evolution through each segment with constant density assumption

Atmospheric Neutrinos





Nue appearance at the surface



Oscillations are **resonant** at certain energies

E_{res} ~ 7 GeV in Mantle
E_{res} ~ 3 GeV in Core

Extended Models

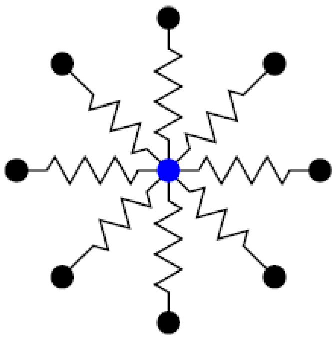
Non-Standard Interactions (NSI)

$$H_{eff} = U \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\Delta m^2_{21}}{2E} & 0 \\ 0 & 0 & \frac{\Delta m^2_{31}}{2E} \end{bmatrix} U^\dagger + V_e \begin{bmatrix} 1+\epsilon_{ee} & \epsilon_{e\mu} & \epsilon_{e\tau} \\ \epsilon^*_{e\mu} & \epsilon_{\mu\mu} & \epsilon_{\mu\tau} \\ \epsilon^*_{e\tau} & \epsilon^*_{\mu\tau} & \epsilon_{\tau\tau} \end{bmatrix}$$

Sterile Neutrinos (3+N Flavours)

$$H_{eff} = U_S \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots \\ 0 & \frac{\Delta m^2_{21}}{2E} & 0 & 0 & \cdots \\ 0 & 0 & \frac{\Delta m^2_{31}}{2E} & 0 & \cdots \\ 0 & 0 & 0 & \frac{\Delta m^2_{41}}{2E} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} U_S^\dagger + \begin{bmatrix} V_e & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & V_n/2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

$$U_S = U_{N-1,N} \cdots U_{34} U_{24}^{(c)} U_{14}^{(c)} U_{23} U_{13}^{(c)} U_{12}$$



Extended Models

Decoherence

HELP NEEDED

Unitary **Non-Unitary**

$$\partial_t \rho = -i[H,\rho] + \frac{1}{2}\sum_j 2A_j\rho A_j^\dagger - \{A_j^\dagger A_j, \rho\}$$

Operators as sum SU(3) of generators

$$\mathscr{O} = \sum_j \mathrm{tr}\left[\mathscr{O}F_j\right] F_j$$

In general* 36 parameters!

$$\widetilde{L}_{jk} = \frac{1}{2}\sum_{lmn} (\vec{a}_l \cdot \vec{a}_m) f_{lkn} f_{nmj}$$

$$\partial_t \vec{\rho} = \left(\widetilde{H} - \widetilde{L}\right)\vec{\rho}$$

System of 9 coupled equations

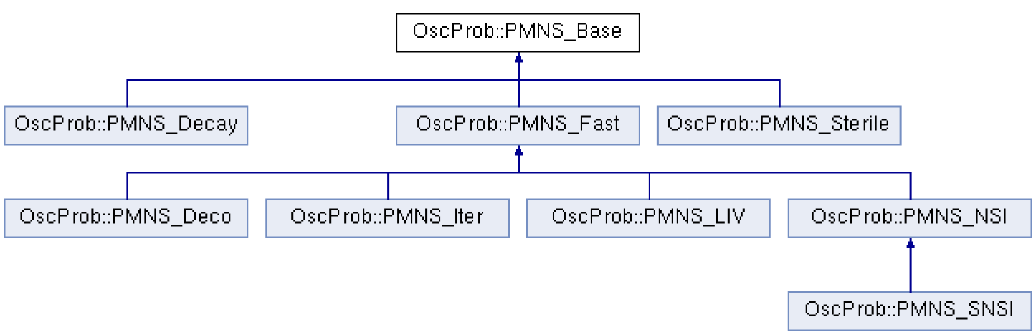
Diagonal w/ energy conserv.

$$\widetilde{L} = \begin{bmatrix} 0_{3\times 3} & 0 & 0 & 0 \\ 0 & I_2\Gamma_{21} & 0 & 0 \\ 0 & 0 & I_2\Gamma_{31} & 0 \\ 0 & 0 & 0 & I_2\Gamma_{32} \end{bmatrix}$$

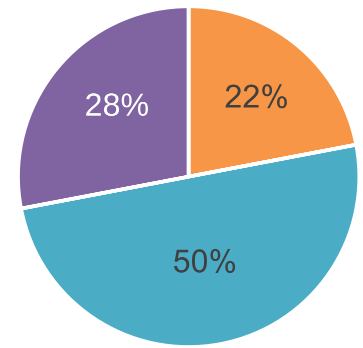
OscProb Package

- Diagonalises Hamiltonian to obtain **exact probabilities**
- Three step process:
  - **Build Hamiltonian** from parameters
  - **Solve Hamiltonian**
    - Fast algorithm from GLoBES for 3 neutrinos*
  - **Propagate neutrino** state
- Repeat for each step of constant matter in neutrino path
- PremModel class has built-in Earth layers model

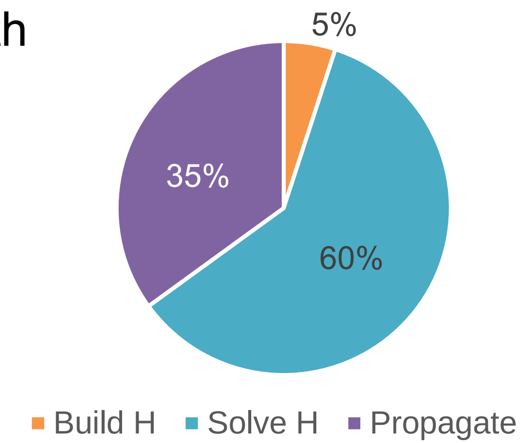
https://github.com/joaoabcoelho/OscProb



Single Step (2.2μs)



