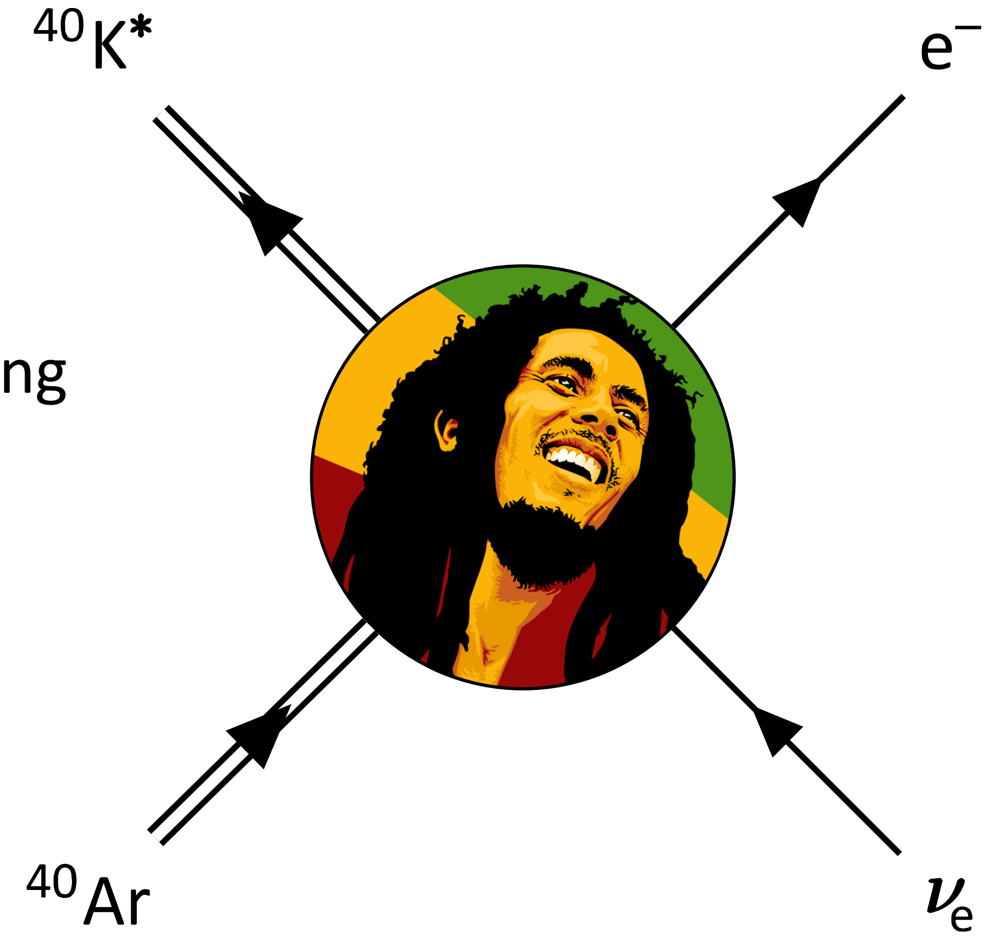
MARLEY updates for LArSoft v06_48_00

Steven Gardiner

LArSoft Coordination Meeting

12 September 2017



Model of **A**rgon **R**eaction **L**ow **E**nergy **Y**ields

New version of marley product (v1_0_0)

- Event generator for low-energy neutrino scattering on argon
- Included as v0_9_5 in LArSoft since v06_04_00
- Upgraded to new version for LArSoft v06_48_00
  - Lots of refactoring of MARLEY internals
  - Breaking changes to command-line executable (mostly used by CAPTAIN folks)
  - Physics output unchanged (saved for a later release)
  - Users of MARLEY through LArSoft probably won't notice much difference
- Near-term physics improvements (not merged yet, but stay tuned)
  - Refinement to nuclear de-excitation calculations (IWBC model)
  - Electron elastic scattering channel

Updates to MARLEY modules in LArSoft

1) Refactored MarleyGen larsim module

2) New MarleyTimeGen larsim module

3) New data product in lardataobj: `sim::SupernovaTruth`

Refactored MarleyGen larsim module

Overview

- An attempt to follow LArSoft best practices more closely
- Run-time FHiCL configuration validation (fhicl::Atom<T>, etc.)
- Most of the code moved into two new algorithm classes:
  - evgen::ActiveVolumeVertexSampler
  - evgen::MARLEYHelper
- FHiCL parameters forwarded to MARLEY's internal configuration format (marley::JSON)
  - Reduces code duplication in the module
  - Helps prevent problems like LArSoft issue #14847

evgen::ActiveVolumeVertexSampler

- Chooses vertex locations for supernova neutrino interactions
  - TPC chosen using its active mass as a sampling weight
  - Vertex position sampled uniformly within the chosen TPC's active volume

- Time sampling not yet implemented
  - For now, all interactions occur at readout window time t = 0

