GRPropa: 3D propagation of electromagnetic cascades

Rafael Alves Batista

rafael.alvesbatista@physics.ox.ac.uk

7/Nov/2016

GRPropa

- Based on the modular code structure of the CRPropa 3 code for cosmic-ray propagation

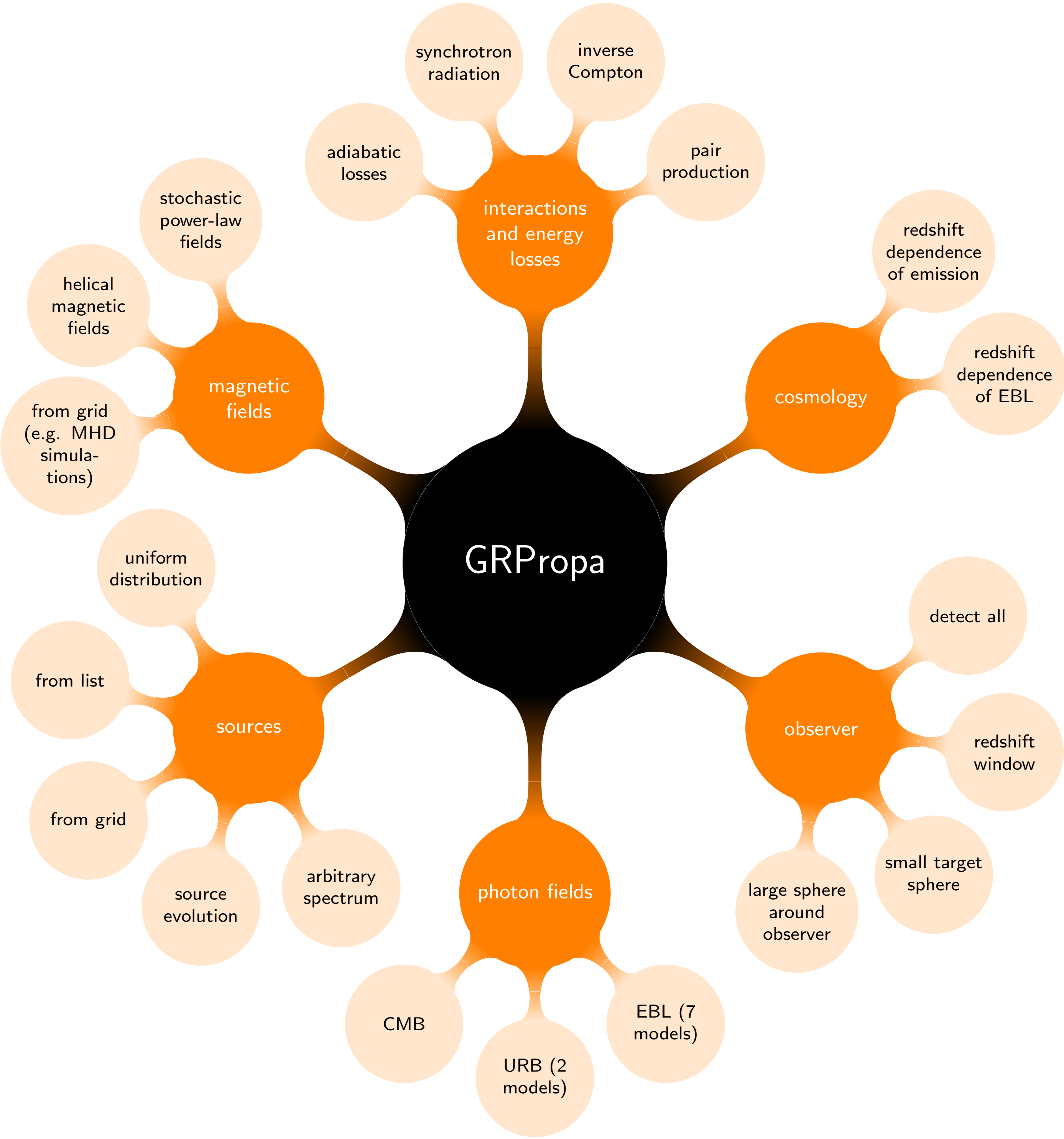
  CRPropa: github.com/CRPropa/CRPropa3 *[RAB et al. JCAP 05 (2016) 038. arXiv:1603.07142]*