85 steps (110μs) †

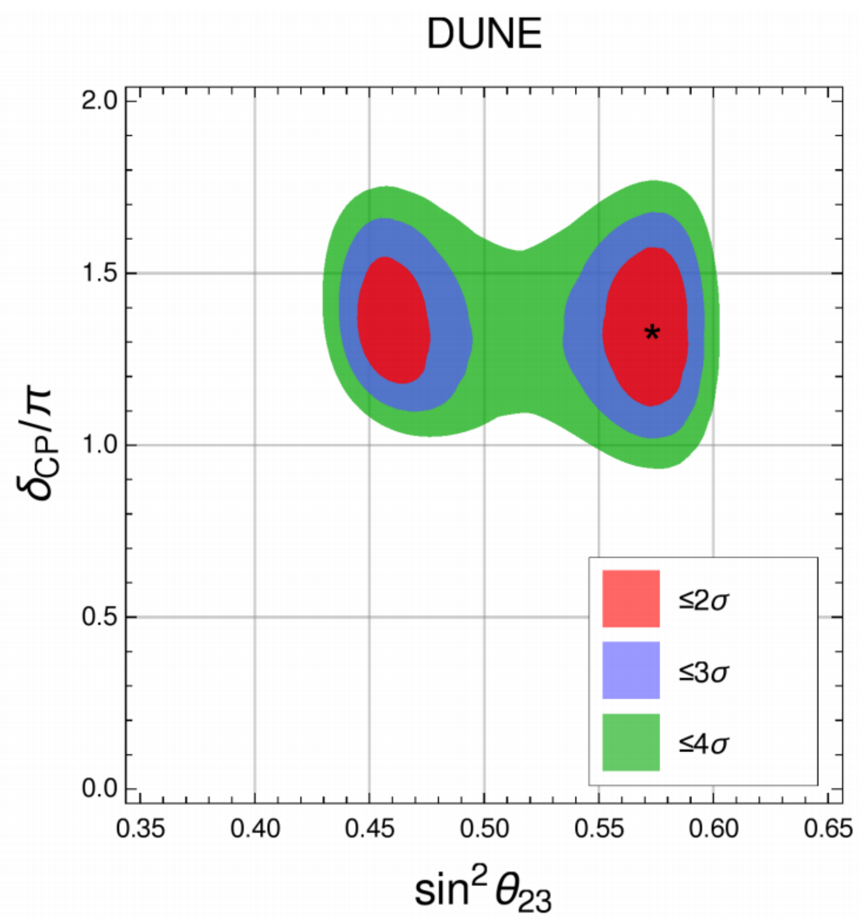


† Up-going (42+2 layers)

*J. Kopp, Int. J. Mod. Phys. C, **19**, 523 (2008)

Why Performance Matters

- Computing oscillation probabilities is a big part of CPU time
- Total computations: $\mathbf{F_p \times F_d \times CP \times E_b \times \Theta_b \times L \times C \times Pn \times M}$
  - Fp: Initial flavours produced (2)
  - Fd: Flavours detected (3+NC)
  - CP: Nu and nubar (2)
  - E_b: Energy bins (~100)
  - Θ_b: Direction bins (~100)
  - L: Earth layers to cross (~40)
  - C: $\Delta\chi^2$ surface points (e.g. contour: ~50x50)
  - P_n: Nuisance parameters (e.g. syst.: ~20)
  - M: Minimisation steps (~100)



- Typically **~10^{13}** computations to obtain a full likelihood surface
- At 1µs / comp.: **~1 CPU-year**
- M typically grows with P_n
- **Feldman-Cousins corrections would require ~10k contours**

Main Optimisations

- Use a fast algorithm for solving **eigensystem**
  - J. Kopp, Int. J. Mod. Phys. C, **19**, 523 (2008)
  - Uses analytical solutions for 3x3 matrix
  - Developed for GLoBES

- Some BSM models need different methods: Eigen library
  - PMNS_Sterile class solves NxN matrices
  - PMNS_Decay class solves non-Hermitian matrices

- Reduce number of operations when **computing Hamiltonian**
  - Take into account known form of PMNS matrix

- Caching of eigensystem for reusing known solutions

Hamiltonian Optimisation

$$U_S = U_{N-1,N} \cdots U_{34} U_{24}^{(c)} U_{14}^{(c)} U_{23} U_{13}^{(c)} U_{12}$$

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 40 \end{bmatrix} \xrightarrow{U_{12}} \begin{bmatrix} 3.2 & 4.7 & 0 & 0 \\ 4.7 & 6.8 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 40 \end{bmatrix}$$

$$\xrightarrow{U_{13}} \begin{bmatrix} 3.9 & 4.6 & 3.3 & 0 \\ 4.6 & 6.8 & -0.9 & 0 \\ 3.3 & -0.9 & 19 & 0 \\ 0 & 0 & 0 & 40 \end{bmatrix} \xrightarrow{U_{23}} \begin{bmatrix} 3.9 & 5.6 & -0.7 & 0 \\ 5.6 & 12 & 6.2 & 0 \\ -0.7 & 6.2 & 14 & 0 \\ 0 & 0 & 0 & 40 \end{bmatrix}$$

