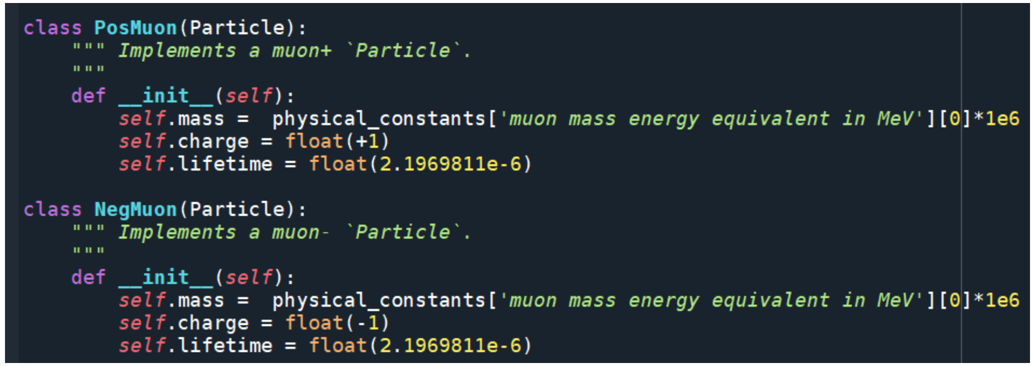
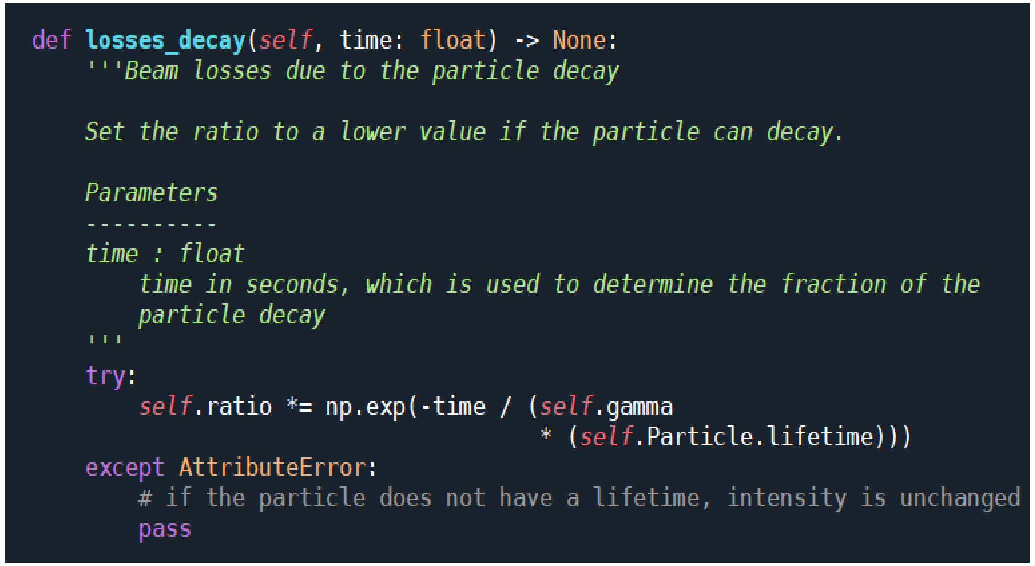
Muon Particle Merge Request

- Added `MuPlus` and `MuMinus` particles to `beam`.
- Both have a new parameter, `self.lifetime`.
- This variable is then used in the function `beam.decay_losses(time: float)` to reduce `beam.ratio` according to the decay law
- A change in intensity has no effect, as only `ratio*n_macroparticles` is used in most intensity effect functions.
- The intensity of the beam is not adjusted if particles are lost or decay.
  - ➢ Different MR
- The decay function has to be individually called by the user (as all other losses too).



```python
class PosMuon(Particle):
    """ Implements a muon+ `Particle`.
    """
    def __init__(self):
        self.mass =  physical_constants['muon mass energy equivalent in MeV'][0]*1e6
        self.charge = float(+1)
        self.lifetime = float(2.1969811e-6)

class NegMuon(Particle):
    """ Implements a muon- `Particle`.
    """
    def __init__(self):
        self.mass =  physical_constants['muon mass energy equivalent in MeV'][0]*1e6
        self.charge = float(-1)
        self.lifetime = float(2.1969811e-6)
```



```python
def losses_decay(self, time: float) -> None:
    '''Beam losses due to the particle decay

    Set the ratio to a lower value if the particle can decay.

    Parameters
    ----------
    time : float
        time in seconds, which is used to determine the fraction of the
        particle decay
    '''
    try:
        self.ratio *= np.exp(-time / (self.gamma
                                      * (self.Particle.lifetime)))
    except AttributeError:
        # if the particle does not have a lifetime, intensity is unchanged
        pass
```