- four-dimensional (3D + time) simulation of gamma-ray propagation
- modular C++ code with Python bindings
- other codes: Elmag (1D + time; small-angle approximation)

  [M. Kachelriess et al. Comp. Phys. Comm. 183 (2011) 1036]

- energy range: 1 GeV - 1 PeV (will be extended)
- particles weighted according to differential cross section → "thinning" to optimise performance
- arbitrary magnetic field configurations and a few default options
- code already used in arXiv:1607.00320

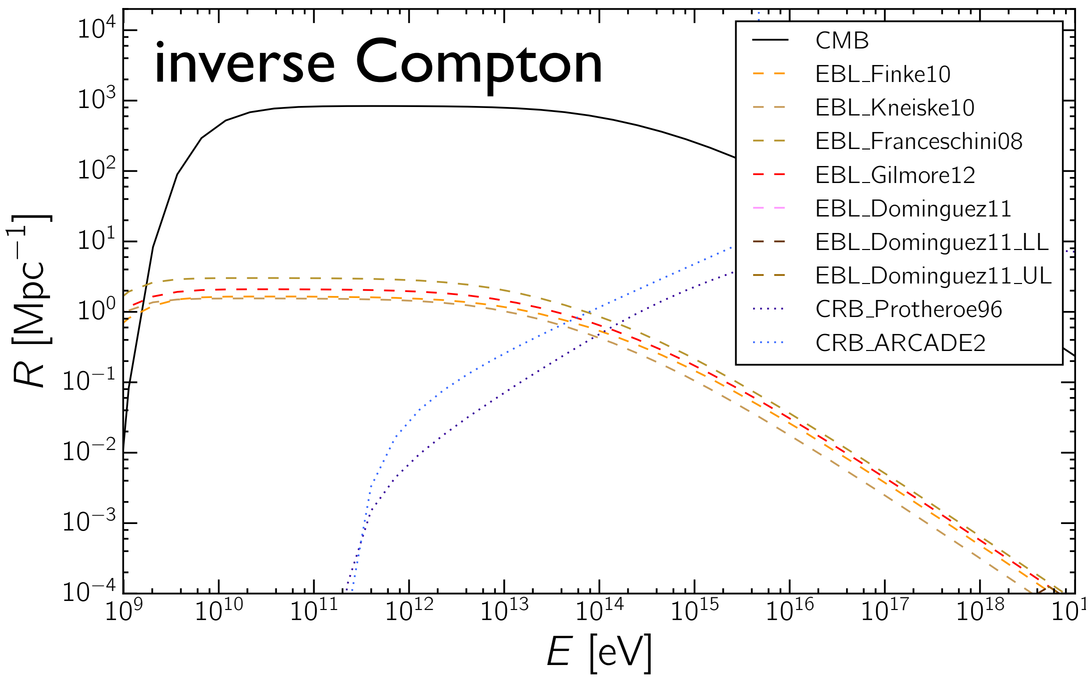
  [RAB et al. Phys. Rev. D 94 (2016) 083005]