$$\xrightarrow{U_{14}} \begin{bmatrix} 5.3 & 5.5 & -0.7 & 7 \\ 5.5 & 12 & 6.2 & -1.1 \\ -0.7 & 6.2 & 14 & 0.1 \\ 7 & -1.1 & 0.1 & 39 \end{bmatrix} \xrightarrow{U_{24}} \begin{bmatrix} 5.3 & 6.8 & -0.7 & 5.8 \\ 6.8 & 12 & 6.1 & 4.2 \\ -0.7 & 6.1 & 14 & -1.1 \\ 5.8 & 4.2 & -1.1 & 38 \end{bmatrix}$$

$$\xrightarrow{U_{34}} \begin{bmatrix} 5.3 & 6.8 & 2.2 & 5.4 \\ 6.8 & 12 & 7.4 & 0.8 \\ 2.2 & 7.4 & 19 & 9.3 \\ 5.4 & 0.8 & 9.3 & 34 \end{bmatrix}$$

# of Operations

	Opt.	Std.
2x2	3	8
3x3	13	54
4x4	34	192
5x5	70	500
6x6	125	1080

- Each rotation only affects some columns and rows
- Hermitian, so only upper triangle
- Total of **$O(n^3)$ operations**
- Std. matrix mult. $\rightarrow O(n^4)$

Extra Considerations

1. Caching of eigensystem solutions
    - Often we have to solve the same eigensystem multiple times
    - E.g.: Neutrinos cross same Earth layer from different angles
    - OscProb can save these to avoid repeated computations
    - Balance between hashing overhead and eigensystem

2. Earth models can be too detailed for our needs
    - Default is 44 layers of constant density
    - Can easily go down to 15 layers
    - Even better, OscProb can dynamically merge similar layers

3. Parallelise oscillation propagations
    - Usually we need to compute 2x3 transitions ($\nu_{e,\mu}$ -> $\nu_{e,\mu,\tau}$)
    - OscProb can compute all of these in parallel
    - Avoids some extra propagation costs

Iterative Approximation

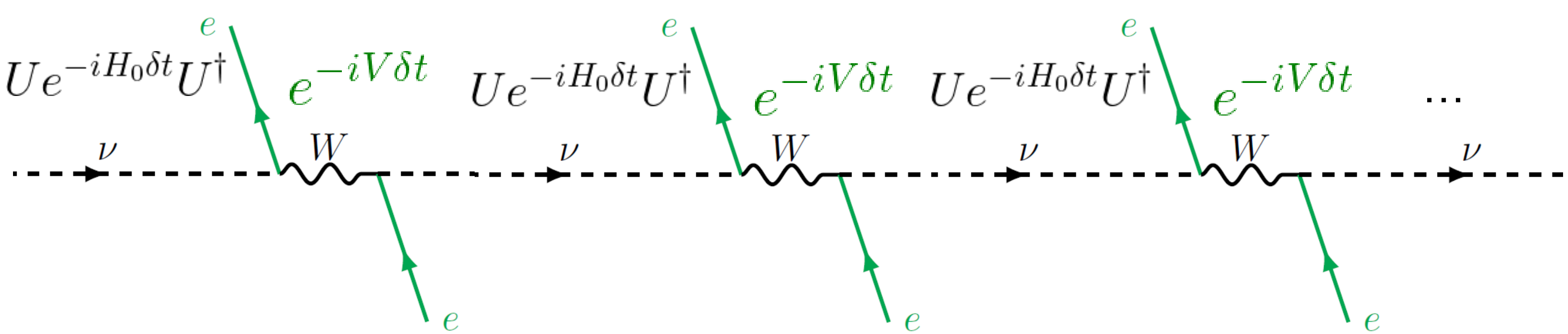
Zassenhaus Formula

$$e^{(X+Y)t} = e^{Xt} e^{Yt} e^{-[X,Y]\frac{t^2}{2}} e^{(2[Y,[X,Y]]+[X,[X,Y]])\frac{t^3}{6}} \dots$$

If t is small, we can ignore higher order terms

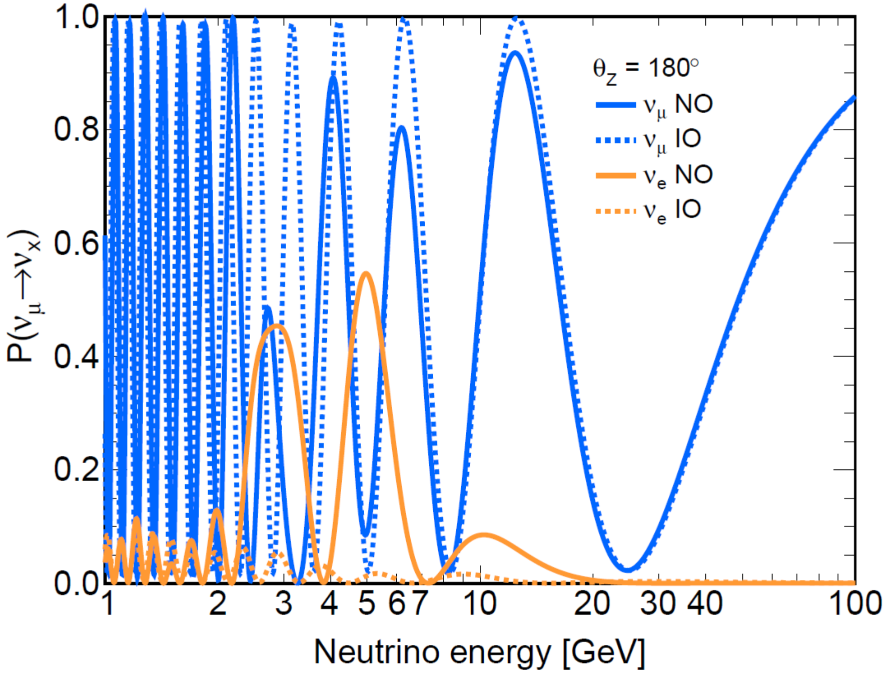
$$e^{-i(UH_0U^\dagger+V)t} \approx e^{-iUH_0U^\dagger t} e^{-iVt} = Ue^{-iH_0t}U^\dagger e^{-iVt}$$

