Algorithms for computing neutrino oscillation probabilities in sharply varying matter potentials such as the Earth are becoming increasingly important. As the next generation of experiments, DUNE and HyperK as well as the IceCube upgrade and KM3NeT, come online, the computational cost for atmospheric and solar neutrinos will continue to increase. To address these issues, we expand upon our previous algorithm for long-baseline calculations to efficiently handle probabilities through the Earth for atmospheric, nighttime solar, and supernova neutrinos. The algorithm is fast, flexible, and accurate. It can handle arbitrary Earth models with two different schemes for varying density profiles. We also provide a c++ implementation of the code called NuFast-Earth along with a detailed user manual. The code intelligently keeps track of repeated calculations and only recalculates what is needed on each successive call which can also help provide significant speed-ups.

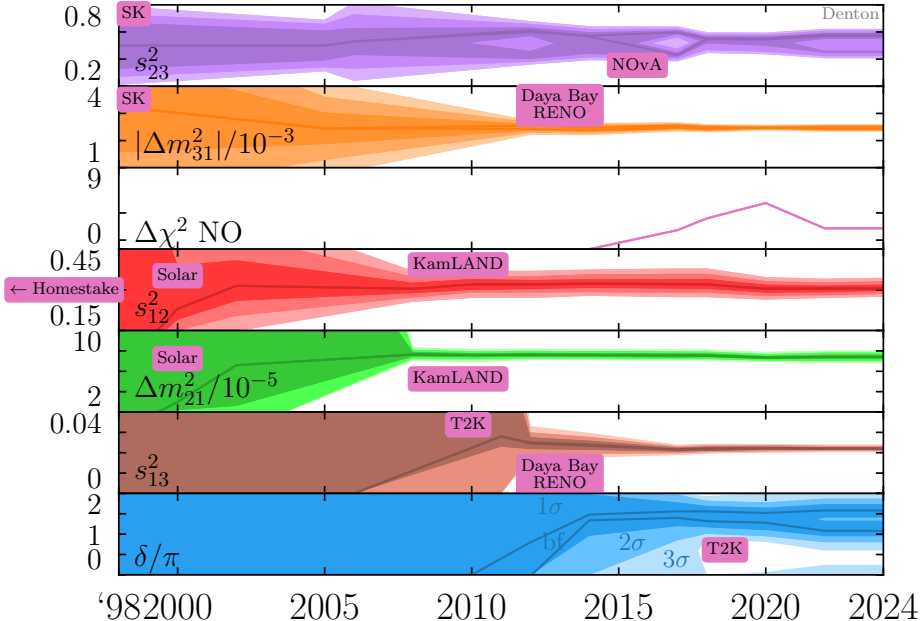
NuFast

Peter B. Denton

IJWSMB

December 11/12, 2025

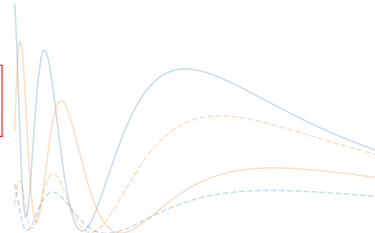
2405.02400 & 2511.04735 with S. Parke
github.com/PeterDenton/NuFast-LBL
github.com/PeterDenton/NuFast-Earth


Denton
0.8
SK
s^2_{23}
0.2
NOvA
4
SK
Daya Bay
RENO
$|\Delta m^2_{31}|/10^{-3}$
1
9
0
$\Delta\chi^2$ NO
0.45
Solar
KamLAND
← Homestake
s^2_{12}
0.15
10
Solar
KamLAND
$\Delta m^2_{21}/10^{-5}$
2
0.04
T2K
s^2_{13}
0
Daya Bay
RENO
2
1σ
1
bf
2σ
3σ
T2K
δ/π
0
'98
2000
2005
2010
2015
2020
2024

The Problem

$$\Delta m^2_{21},\ \Delta m^2_{31}$$
$$s^2_{23},\ s^2_{13},\ s^2_{12}$$
$$\delta$$

$$\Rightarrow$$
in (sharply varying) matter



Many Approaches

Modify vacuum probabilities

- Get the eigenvalues by solving the cubic

Cardano 1545
V. Barger, et al. PRD 22 (1980) 2718

- Get the eigenvectors

H. Zaglauer, K. Schwarzer Z.Phys. C40 (1988) 273
K. Kimura, A. Takamura, H. Yokomakura hep-ph/0205295
PBD, S. Parke, X. Zhang 1907.02534
A. Abdulahi, S. Parke 2212.12565

Advantage: can use existing intuition about the parameters

Atmospheric/Solar: Many Approaches

Solve the Schrödinger equation

$$i\frac{d}{dt}|\nu\rangle = H(t)|\nu\rangle$$

For region j where H is approximately constant: H_j (constant density)

$$\mathcal{A}_j = e^{-iH_jL} \qquad P(\nu_\alpha \to \nu_\beta) = \left| \left[\prod_j \mathcal{A}_j \right]_{\beta\alpha} \right|^2$$

Exponential requires computing eigenvalues and eigenvectors of H_i,
for good precision through the Earth, repeat this many times

Fermilab computing experts bolster NOvA evidence, 1 million cores consumed

July 3, 2018 | Marcia Teckenbrock

Share Tweet Email

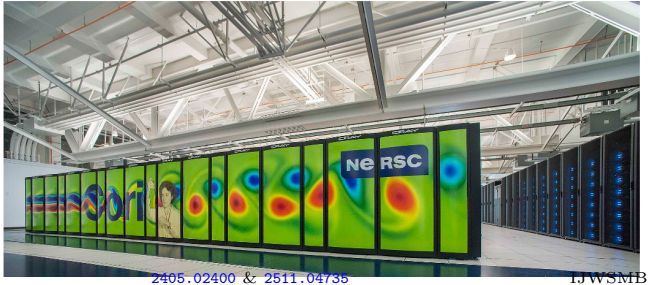
Array

How do you arrive at the physical laws of the universe when you're given experimental data on a renegade particle that interacts so rarely with matter, it can cruise through light-years of lead? You call on the power of advanced computing.

