PROPOSAL: A library to propagate leptons and high energy photons

Jean-Marco Alameddine, Maximilian Sackel, Jan Soedingrekso, Alexander Sandrock

October 9, 2020

Technische Universität Dortmund
What is PROPOSAL?

- **PROPOSAL**: Software library to propagate high-energy leptons and photons
- Written in C++11, callable from Python as well
  - Try: `pip install proposal`
- Easy-to-use, but still very customizable for different applications
- Actively maintained
  - Visit our GitHub: [https://github.com/tudo-astroparticlephysics/PROPOSAL](https://github.com/tudo-astroparticlephysics/PROPOSAL)
PROPOSAL originally specialized on $\mu$ and $\tau$ propagation.

Recently, $\gamma$ propagation and an improved treatment of $e^-/e^+$ has been added.

Selection of different parametrizations for each process:
- Several up-to-date parametrizations available
- Including effects such as LPM
- Rare processes can be included
- Easy to implement new parametrizations
Next to energy losses, PROPOSAL can simulate...

- Multiple scattering effects
- Particle decays
- Creation of secondary particles
Interaction are characterized by their relative energy loss $\nu$

PROPOSAL differentiates continuous energy losses and stochastic energy losses:

$\nu < \nu_{\text{cut}}$  
continuous

$\nu > \nu_{\text{cut}}$  
stochastic

with

$\nu_{\text{cut}} = \min \left[ e_{\text{cut}}/E, \nu'_{\text{cut}} \right]$

→ Vary values for $e_{\text{cut}}$ and $\nu'_{\text{cut}}$ to adjust precision
(Simplified) PROPOSAL propagation algorithm

1. Initial energy $E_i$
2. Sample energy $E_f$ where next stochastic loss occurs
3. Calculate distance until next stochastic loss
4. Sampling stochastic loss
5. Repeat until fatal interaction, decay, $E < e_{\text{low}}$ or $d = d_{\text{max}}$
(Simplified) PROPOSAL propagation algorithm

Initial energy $E_i$

Sample energy $E_f$ where next stochastic loss occurs

Calculate distance until next stochastic loss

Sampling stochastic loss

Repeat until fatal interaction, decay, $E < e_{\text{low}}$ or $d = d_{\text{max}}$

Energy integral

$$\int_{E_i}^{E_f} \frac{\sigma(E)}{-f(E)} \cdot dE = -\log(\xi)$$

- $\sigma(E) = \sigma_{\text{total, stochastic}}$
- $f(E) = \left. \frac{dE}{dx} \right|_{\text{cont}} \propto E \int_{v_{\text{min}}}^{v_{\text{cut}}} v \frac{d\sigma}{dv} \, dv$
- $\xi \in [0, 1)$

Stochastic losses are all energy losses with a fractional energy loss $v > v_{\text{cut}}$. !
(Simplified) PROPOSAL propagation algorithm

- **Initial energy** $E_i$
- Sample energy $E_f$ where next stochastic loss occurs
- Calculate distance until next stochastic loss
- Sampling stochastic loss
- Repeat until fatal interaction, decay, $E < e_{\text{low}}$ or $d = d_{\text{max}}$

**Displacement integral**

$$x_f = x_i - \int_{E_i}^{E_f} \frac{dE}{f(E)}$$

$$f(E) = \left. \frac{dE}{dx} \right|_{\text{cont.}} \propto E \int_{v_{\text{min}}}^{v_{\text{cut}}} v \frac{d\sigma}{dv} dv$$
(Simplified) PROPOSAL propagation algorithm

- Initial energy $E_i$
- Sample energy $E_f$ where next stochastic loss occurs
- Calculate distance until next stochastic loss
- Sampling stochastic loss
- Repeat until fatal interaction, decay, $E < e_{\text{low}}$ or $d = d_{\text{max}}$

Stochastic loss

$$\frac{1}{\sigma_{\text{total}}} \int_{v_{\text{cut}}}^{v} \frac{d\sigma}{dv} = \xi$$