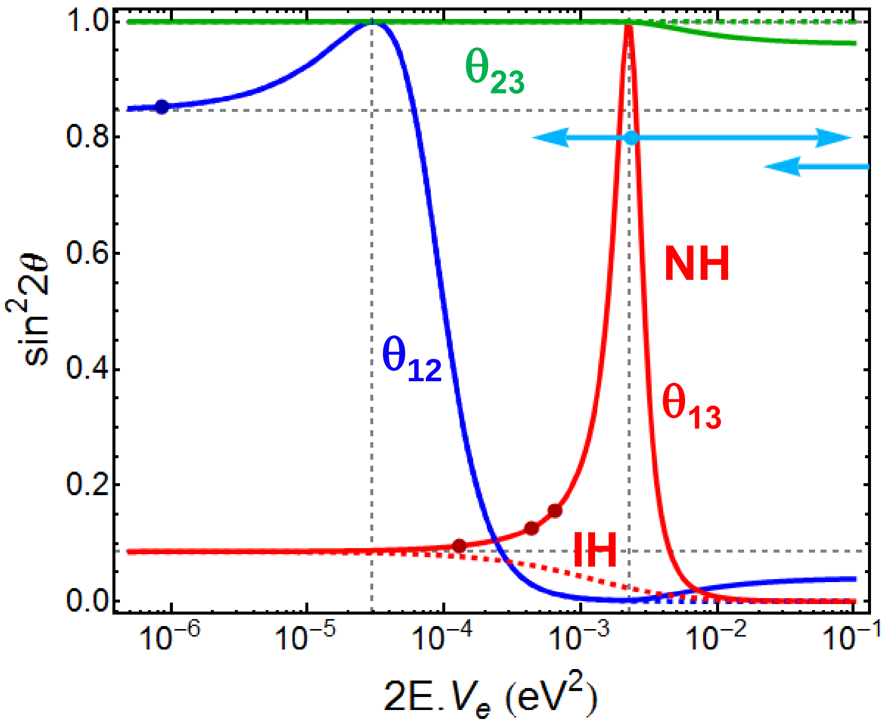
Everything is already diagonal. No need for solving eigensystems



Interpolation?

HELP NEEDED

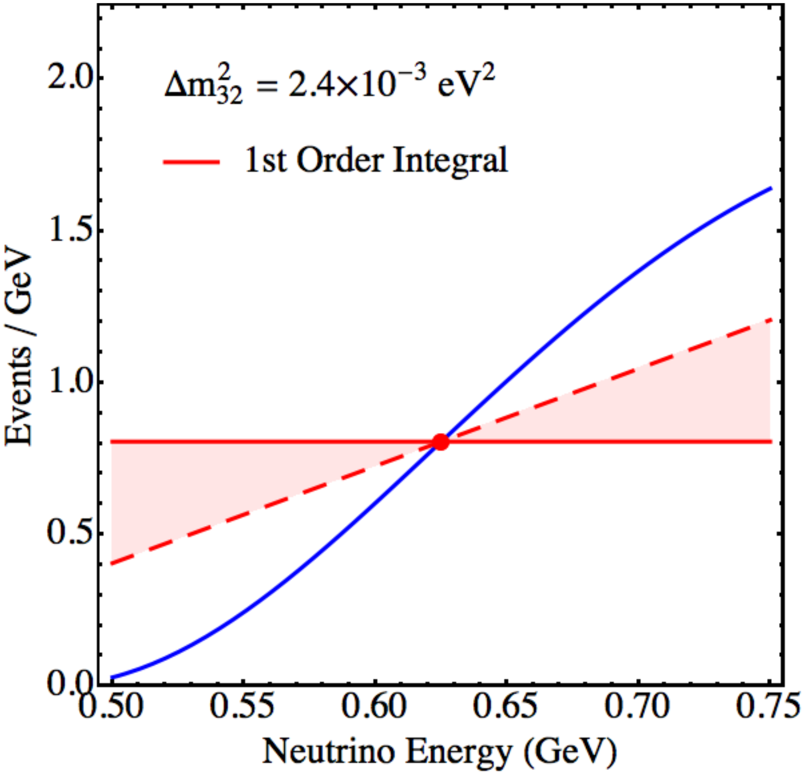




- Interpolating oscillation probabilities can be difficult due to fast oscillations at some energies
- One idea would be to instead interpolate the eigensystem solutions
- Equivalent to interpolating the effective mixing parameters