The NOvA neutrino experiment, in collaboration with the Department of Energy's Scientific Discovery through Advanced Computing (SciDAC-4) program and the HEPCloud program at DOE's Fermi National Accelerator Laboratory, was able to perform the largest-scale analysis ever to support the recent evidence of antineutrino oscillation, a phenomenon that may hold clues to how our universe evolved.

Using Cori, the newest supercomputer at the National Energy Research Scientific Computing Center (NERSC), located at Lawrence Berkeley National Laboratory, NOvA used over 1 million computing cores, or CPUs, between May 14 and 15 and over a short timeframe one week later. This is the largest number of CPUs ever used concurrently over this duration — about 54 hours — for a single high-energy physics experiment. This unprecedented amount of computing enabled scientists to carry out some of the most complicated techniques used in neutrino physics, allowing them to dig deeper into the seldom seen interactions of neutrinos. This Cori allocation was more than 400 times the amount of Fermilab computing allocated to the NOvA experiment and 50 times the total computing capacity at Fermilab allocated for all of its rare-physics experiments. A continuation of the analysis was performed on NERSC's Cori and Edison supercomputers one week later. In total, nearly 35 million core-hours were consumed by NOvA in the 54-hour period. Executing the same analysis on a single desktop computer would take 4,000 years.

FNAL Newsroom



Monte-Carlo Estimates of Statistical Significances

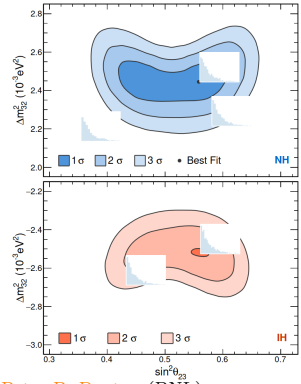
Wilks' theorem is often wrong

At each point in parameter space, simulate the experiment many times

"many" means $\gg 1/p$ for a desired p-value

This is sometimes called Feldman-Cousins

G. Feldman, R. Cousins physics/9711021
This isn't actually what was novel in the FC paper



Study found most of
the time was spent
computing probabilities

NOvA is a $\sim 3\sigma$ experiment,
but DUNE will be a $\gtrsim 5\sigma$ experiment!

How to Achieve Speed

1. Avoid costly operations
   - `sin`, `cos` (and inverse functions) are very slow
   - `sqrt` is quite slow, but not as bad as trigs
   - Division is slower than multiplication ($0.2x$ may be faster than $x/5$)
   - Avoid complex numbers: addition: $2\times$, multiplication: $4\times$, division: $> 8\times$
2. Reduce repeated calculations
   - Compute $\frac{L}{4E}$ in the correct units once
   - Compute each of the three $\sin \frac{\Delta m^2_{ij} L}{4E}$ once

Optimal Structure of the Probability for Long-Baseline

1. Amplitude requires four trig functions of kinematic variables ($\Delta m^2_{ij} L/4E$) ×
2. Writing the probabilities out requires three trig functions ✓
3. Disappearance structure is straightforward:

$$P_{\alpha\alpha} = 1 - 4\sum_{i>j} |V_{\alpha i}|^2 |V_{\alpha j}|^2 \sin^2 \frac{\Delta\lambda_{ij} L}{4E}$$

H in matter has eigenvalues λ_i and eigenvectors $V_{\alpha i}$

Optimal Structure of the Probability for Long-Baseline

4. Appearance structure:
   T conserving:

$$P^{TC}_{\mu e} = 2\sum_{i>j}(|V_{\tau k}|^2 - |V_{\mu i}|^2|V_{ej}|^2 - |V_{\mu j}|^2|V_{ei}|^2)\sin^2\frac{\Delta\lambda_{ij}L}{4E}$$

Fun fact:
$2\Re(V_{\alpha i}V^*_{\beta j}V^*_{\alpha j}V_{\beta i})$
$= |V_{\alpha k}|^2|V_{\beta k}|^2 - |V_{\alpha i}|^2|V_{\beta i}|^2 - |V_{\alpha j}|^2|V_{\beta j}|^2$
$= |V_{\gamma k}|^2 - |V_{\alpha i}|^2|V_{\beta j}|^2 - |V_{\alpha j}|^2|V_{\beta i}|^2$

   T violating:

$$P^{TV}_{\mu e} = -8J\frac{\Delta m^2_{21}\Delta m^2_{31}\Delta m^2_{32}}{\Delta\lambda_{21}\Delta\lambda_{31}\Delta\lambda_{32}}\sin\frac{\Delta\lambda_{21}L}{4E}\sin\frac{\Delta\lambda_{31}L}{4E}\sin\frac{\Delta\lambda_{32}L}{4E}$$

C. Jarlskog PRL 55, 1039 (1985)

Leverages NHS identity:
V. Naumov IJMP 1992
P. Harrison, W. Scott hep-ph/9912435

Account for Matter in Long-Baseline

1. Need the eigenvalues λ_i
2. For eigenvectors, naively need $\Re(V_{\alpha i} V^*_{\beta j} V^*_{\alpha j} V_{\beta i})$
3. Given our form, need only the $|V_{\alpha i}|^2$ and J in vacuum
   - Don't need any phase information of the eigenvectors!