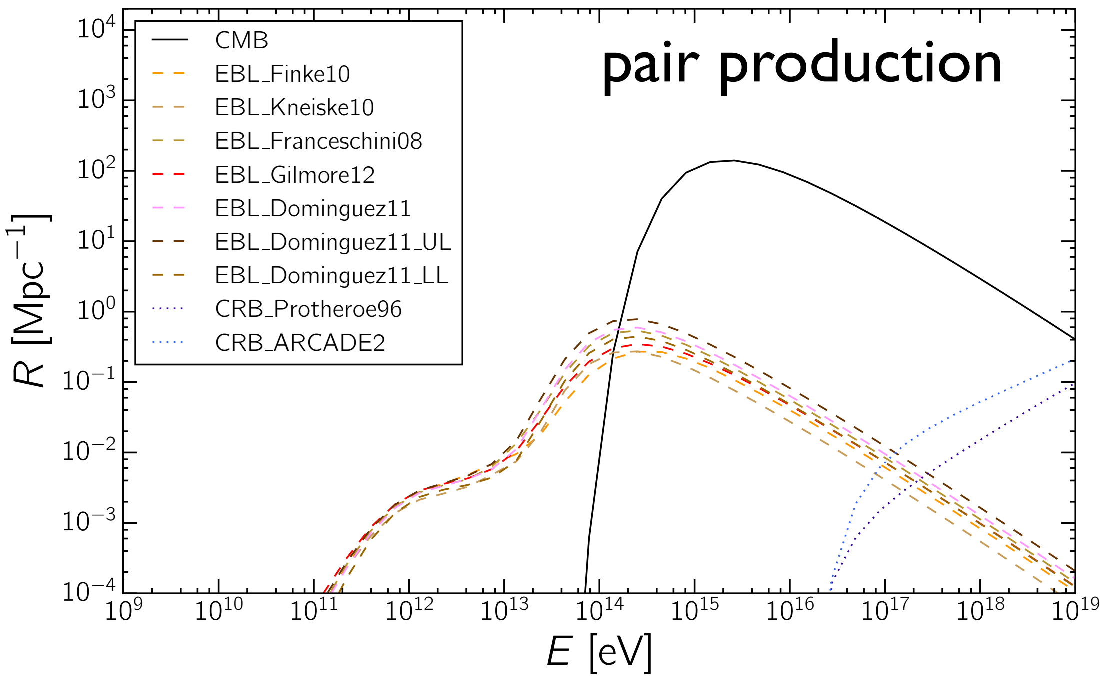


GRPropa

- interactions and energy losses
  - adiabatic losses
  - synchrotron radiation
  - inverse Compton
  - pair production
- cosmology
  - redshift dependence of emission
  - redshift dependence of EBL
- observer
  - detect all
  - redshift window
  - small target sphere
  - large sphere around observer
- photon fields
  - EBL (7 models)
  - URB (2 models)
  - CMB
- sources
  - arbitrary spectrum
  - source evolution
  - from grid
  - from list
  - uniform distribution
- magnetic fields
  - from grid (e.g. MHD simulations)
  - helical magnetic fields
  - stochastic power-law fields

interactions and photon fields





synchrotron

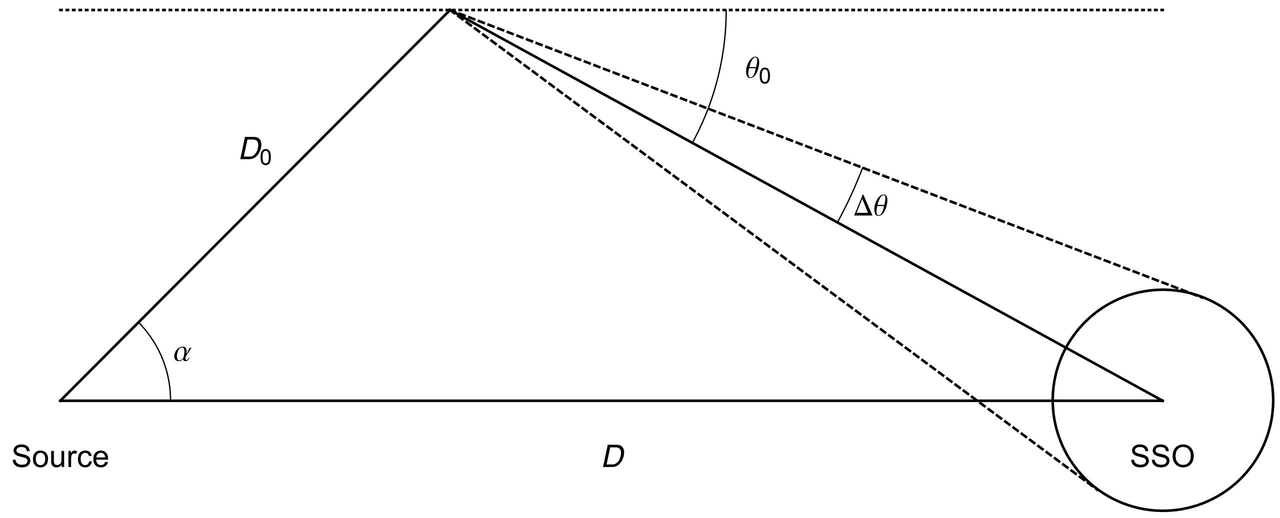
$$\frac{dE}{dx} \approx \frac{m_e^2 c^4 \chi^2}{\hbar c \left(1 + 4.8(1+\chi)\ln(1+1.7\chi) + 3.44\chi^2\right)^{\frac{2}{3}}} \qquad \chi = \frac{\left|\vec{p} \times \vec{B}\right|}{m_e c \, 4.4 \times 10^{13}\ \mathrm{G}}$$

redshift losses

$$\frac{dt}{dz} = \frac{1}{H_0(1+z)} \frac{1}{\sqrt{\Omega_m (1+z)^3 + \Omega_\Lambda}}$$