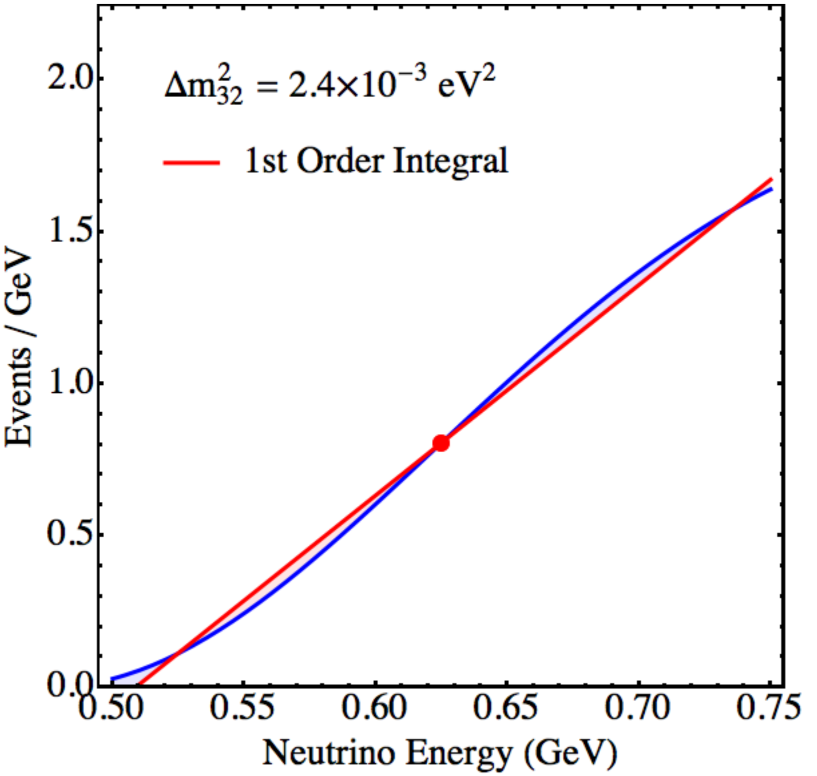
Oscillation Averaging

- Another problem with fast oscillations is computing the average oscillation over a bin
- Simply taking the bin center value is equivalent to a linear approximation of the function



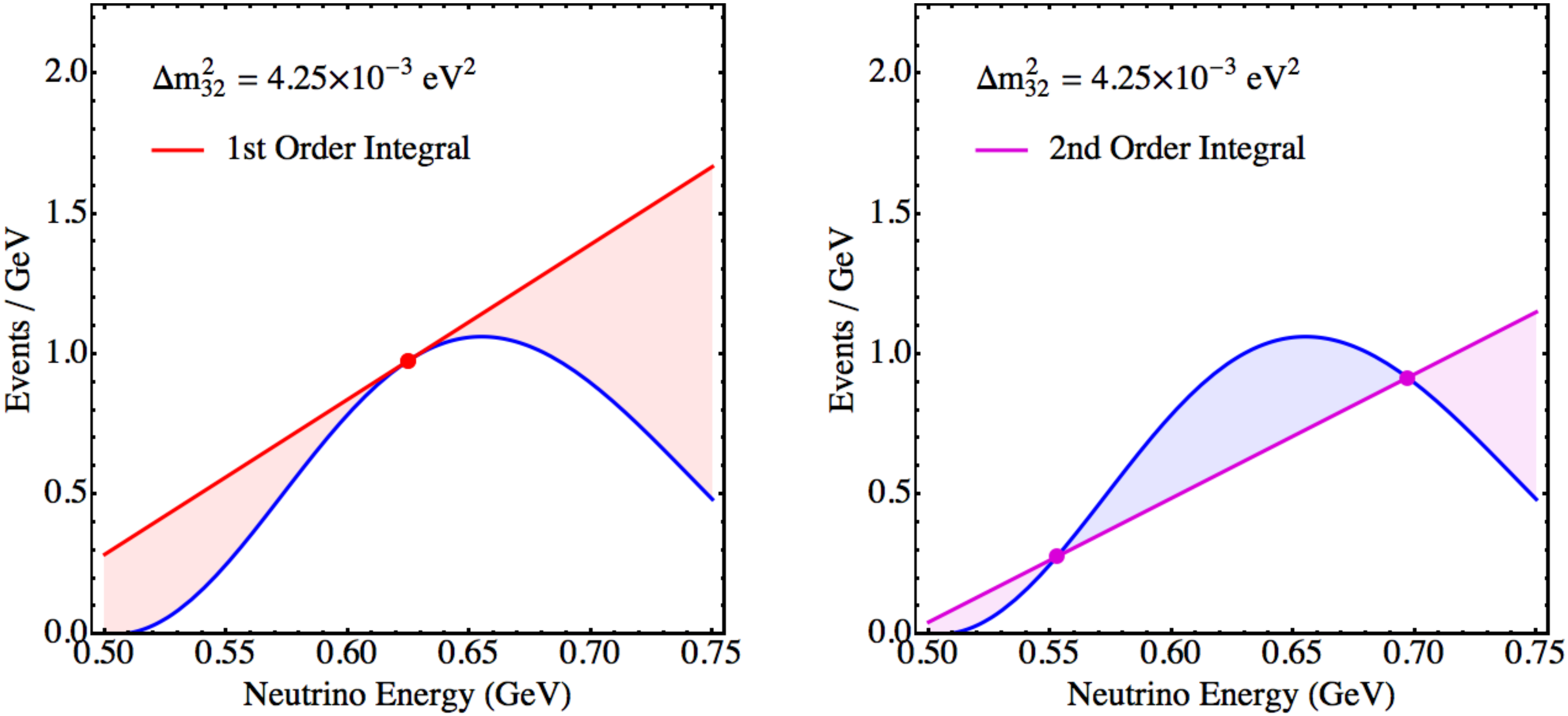
Oscillation Averaging

- Another problem with fast oscillations is computing the average oscillation over a bin
- Simply taking the bin center value is equivalent to a linear approximation of the function