Leverages PBD, S. Parke, X. Zhang 1907.02534

4. Can compute the $|V_{\alpha i}|^2$ from the λ_i and submatrix eigenvalues (requires only a square root) using Eigenvector-Eigenvalue Identity

$$|V_{\alpha i}|^2 = \frac{\prod_{k=1}^{n-1}(\lambda_i - \xi_k^\alpha)}{\prod_{k=1;k\neq i}^{n}(\lambda_i - \lambda_k)}$$

See e.g. PBD, S. Parke, T. Tao, X. Zhang 1908.03795
Can actually avoid the $\sqrt{\ }$ in practice

Eigenvalues are Hard

The eigenvalues in matter λ_i depend on S (and other things):

$$S = \cos\left\{\frac{1}{3}\cos^{-1}\left[\frac{2A^3 - 9AB + 27C}{2\left(A^2 - 3B\right)^{3/2}}\right]\right\}$$

where

$$A = \sum \lambda_i = \Delta m^2_{21} + \Delta m^2_{31} + a$$
$$B = \sum_{i>j} \lambda_i \lambda_j = \Delta m^2_{21}\Delta m^2_{31} + a[\Delta m^2_{21}(1 - |U_{e2}|^2) + \Delta m^2_{31}(1 - |U_{e3}|^2)]$$
$$C = \prod \lambda_i = a\Delta m^2_{21}\Delta m^2_{31}|U_{e1}|^2$$

Approximate Eigenvalues

1. Instead, approximate one eigenvalue
   - λ_3 is best because is never parametrically small and easy to approximate
2. From DMP:

$$\lambda_3 \approx \Delta m^2_{31} + \frac{1}{2}\Delta m^2_{ee}\left(x - 1 + \sqrt{(1-x)^2 + 4x s^2_{13}}\right)$$

$$x \equiv \frac{a}{\Delta m^2_{ee}} \qquad \Delta m^2_{ee} \equiv \Delta m^2_{31} - s^2_{12}\Delta m^2_{21}$$

H. Minakata, S. Parke 1505.01826
PBD, H. Minakata, S. Parke 1604.08167
H. Nunokawa, S. Parke, R. Funchal hep-ph/0503283

3. Get other two eigenvalues by picking two of A, B, C conditions

Requires one more $\sqrt{}$

The Approximation

- This is the only approximation used in the entire approach
- In vacuum the approximation returns to the correct value

Many approximations in the literature are not correct in vacuum limit
See G. Barenboim, PBD, S. Parke, C. Ternes 1902.00517

- In fact can iteratively improve λ_3 with rapid convergence via Newton's method:

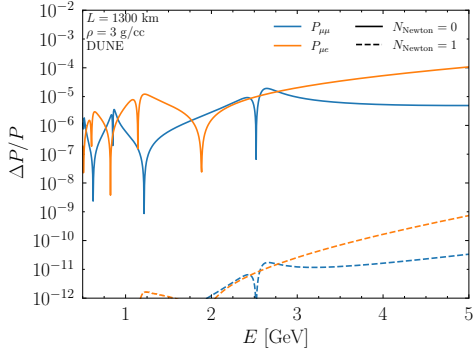
$$\lambda_3 \to \lambda_3 - \frac{X(\lambda_3)}{X'(\lambda_3)}$$

$$X(\lambda) = \lambda^3 - A\lambda^2 + B\lambda - C = 0$$

  - Precision improvement starts at 10^{-5} for the first step
  - The improvement is quadratic thereafter
- One line of code, just loop as many times as desired

LBL Precision

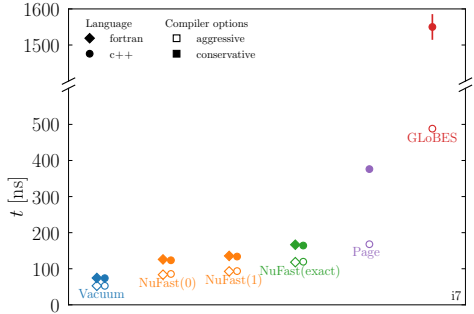
Is this approximation okay?
DUNE requires $\lesssim 1\%$ level precision



Slightly better for HK
$\sim 10^{-10}$ for JUNO
See backups

LBL Speed

Is this algorithm fast?
How does it compare?



- "Conservative" = default, `-O0`
- "Aggressive" = `-Ofast` and `-ffast-math`
- Some variation expected due to architecture

See also J. Page 2309.06900
and P. Huber, et al. hep-ph/0701187

NuFast-LBL: the Code

- Code is on github: github.com/PeterDenton/NuFast-LBL
- Implementations in c++ and f90
- Easy to use and there are comments!

```
// NuFast.cpp
// Calculate the probabilities:
Probability_Matter_LBL(s12sq, s13sq, s23sq, delta,
    Dmsq21, Dmsq31, L, E, rho, Ye, N_Newton,
    &probs_returned);
// Print out the probabilities to terminal
for (int alpha = 0; alpha < 3; alpha++)
{
   for (int beta = 0; beta < 3; beta++)
   {
      printf("%d %d %g\n", alpha, beta,
          probs_returned[alpha][beta]);
   } // beta, 3
} // alpha, 3
```

- $\bar{\nu}$: $E < 0$; IO: $\Delta m^2_{31} < 0$
- Folder called Benchmarks to make the plots in the paper
- Used in NuOscillator, MaCh3, and GUNDAM