- Three FHiCL configuration parameters
  - **type**: a string with allowed values of
    - "sampled" (vertex chosen as described above)
    - "fixed" (every interaction will be generated at the same location)
  - **seed**: optional RNG seed for this algorithm
  - **position**: sequence giving the vertex position 3-vector (type == "fixed" only)

- Does something like this already exist in LArSoft elsewhere?
  - Proton decay, etc., might use a similar approach

evgen::MARLEYHelper

- Encapsulates the LArSoft interface to MARLEY
- Intended to be similar to evgb::GENIEHelper
- FHiCL configuration shown in **`$LARSIM_DIR/job/marley.fcl`**
- Main use case is through

```
simb::MCTruth create_MCTruth(const TLorentzVector& vtx_pos,
  marley::Event* marley_event = nullptr)
```

  - Takes a TLorentzVector specifying the vertex 4-position as input
  - Returns a simb::MCTruth object generated using the current MARLEY configuration
  - Optional second argument allows access to the corresponding native MARLEY representation of the event
- Include in your own module using a **`fhicl::Table<evgen::MARLEYHelper::Config>`** object
- Modules can also access the **`marley::Generator`** object directly if needed using the **`get_generator()`** method

New MarleyTimeGen larsim module

- Project from DUNE Supernova / Low Energy Hack Days II & III

- Allows MARLEY to sample events using time-dependent supernova fluxes
  - Triggering studies
  - Time-dependent SN physics, e.g., standing accretion shock instability (SASI)

- FHiCL configuration uses both new algorithms mentioned previously. Other parameters include
  - **sampling_mode**: "histogram" (unbiased), "uniform time", "uniform energy"
  - **nu_per_event**: Integer number of neutrino vertices to generate per art::Event
  - **spectrum_file**: Name of a file describing the time-dependent flux
  - **spectrum_file_format**: "th2d" (ROOT file) or "fit" (ASCII fit parameters)
  - **namecycle**: Name of the TH2D object to use (when using a ROOT file)
  - (and a few others that can usually be set to their default values)

MarleyTimeGen spectrum files

- **Warning**: all input files must specify an *incident* (i.e., not cross-section weighted) neutrino flux.
  - MARLEY applies its own cross section model to all input spectra
  - Deliberate design choice to avoid subtle physics inconsistencies

ROOT input format

- TH2D object describes the incident flux
- Time bins on the X axis (s)
- Energy bins on the Y axis (MeV)
- No convention established yet for bin count units. MARLEY internally renormalizes the input spectrum.

MarleyTimeGen spectrum files

ASCII fit parameter input format

$$\phi(E_\nu) = N_0 \frac{(\alpha+1)^{\alpha+1}}{\langle E_\nu\rangle \Gamma(\alpha+1)} \left(\frac{E_\nu}{\langle E_\nu\rangle}\right)^{\alpha} \exp\left[-(\alpha+1)\frac{E_\nu}{\langle E_\nu\rangle}\right]$$

- Text file that gives fit parameters for a standard pinched thermal spectrum

- Each line of the file has the following format:

  time bin low edge (s), E_{mean} for ν_e (MeV), α for ν_e (dimensionless), luminosity for ν_e (erg / s), E_{mean} for anti-ν_e (MeV), α for anti-ν_e (dimensionless), luminosity for anti-ν_e (erg / s), E_{mean} for ν_x (MeV), α for ν_x (dimensionless), luminosity for ν_x (erg / s)

- Whitespace-delimited
- The entries for ν_x should be for a single flavor
- Final line should contain a single entry: the high edge for the last time bin (if it is missing, the bin size for the penultimate bin in assumed for the last bin as well)

New data product in lardataobj: `sim::SupernovaTruth`

- The MarleyTimeGen module produces both simb::MCTruth objects and a new object called sim::SupernovaTruth
- Also produces one-to-one associations between them

sim::SupernovaTruth

- Simple struct with 4 members:
  - double **SupernovaTime**:
    - neutrino emission time (s) since supernova core bounce (can be negative)
  - SupernovaSamplingMode_t **SamplingMode**:
    - Enum describing how the events were sampled
    - kUnbiased (sample directly from cross-section weighted distribution)
    - kUniformEnergy (sample neutrino energies uniformly and reweight)
    - kUniformTime (sample times uniformly and reweight)

New data product in lardataobj: `sim::SupernovaTruth`

sim::SupernovaTruth

- Simple struct with 4 members: (continued)
  - double **Weight**:
    - Statistical weight for the neutrino vertex
    - Unity for unbiased sampling, otherwise equal to the likelihood ratio
  - double **FluxAveragedCrossSection**:
    - Average neutrino interaction cross section for the input flux used by MARLEY to generate the neutrino vertex
    - Used to help compute expected event rates

expected number of events

detector active mass

energy- and time-integrated neutrino flux

$$N_{f,x} = \frac{M}{m_{\mathrm{atom}}} \widetilde{\sigma}_{f,x} \, \Phi_f$$

argon atomic mass

flux-averaged cross section