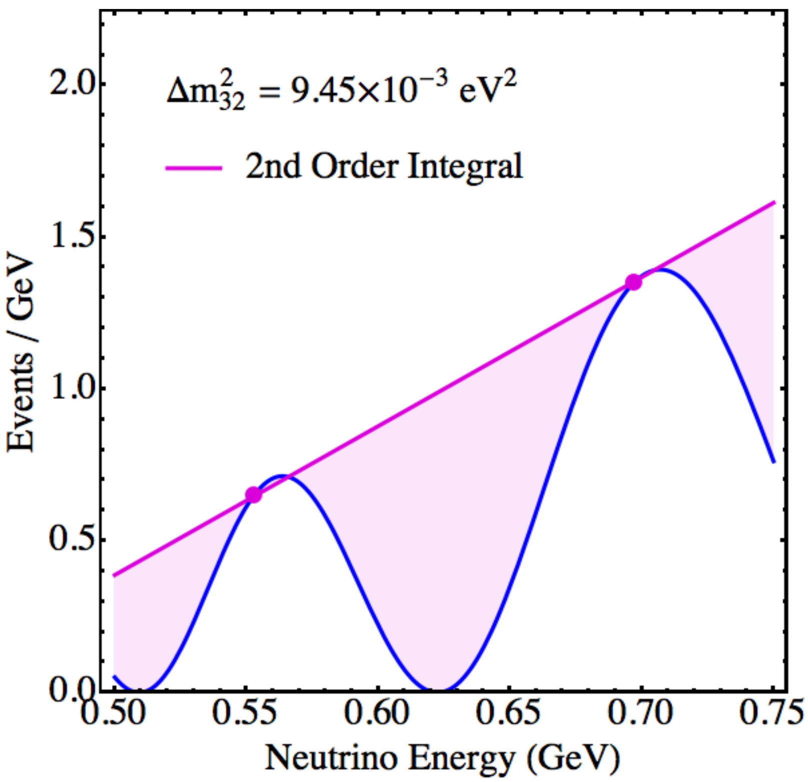
Oscillation Averaging

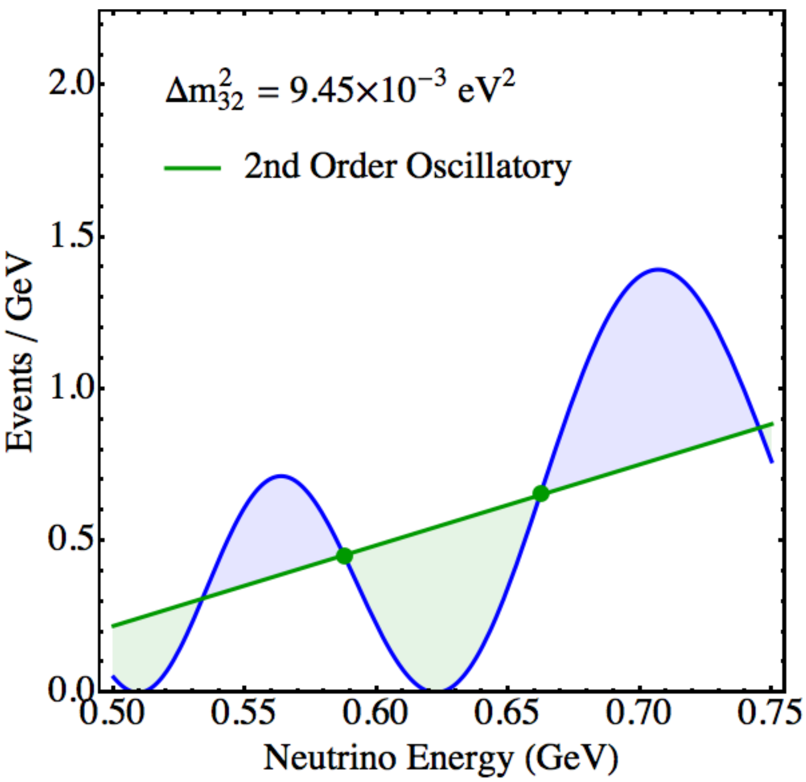
- Another problem with fast oscillations is computing the average oscillation over a bin
- Will fail if function is not approximately linear
- Gaussian quadrature improves this to 3rd order polynomial



Oscillation Averaging

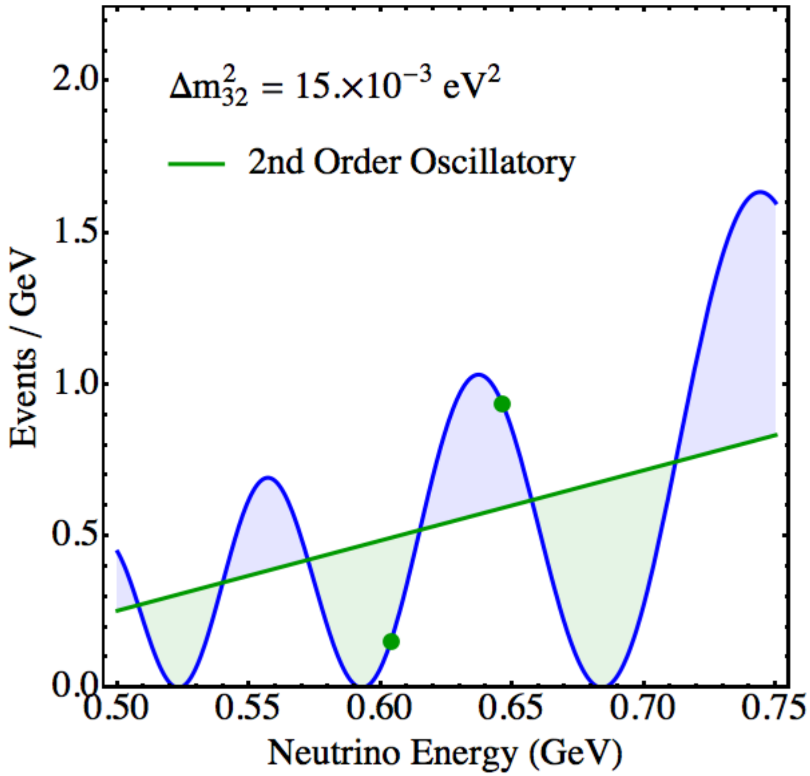
- Another problem with fast oscillations is computing the average oscillation over a bin
- But also fails for oscillating functions
- Can be extended by moving from polynomial to trig functions