Optimal Structure of the Probability for Atmospherics

1. Cannot apply the previous tricks through the Earth
2. Must compute the amplitude in Each layer

$$\prod_j \mathcal{A}_j$$

Optimal Structure of the Probability for Atmospherics

3. Work in the tilde basis with θ_{23} and δ pulled out
   - Can shift δ from θ_{13} to θ_{23}

$$\begin{pmatrix} 1 & & \\ & c_{23} & s_{23} \\ & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & & s_{13}e^{-i\delta} \\ & 1 & \\ -s_{13}e^{i\delta} & & c_{13} \end{pmatrix} R_{12} \to \begin{pmatrix} 1 & & \\ & c_{23} & s_{23}e^{i\delta} \\ & -s_{23}e^{-i\delta} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & & s_{13} \\ & 1 & \\ -s_{13} & & c_{13} \end{pmatrix} R_{12}$$

Optimal Structure of the Probability for Atmospherics

3. Work in the tilde basis with θ_{23} and δ pulled out
   - Can shift δ from θ_{13} to θ_{23}

$$\begin{pmatrix} 1 & & \\ & c_{23} & s_{23} \\ & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & & s_{13}e^{-i\delta} \\ & 1 & \\ -s_{13}e^{i\delta} & & c_{13} \end{pmatrix} R_{12} \to \begin{pmatrix} 1 & & \\ & c_{23} & s_{23}e^{i\delta} \\ & -s_{23}e^{-i\delta} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & & s_{13} \\ & 1 & \\ -s_{13} & & c_{13} \end{pmatrix} R_{12}$$

   - U_{23} commutes with the matter potential

$$(2E)H_f = U_{23}(\theta_{23},\delta)\left[R_{13}R_{12}\begin{pmatrix} 0 & & \\ & \Delta m^2_{21} & \\ & & \Delta m^2_{31} \end{pmatrix} R_{12}^T R_{13}^T + \begin{pmatrix} a & & \\ & 0 & \\ & & 0 \end{pmatrix}\right] U_{23}^\dagger(\theta_{23},\delta)$$

$$H_f = U_{23}(\theta_{23},\delta)\tilde{H}U_{23}^\dagger(\theta_{23},\delta)$$

Optimal Structure of the Probability for Atmospherics

3. Work in the tilde basis with θ_{23} and δ pulled out
   - Can shift δ from θ_{13} to θ_{23}

$$\begin{pmatrix} 1 & & \\ & c_{23} & s_{23} \\ & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & & s_{13}e^{-i\delta} \\ & 1 & \\ -s_{13}e^{i\delta} & & c_{13} \end{pmatrix} R_{12} \to \begin{pmatrix} 1 & & \\ & c_{23} & s_{23}e^{i\delta} \\ & -s_{23}e^{-i\delta} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & & s_{13} \\ & 1 & \\ -s_{13} & & c_{13} \end{pmatrix} R_{12}$$

   - U_{23} commutes with the matter potential

$$(2E)H_f = U_{23}(\theta_{23},\delta)\left[R_{13}R_{12}\begin{pmatrix} 0 & & \\ & \Delta m^2_{21} & \\ & & \Delta m^2_{31} \end{pmatrix} R_{12}^T R_{13}^T + \begin{pmatrix} a & & \\ & 0 & \\ & & 0 \end{pmatrix}\right] U_{23}^\dagger(\theta_{23},\delta)$$

$$H_f = U_{23}(\theta_{23},\delta)\tilde{H}U_{23}^\dagger(\theta_{23},\delta)$$

$$\mathcal{A} = \prod_j \mathcal{A}_j = \prod_j e^{-iH_{f,j}L_j} = U_{23}(\theta_{23},\delta)\left(\prod_j e^{-i\tilde{H}_j L_j}\right) U_{23}^\dagger(\theta_{23},\delta)$$

$$U_{23}U_{23}^\dagger = \mathbb{1}$$

Optimal Structure of the Probability for Atmospherics

3. Work in the tilde basis with θ_{23} and δ pulled out
   - Can shift δ from θ_{13} to θ_{23}

$$\begin{pmatrix} 1 & & \\ & c_{23} & s_{23} \\ & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & & s_{13}e^{-i\delta} \\ & 1 & \\ -s_{13}e^{i\delta} & & c_{13} \end{pmatrix} R_{12} \to \begin{pmatrix} 1 & & \\ & c_{23} & s_{23}e^{i\delta} \\ & -s_{23}e^{-i\delta} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & & s_{13} \\ & 1 & \\ -s_{13} & & c_{13} \end{pmatrix} R_{12}$$

   - U_{23} commutes with the matter potential

$$(2E)H_f = U_{23}(\theta_{23},\delta)\left[R_{13}R_{12}\begin{pmatrix} 0 & & \\ & \Delta m^2_{21} & \\ & & \Delta m^2_{31} \end{pmatrix} R_{12}^T R_{13}^T + \begin{pmatrix} a & & \\ & 0 & \\ & & 0 \end{pmatrix}\right] U_{23}^\dagger(\theta_{23},\delta)$$

$$H_f = U_{23}(\theta_{23},\delta)\tilde{H}U_{23}^\dagger(\theta_{23},\delta)$$

$$\mathcal{A} = \prod_j \mathcal{A}_j = \prod_j e^{-iH_{f,j}L_j} = U_{23}(\theta_{23},\delta)\left(\prod_j e^{-i\tilde{H}_j L_j}\right) U_{23}^\dagger(\theta_{23},\delta)$$

$\tilde{H}$ is real! And doesn't depend on θ_{23} or δ

Reduces many unnecessary computations

$U_{23}U_{23}^\dagger = \mathbb{1}$

Further Reductions

- Calculate eigenvalues λ_i as in NuFast-LBL: exactly or approximately with Newton-Raphson corrections
- Further unitarity reductions:

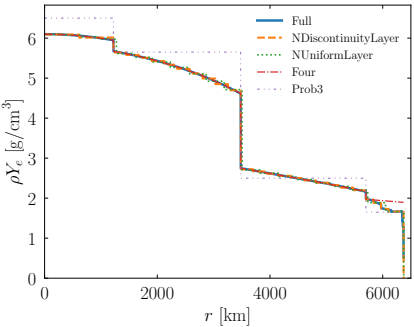
$$\mathcal{A}_{\alpha\beta} = \delta_{\alpha\beta} + V_{\alpha 2}V^*_{\beta 2}(e^{-i\Delta\lambda_{21}L/(2E)} - 1) + V_{\alpha 3}V^*_{\beta 3}(e^{-i\Delta\lambda_{31}L/(2E)} - 1)$$

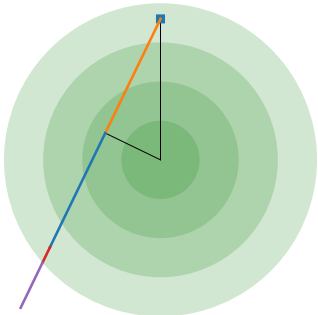
- In the tilde basis, the $\tilde{V}\tilde{V}^*$ terms are real
- Use adjugate matrices and advanced Eigenvector-Eigenvalue Identity

PBD, S. Parke, T. Tao, X. Zhang 1908.03795

$$\tilde{V}_{\alpha i}\tilde{V}_{\beta i} = \frac{\mathrm{Adj}[\lambda_i \mathbb{1} - (2E)\tilde{H}]}{\prod_{k\neq i}(\lambda_i - \lambda_k)}$$

Earth Trajectories





- Various Earth density profiles
- Varying or constant density in each shell

Trajectory goes from:

1. Production height to surface
2. Surface to detector depth
3. Detector depth to deepest point
4. Deepest point to detector depth: this is the transpose of #3

Example Earth Model

```
// Earth.cpp
PREM_Prob3::PREM_Prob3()
{
    n_discontinuities = 4;
    discontinuities.reserve(n_discontinuities);
    discontinuities[0] = 1220.;
    discontinuities[1] = 3480.;
    discontinuities[2] = 5701.;
    discontinuities[3] = 6371.;
    Ye = 0.5;
    constant_shells = true;
}
double PREM_Prob3::rhoYe(double r) // g/cm^3
{
    assert(r >= 0);
    if (r > 6371.) return 0.;
    if (r > 5701.) return Ye * 3.3;
    if (r > 3480.) return Ye * 5.;
    if (r > 1220.) return Ye * 11.3;
    return Ye * 13.;
}
```

Many Earth models coded up

- `PREM_NDiscontinuityLayer` takes four integers indicating the number of layers in each major region: inner core, outer core, inner mantle, and outer mantle
  `PREM_NDiscontinuityLayer earth_density(2, 10, 10, 5);`

  Can also take one integer and apply that to each of the four regions
- `PREM_NUniformLayer` takes one integer splitting the whole Earth into N layers

  Can struggle near sharp boundaries
- `PREM_Prob3` Four constant layers matching `prob3` code
- `PREM_Full` Uses **averaging** across major regions
- `PREM_Four` Same as `PREM_Full` but only has four total regions ignoring the small fluctuations near the surface

Two Earth Model Calculation Techniques

Constant shells:

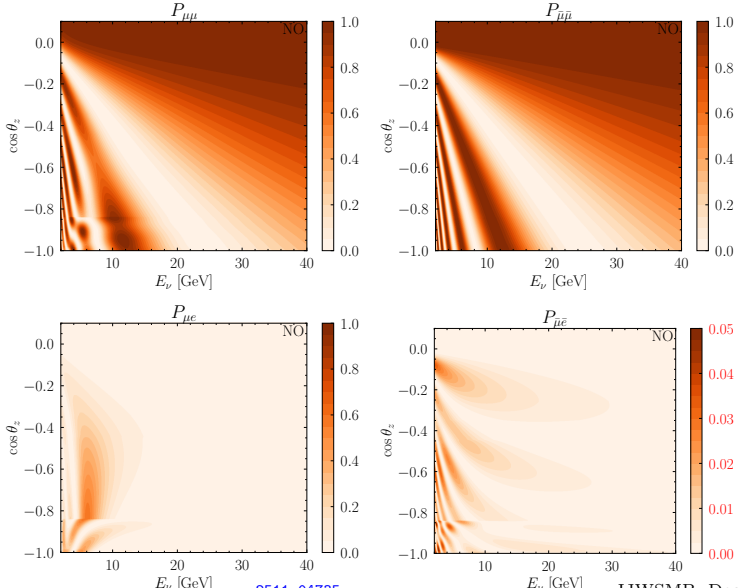
1. Contains some number of discrete shells
2. Compute the eigenvalues and eigenvectors in each shell once
3. Good if there are many energy and zenith angles

Averaging over varying shells:

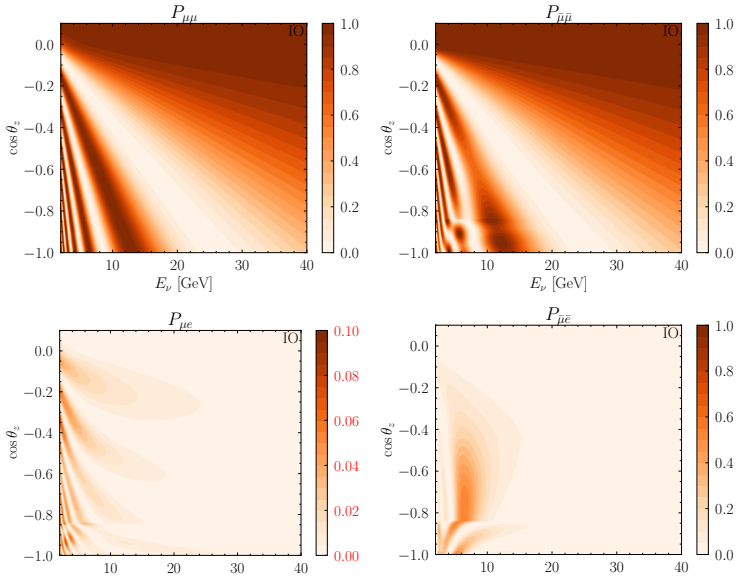
1. (Typically) uses only several (4 or 10) major regions
2. Computes the average density in each region for each trajectory
3. Handles the fact that e.g. a trajectory through the middle of the core sees a larger average inner core density than one that skims the inner core