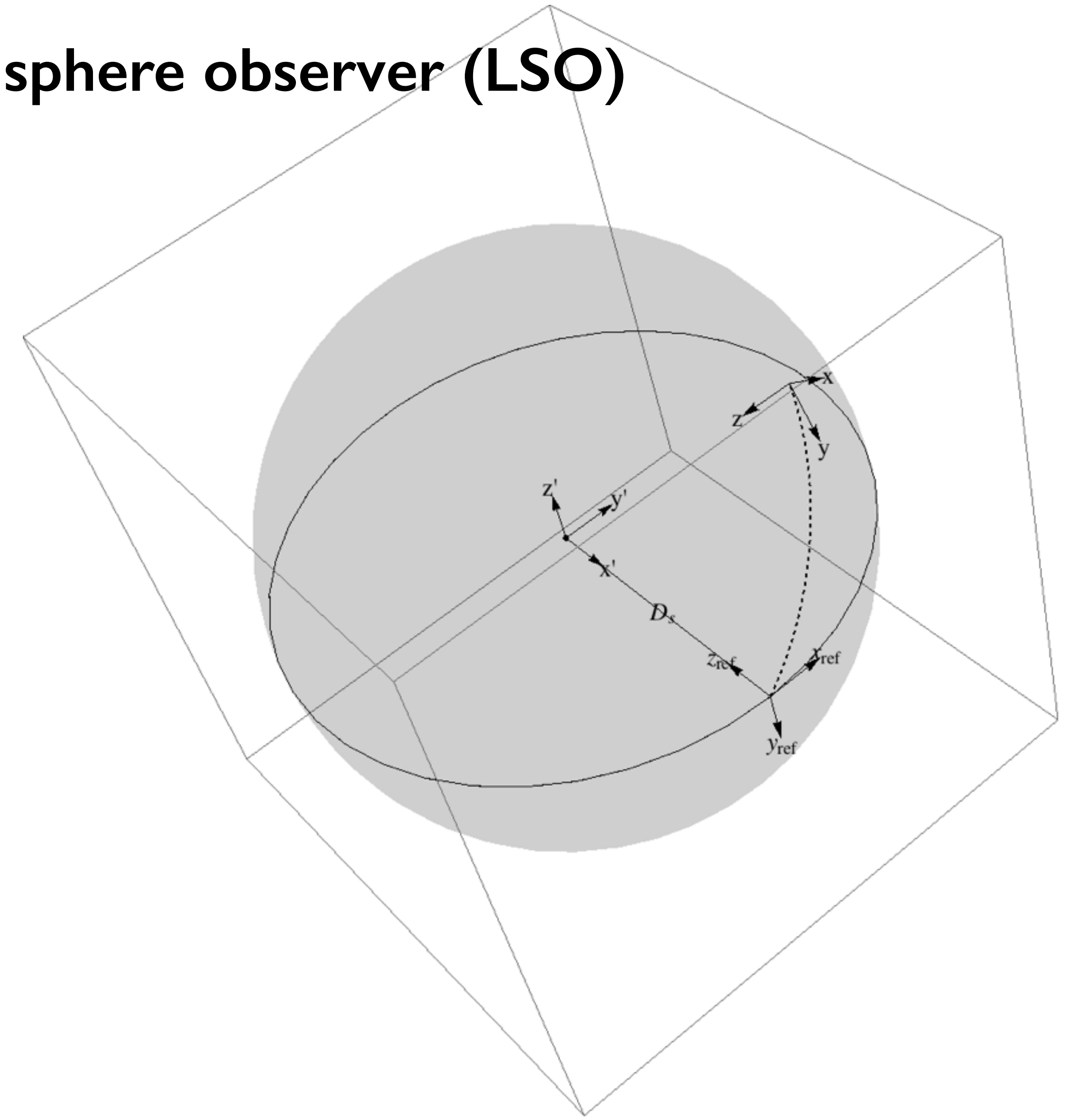
detection methods and observer geometry

small sphere observer (SSO)