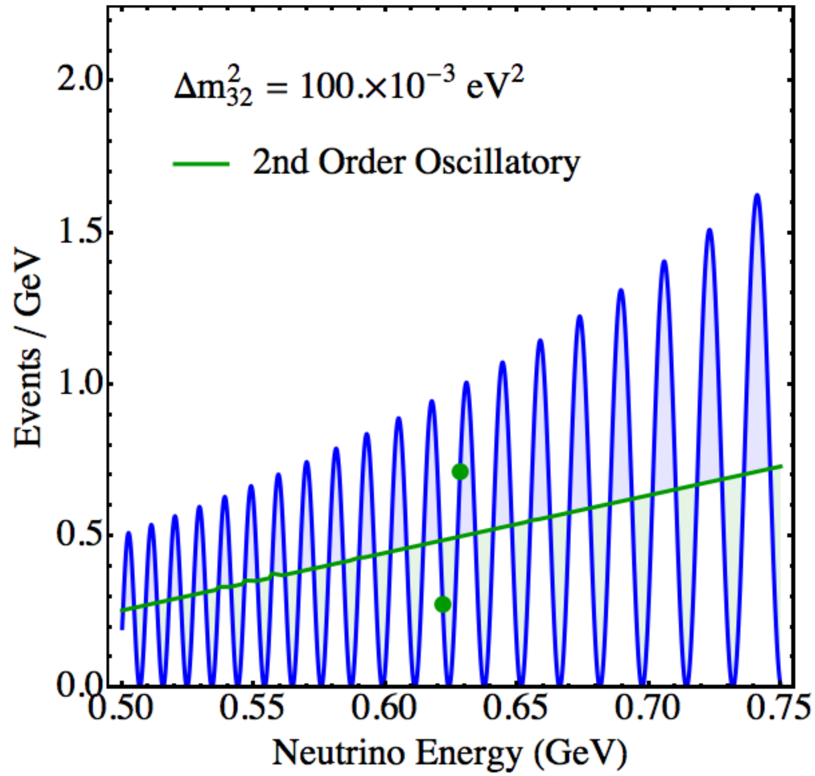




Oscillation Averaging

- Another problem with fast oscillations is computing the average oscillation over a bin
- Works even at high frequencies, but requires known freqs
- Also, only valid for single frequency





Oscillation Averaging

HELP NEEDED

- Solution can be generalized for multiple frequencies
- However, no analytical solution and hard to solve numerically
- Current implementation approximates by assuming frequencies are hierarchical and can be solved independently
- Numerical solution improves on this, but doesn't work reliably

perf: Use improved Gaussian quadrature rule in AvgProb #43

Open joaoabcoelho wants to merge 1 commit into `master` from `dev-avgprob`

Conversation 2 | Commits 1 | Checks 1 | Files changed 1

joaoabcoelho commented 2 weeks ago • edited · Owner

Given that oscillation probabilities typically are of the form:

$$f(x) = a + b\,x + \sum_{i=1}^{n} c_i \cos(k_i x + \phi_i)$$

We can compute the integral of $f(x)$ via a quadrature rule solving for:

$$\int_{x_0-\Delta x}^{x_0+\Delta x} f(x)dx = \frac{1}{2^n}\sum_{j=1}^{2^n} f(x_0 + \sum_{i=1}^{n}(-1)^{\lfloor (j-1)/2^{i-1}\rfloor}\delta x_i)$$

The solution satisfies:

$$\forall j \in \{1 \ldots n\} \;\; \prod_{i=1}^{n} \cos(k_j \delta x_i) = \mathrm{sinc}(k_j \Delta x)$$

Here I'm implementing solutions to this system of equations by Newton's method.

Alternative Approach

HELP NEEDED

- Use perturbation theory to approximate evolution:

$$\boldsymbol{S}(\bar{E} + \xi_E) \approx \bar{\boldsymbol{S}}\, e^{-i\boldsymbol{K}_E\, \xi_E} \quad \text{with} \quad \bar{\boldsymbol{S}} = e^{-i\bar{\boldsymbol{H}}L}$$

- Based on very interesting new paper:
  - https://arxiv.org/abs/2308.00037

From ray to spray: augmenting amplitudes and taming fast oscillations in fully numerical neutrino codes

---

Michele Maltoni

Instituto de Física Teórica (IFT-CFTMAT), CSIC-UAM, Calle de Nicolás Cabrera 13–15, Campus de Cantoblanco, E-28049 Madrid, Spain