In general we find that the averaging approach is not beneficial

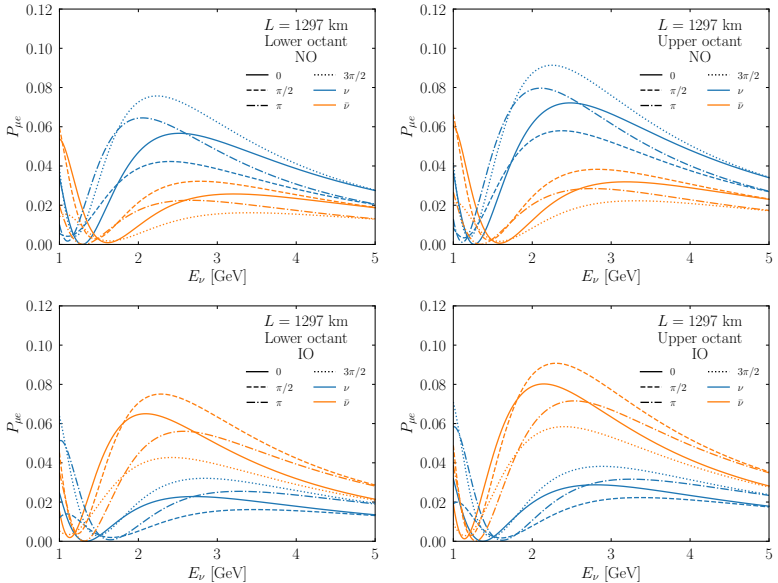
Atmospheric Results: NO



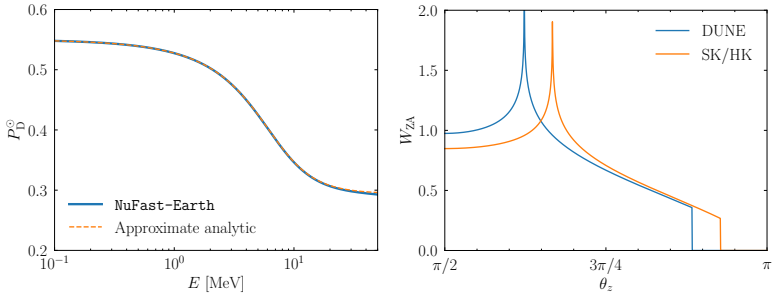
Atmospheric Results: IO



NuFast-Earth LBL Validation



NuFast-Earth Solar Neutrino Validation

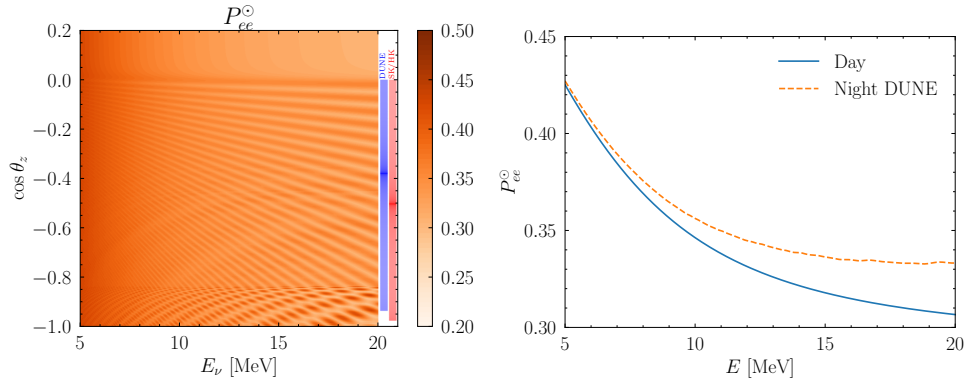


Analytically understand the small deviation at high energy

Nighttime Solar Neutrinos

At night solar neutrinos experience partial regeneration:
There are more ν_e's from the Sun at night than during the day!

SuperK has $\sim 2\sigma$ evidence for this effect; DUNE and HK aim to measure it well



Solar Neutrinos

We compute and provide code for:

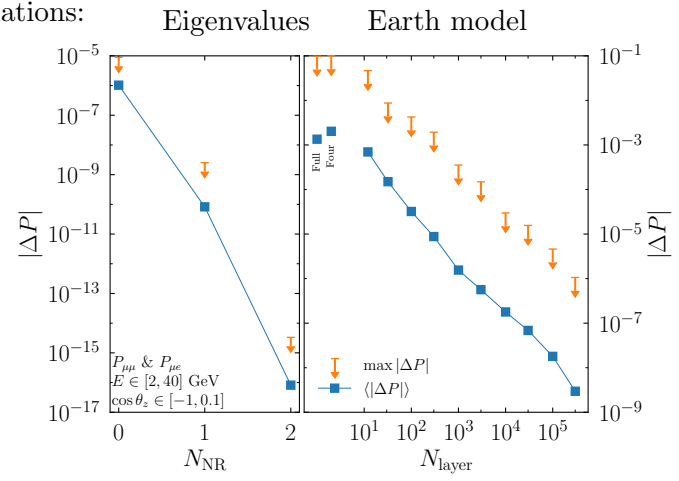
- The Hamiltonian at a given production density
  - This is just the eigenvectors at the Sun's density
  - From here it is trivial to compute the day time solar neutrino oscillation probability
- The nighttime probability
- The annual averaged weight function

