FlatPPL 0.1 / NEUTRINO PHYSICS

Neutrino oscillations

TWO-FLAVOUR VACUUM APPEARANCE

$$P = \sin^2(2\theta)\sin^2\Phi$$

COMPLETE EXAMPLE

Appearance in four energy bins

```
%%%
Two-flavour vacuum appearance, synthetic counts.
%%%
flatppl_compat = "0.1"
energy = [0.45, 0.6, 0.9, 1.2]
baseline = 295.0
trials = [1000, 1000, 1000, 1000]
appearance_data = [67, 87, 65, 43]

# GeV, km, eV^2, and radians.
phase(E, d, L) = 1.266932679039099 * d * L / E
appearance(E, t, d, L) = (
    sin(2 * t)^2 * sin(phase(E, d, L))^2)

theta ~ Uniform(interval(0.1, 0.2))
dm2 ~ Uniform(interval(0.002, 0.003))
probability = appearance.(energy, theta,
    dm2, baseline)
y ~ Binomial.(trials, probability)

# Infer angle and mass-squared splitting.
prior = lawof(record(theta = theta, dm2 = dm2))
K = kernelof(record(y = y), theta, dm2)
L = likelihoodof(K, record(y = appearance_data))
posterior = bayesupdate(L, prior)

point = record(theta = 0.15, dm2 = 0.0025)
log_L = logdensityof(L, point)
f_prob = functionof(probability, theta, dm2)
prob_at_point = f_prob(point)
```

Values at the stated point

```
pred = f_prob(theta = 0.15, dm2 = 0.0025)
```

At zero baseline or zero mixing, all four probabilities vanish.

Energy (GeV)	0.45	0.6	0.9	1.2
Probability	0.06685	0.08732	0.06481	0.04308
Observed fraction	0.067	0.087	0.065	0.043

Fold in energy response

```
true_mean = trials .* probability
M = rowstack([[0.9, 0.8, 0.1, 0.0],
              [0.1, 0.2, 0.9, 1.0]])
detected = M * true_mean
```

Each column maps one true energy to two detector bins and sums to one. For broad bins, integrate the oscillated flux and cross section over energy.

ANNOTATED LINES

A Units of the oscillation phase 11–13

```
phase(E, d, L) = 1.266932679039099 * d * L / E
appearance(E, t, d, L) = (
    sin(2 * t)^2 * sin(phase(E, d, L))^2)
```

Supply energy `E` in GeV, splitting `d` in eV² and baseline `L` in km. The coefficient in `phase` converts these units. Supply the mixing angle in radians.

B Appearance counts by energy 15–19

```
theta ~ Uniform(interval(0.1, 0.2))
dm2 ~ Uniform(interval(0.002, 0.003))
probability = appearance.(energy, theta,
    dm2, baseline)
y ~ Binomial.(trials, probability)
```

Broadcast `appearance` over energy, holding the angle, splitting and baseline fixed. Each trial total has its own appearance probability. The product of Binomial laws conditions on the four totals.

C Condition on appearance counts 22–25

```
prior = lawof(record(theta = theta, dm2 = dm2))
K = kernelof(record(y = y), theta, dm2)
L = likelihoodof(K, record(y = appearance_data))
posterior = bayesupdate(L, prior)
```

Declare `theta` and `dm2` as kernel inputs with the same names as their prior fields. The kernel retains the fixed energies, baseline and trial totals. Bind the measured counts to its output field `y`.

D Appearance probabilities at a parameter point 27–30

```
point = record(theta = 0.15, dm2 = 0.0025)
log_L = logdensityof(L, point)
f_prob = functionof(probability, theta, dm2)
prob_at_point = f_prob(point)
```

Pass `theta` and `dm2` by name in a record to obtain the probability vector from `f_prob`. Use `logdensityof(L, point)` for the log likelihood of the four counts at those values.

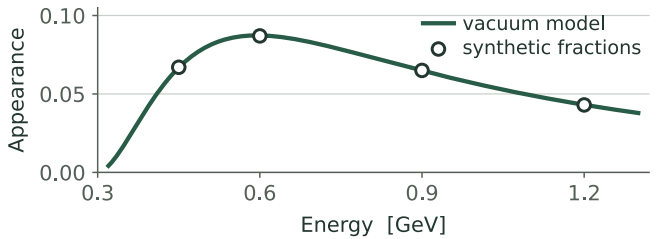
Predict survival

```
survival = 1 .- probability
survived ~ Binomial.(trials, survival)
```

Subtract each appearance count from its trial total to obtain the survival count. Use one of the two complementary counts per bin.

PROPAGATION AND RESPONSE

Appearance versus energy



The curve uses $L = 295$ km, $\theta = 0.15$ rad and $\Delta m^2 = 0.0025$ eV². Each point is an observed fraction from 1000 synthetic trials.

Three-flavour transition matrix

```
E = 0.6
p = nu.parameters(0.59, 0.15, 0.78,
    1.2, 0.000075, 0.0025)
P = nu.vacuum_probabilities(E, baseline,
    p, false)
```

Bind `nu` with `load_module` using the neutrino module URL. The attached `three-flavour.flatppl` includes the import. Rows give final flavour and columns initial flavour in electron, muon, tau order. `P[1,2]` gives muon-to-electron appearance.

Propagate through matter

```
P_m = nu.matter_probabilities(E, baseline,
    p, 1.4, false)
```

Electron density is 1.4 mol/cm³. Set the final argument to `true` for antineutrinos. For successive layers, compose amplitudes before taking the absolute square.

Set the mass ordering

The last two arguments of `nu.parameters` are `dm21` and `dm31` in eV². Use `dm21 > 0` and choose `dm31 > dm21` for normal ordering or `dm31 < 0` for inverted ordering.

Use expected event yields

```
unosc = [800, 900, 1000, 750]
expected = 0.9 .* unosc .* probability .+ 2
y_flux ~ Poisson.(expected)
```

`unosc` includes flux, cross section and exposure. Each bin has 90% efficiency and two expected background events. Use Poisson counts when the trial total is unknown.

Jessica Cox · bcox@mpp.mpg.de