E-mail: michele.maltoni@csic.es

- MaCh3 does something similar adapted from SK

AvgProb Precision

HELP NEEDED

- Whatever the solution, we need to be able to estimate the approximation error
- Current estimates are too conservative
- Default precision set to 0.01% but samples too many points
- At analysis level, one needs to optimize the balance between precision and speed in these calculations

Benchmarking

HELP NEEDED

- Compute 100x20 oscillograms for ν_μ -> ν_e
- Using AvgProb to remove fast oscillations

```
Processing StressTest.C...
PMNS_Fast:     Performance = 142 µs/iteration
PMNS_Iter:     Performance =  72 µs/iteration
PMNS_Sterile:  Performance = 940 µs/iteration
PMNS_NSI:      Performance = 155 µs/iteration
PMNS_Deco:     Performance = 267 µs/iteration
PMNS_Decay:    Performance = 241 µs/iteration
PMNS_LIV:      Performance = 328 µs/iteration
PMNS_SNSI:     Performance = 217 µs/iteration
```

- Would be great to have some comparisons with other tools

Conclusion

HELP NEEDED

- Oscillation calculations are the core of many analyses
- Computations are fast, but need to be done trillions of times

- OscProb is an open source option for computing oscillations
- Goal is to be fast and cover many BSM models

- If you have ideas or would like to help with current issues, you are very welcome to contribute to the project in Github

- Integrating OscProb into DUNE tools like MaCh3 would also be very much appreciated

Joao Coelho, Rebekah Pestes, Alba Domi, Simon Bourret, Ushakrhmn, Imaderer, & vicacuen. (2023). joaoabcoelho/OscProb: v2.0.12 (v2.0.12). Zenodo. https://doi.org/10.5281/zenodo.10104847

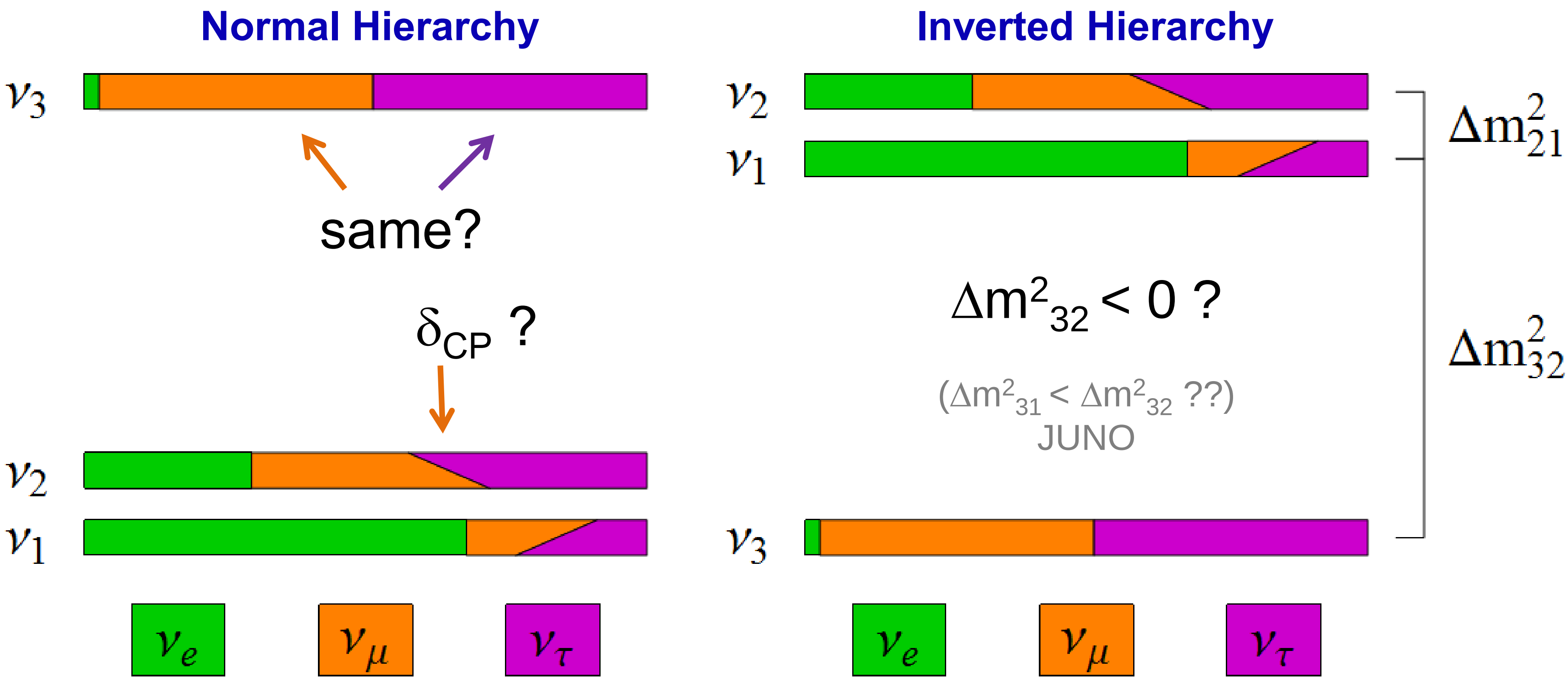
Backup

Missing Pieces

symmetries

$$\sin^2 2\theta \times \sin^2\left(\frac{\Delta m^2 L}{4E}\right)$$

- Is $\theta_{23} = \pi/4$? Underlying symmetry?
- Do neutrinos violate CP? (δ_{CP})
- **What is the mass ordering? (Mass Hierarchy)**



Matter Effects