We do not provide code for:

- The solar model

NuFast-Earth Precision

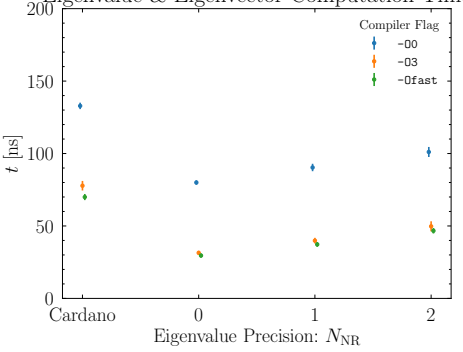
Two approximations:

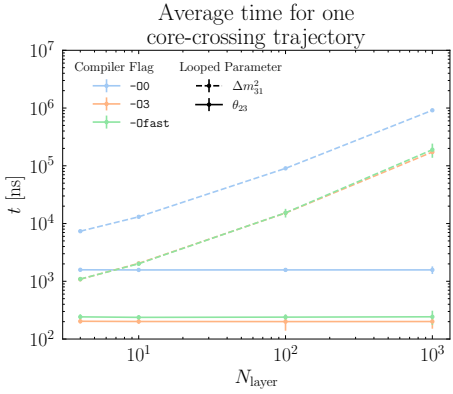


"True" has exact eigenvalues and 1M layers

Calculate $|\Delta P|$ across 100×100 grid in energy and angle

NuFast-Earth Speed





`nuSQuIDS` takes >1M ns per trajectory

C. Argüelles, J. Salvado, C. Weaver 2112.13804

`OscProb` takes 40 000 ns per trajectory at 44 layers (compare to 6000 ns or 200 ns)

J. Coelho, R. Pestes, et al. github.com/joaoabcoelho/OscProb

Other codes are designed with different goals in mind

NuFast-Earth Flow

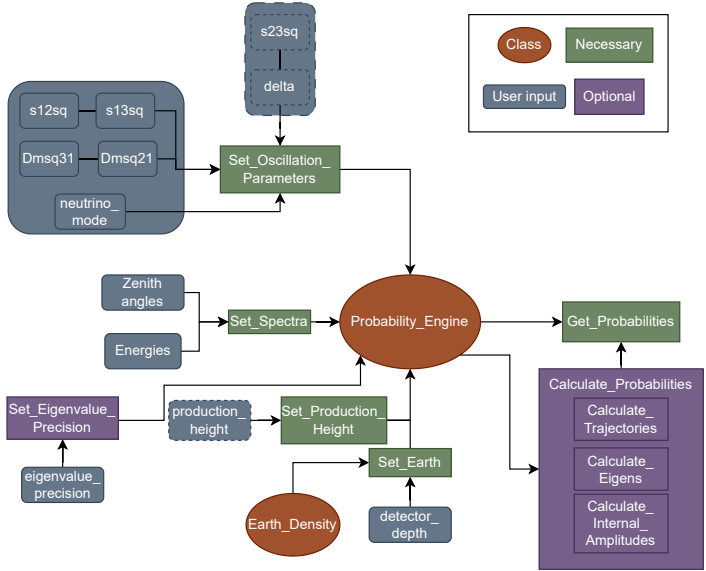
```
// Initialize the engine
Probability_Engine probability_engine;

// Set the oscillation parameters to nu-fit 6 best fit
    values:
probability_engine.Set_Oscillation_Parameters(0.307,
    0.02195, 0.561, 177 * M_PI / 180, 7.49e-5, 2.534e-3,
    true); //(s12sq, s13sq, s23sq, delta, Dmsq21, Dmsq31,
     neutrino_mode)

// Set energy and zenith angle arrays
std::vector<double> energies = {1, 2, 3, 4, 5}; // GeV
std::vector<double> coszs = {-1, -0.5, 0, 1}; // core-
    crossing to horizontal to down-going
probability_engine.Set_Spectra(energies, coszs);

// Create Earth model instance
PREM_NDiscontinuityLayer earth_density(2, 10, 10, 5); //
    2 layers in the inner core, 10 layers in the outer
    core, 10 layers in the inner mantle, and 5 layers in
    the outer mantle
// Set Earth details
probability_engine.Set_Earth(2, &earth_density); //
    detector depth in km
// Set production height (optional)
probability_enginge.Set_Production_Height(10); // km,
    recalculations after changing this are fast

// Do the calculations
std::vector<std::vector<Matrix3r>> probabilities =
    probabilitie_engine.Get_Probabilities();
```



Dashed boxes = fast!

NuFast-Earth Recommendations

Many choices in how to use `NuFast-Earth`
What is the best way for most cases?

1. Eigenvalue precision: set the number of Newton-Raphson corrections to 1

```
probability_engine.Set_Eigenvalue_Precision(1);
```

2. Earth model: 2, 10, 10, 5 layers in each of the four main zones

```
PREM_NDiscontinuityLayer earth_density(2, 10, 10, 5);
```