- $\xi \in [0, 1)$
(Simplified) PROPOSAL propagation algorithm

1. Initial energy $E_i$
2. Sample energy $E_f$ where next stochastic loss occurs
3. Calculate distance until next stochastic loss
4. Sampling stochastic loss
5. Repeat until fatal interaction, decay, $E < e_{\text{low}}$ or $d = d_{\text{max}}$
Many integrals need to be calculated during propagation:

- Usage of interpolation tables to decrease runtime
- Both cross section integrals (left) and integrals necessary for propagation steps (right) are interpolated

\[ \int \frac{d\sigma}{d\nu} \ d\nu \]

\[ \int \frac{d\sigma}{d\nu} \ d\nu \]

\[ \int_{E_i}^{E_f} \frac{\sigma(E)}{-f(E)} \cdot dE = -\log(\xi) \]

\[ x_f = x_i - \int_{E_i}^{E_f} \frac{dE}{f(E)} \]
**Minimal PROPOSAL code example**

```cpp
// read properties from config file
prop Propagator(MuMinusDef(), "config.json");

// define initial state
Vector3D position(0, 0, 0);
Vector3D direction(0, 0, 1);
auto energy = 1e8.f; // MeV
init_state DynamicData(position, direction, energy);

vector<double> energies;

for(int i = 0; i < 1e5; i++) {
    auto track = prop.Propagate(init_state, 1e5); // cm
    double E_final = track.back().GetEnergy();
    energies.push_back(E_final);
}
```

**json file**

```json
"global":
{
    "cuts":
    {
        "e_cut": INF,
        "v_cut": 0.05,
        "cont_rand": false
    }
},

"sectors": [
    {
        "medium": "ice",
        "geometries": [
            {
                "shape": "sphere",
                "origin": [0, 0, 0],
                "outer_radius": 6374134000000
            }
        ]
    }
]
```
Final energies of $10^5$ muons with $E_i = 10^8$ MeV propagated through 1 km of ice

- Simulating muons with identical initial energies causes a peak in the energy distribution.
- All particles with zero stochastic losses will have the same final energy.

Continuous randomization

PROPOSAL provides the feature of continuous randomization, which adds random fluctuations to the continuous losses.
Simulating muons with identical initial energies causes a peak in the energy distribution.

All particles with zero stochastic losses will have the same final energy.

PROPOSAL provides the feature *continuous randomization*.

This adds random fluctuations to the continuous losses.
IceCube Neutrino Observatory

- PROPOSAL used in IceCube simulation chain
  - Interested in energy losses along a particle track, provided by the PROPOSAL propagator
  - Energy losses are further processed by other tools to simulate Cherenkov photons

- Adjustable precision important for all large-scale detectors
  - High precision inside detector (small $v_{\text{cut}}$)
  - High performance in front of detector (higher $v_{\text{cut}}$ with continuous randomization)

Credit: IceCube Collaboration
- Up to CORSIKA7: Electromagnetic shower component simulated by EGS4
- CORSIKA 8: Inclusion of PROPOSAL as an EM shower model (see CORSIKA GitLab)
- CORSIKA is interested in single propagation steps for $e^+$, $e^-$ and $\gamma$
  - Modular structure of PROPOSAL allows to extract individual components of the propagation routine
NuRadioMC

CORSIKA 8

BAIKAL-GVD
Future developments

- Neutrino propagation in PROPOSAL
  → Can be used for tau regeneration studies

- Stochastic deflections
  → Deflections may occur in (very) stochastic interactions (especially for bremsstrahlung and photonuclear interactions)
  → Can be used to examine the influences, e.g. on direction reconstructions

- Backward Monte Carlo simulations [1705.05636]
  → Can be used to increase statistics for relevant event signatures
Current developments

- Current developments on GitHub branch `restructure_parametrization`
  - Several improvements, both internally as well as for users
  - Preparations for inclusion in CORSIKA 8
  - Will be merged soon with our master branch
- If you are interested in using PROPOSAL ...
  ...use `pip install proposal` to try it out
  ...look at our GitHub page for more information
  ...contact us directly! jean-marco.alameddine@udo.edu