3. Looping order: put θ_{23}, δ, and production height on the innermost loops

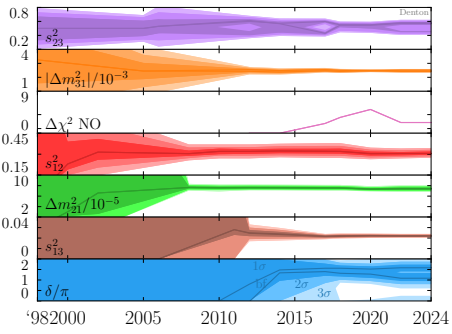
Summary

- Computing neutrino oscillations fast is important
- LBL algorithm contains several innovations:
  - Approximate λ_3 and rapid convergence, if desired
  - Get all eigenvalues from λ_3 via A, B, C
  - Get TV part from NHS identity
  - Structure TC parts to not require phase information
  - Get $|V_{\alpha i}|^2$ from Eigenvalue-Eigenvector Identity
  - Speed up of ∼few to order of magnitude
- Atmospheric code is *fast*
  - Pull out θ_{23} and δ for caching
  - Use adjugate algorithm for eigenvectors
  - Structure the Earth trajectory carefully
  - Speed up of 1-3 orders of magnitude

LBL code is already implemented by experimentalists
they are working on atmospherics

Backups

References



SK hep-ex/9807003

M. Gonzalez-Garcia, et al. hep-ph/0009350

M. Maltoni, et al. hep-ph/0207227

SK hep-ex/0501064

SK hep-ex/0604011

T. Schwetz, M. Tortola, J. Valle 0808.2016

M. Gonzalez-Garcia, M. Maltoni, J. Salvado 1001.4524

T2K 1106.2822

D. Forero, M. Tortola, J. Valle 1205.4018

D. Forero, M. Tortola, J. Valle 1405.7540

P. de Salas, et al. 1708.01186

F. Capozzi et al. 2003.08511

Scope

1. All 9 channels ($\nu_\alpha \to \nu_\beta$)
   - Atmospherics include ν_τ appearance
   - $\nu_\tau \to \nu_\beta$ channels are not needed, but come from free from unitarity
2. Different energies, zenith angles, and Earth models
3. Production height, detector depth
4. ν and $\bar{\nu}$
5. NO and IO
6. Oscillation parameters are mostly known
   - Don't need to consider e.g. $\Delta m^2_{21} > |\Delta m^2_{31}|$ or $\theta_{23} \sim 10^\circ$

All 9 Channels

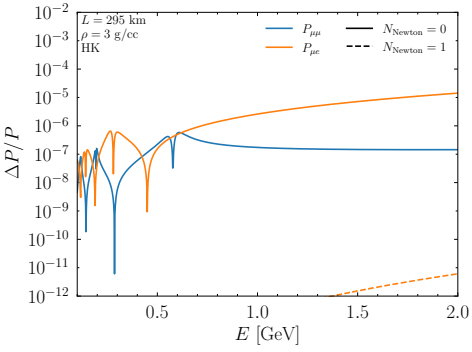
Given P_{ee}, $P_{\mu\mu}$, $P^{TC}_{\mu e}$, and $P^{TV}_{\mu e}$:

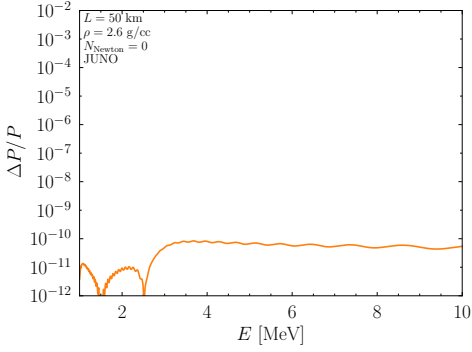
	$P_{\alpha e}$	$P_{\alpha\mu}$	$P_{\alpha\tau}$
$P_{e\beta}$	P_{ee}	$P^{TC}_{\mu e} - P^{TV}_{\mu e}$	$1 - P_{ee} - P^{TC}_{\mu e} + P^{TV}_{\mu e}$
$P_{\mu\beta}$	$P^{TC}_{\mu e} + P^{TV}_{\mu e}$	$P_{\mu\mu}$	$1 - P_{\mu\mu} - P^{TC}_{\mu e} - P^{TV}_{\mu e}$
$P_{\tau\beta}$	$1 - P_{ee} - P^{TC}_{\mu e} - P^{TV}_{\mu e}$	$1 - P_{\mu\mu} - P^{TC}_{\mu e} + P^{TV}_{\mu e}$	$-1 + P_{ee} + P_{\mu\mu} + 2P^{TC}_{\mu e}$

Total LBL Approach

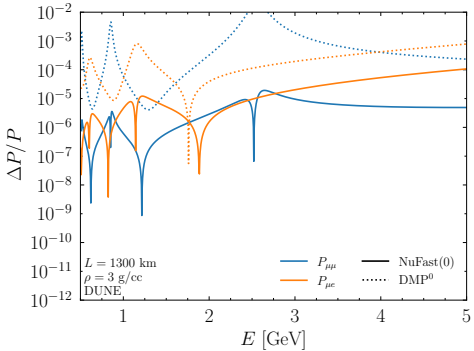
1. Inputs: 6 oscillation parameters, experimental details (L, E, ρ, Y_e)
2. Calculate λ_3 approximately
   - Iteratively improve with Newton's method, if desired
3. Calculate the $|V_{\alpha i}|^2$'s with the Eigenvector-Eigenvalue Identity
4. Calculate the sines of the kinematic terms
5. Calculate the T violating term with the NHS identity
6. Calculate key probabilities: P_{ee}, $P_{\mu\mu}$, and $P_{\mu e}$
7. Calculate remaining probabilities

Precision


$L = 295$ km
$\rho = 3$ g/cc
HK
$P_{\mu\mu}$
$P_{\mu e}$
$N_{\text{Newton}} = 0$
$N_{\text{Newton}} = 1$
$\Delta P/P$
E [GeV]


$L = 50$ km
$\rho = 2.6$ g/cc
$N_{\text{Newton}} = 0$
JUNO
$\Delta P/P$
E [MeV]

Comparison to DMP



DMP: PBD, H. Minakata, S. Parke 1604.08167