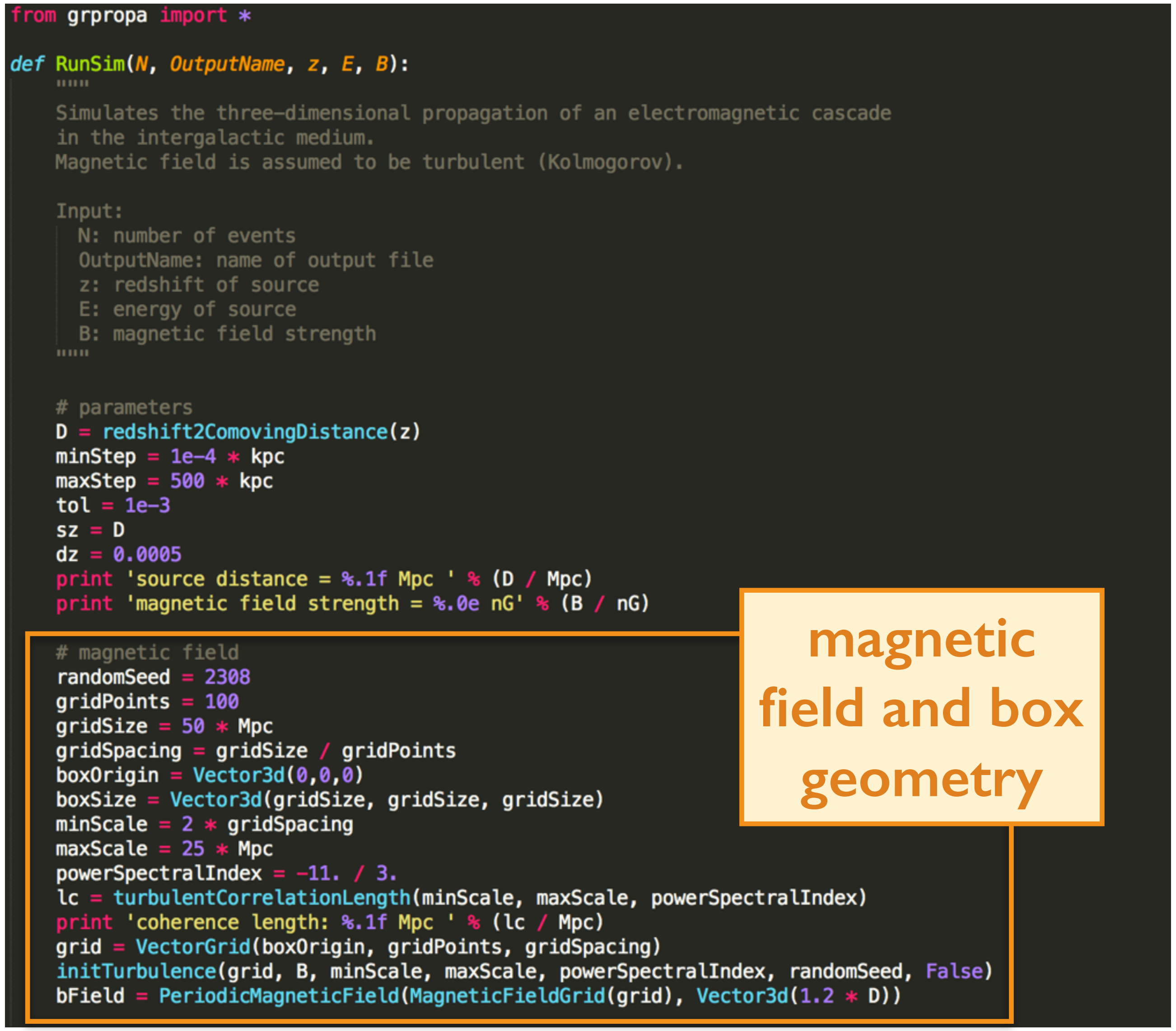
- SSO is the intuitive method, but it is hard to collect events with this setup
- for single sources LSO is the preferred method as it is computationally efficient
- the LSO method is <u>exact</u> as it is a topological transformation

large sphere observer (LSO)



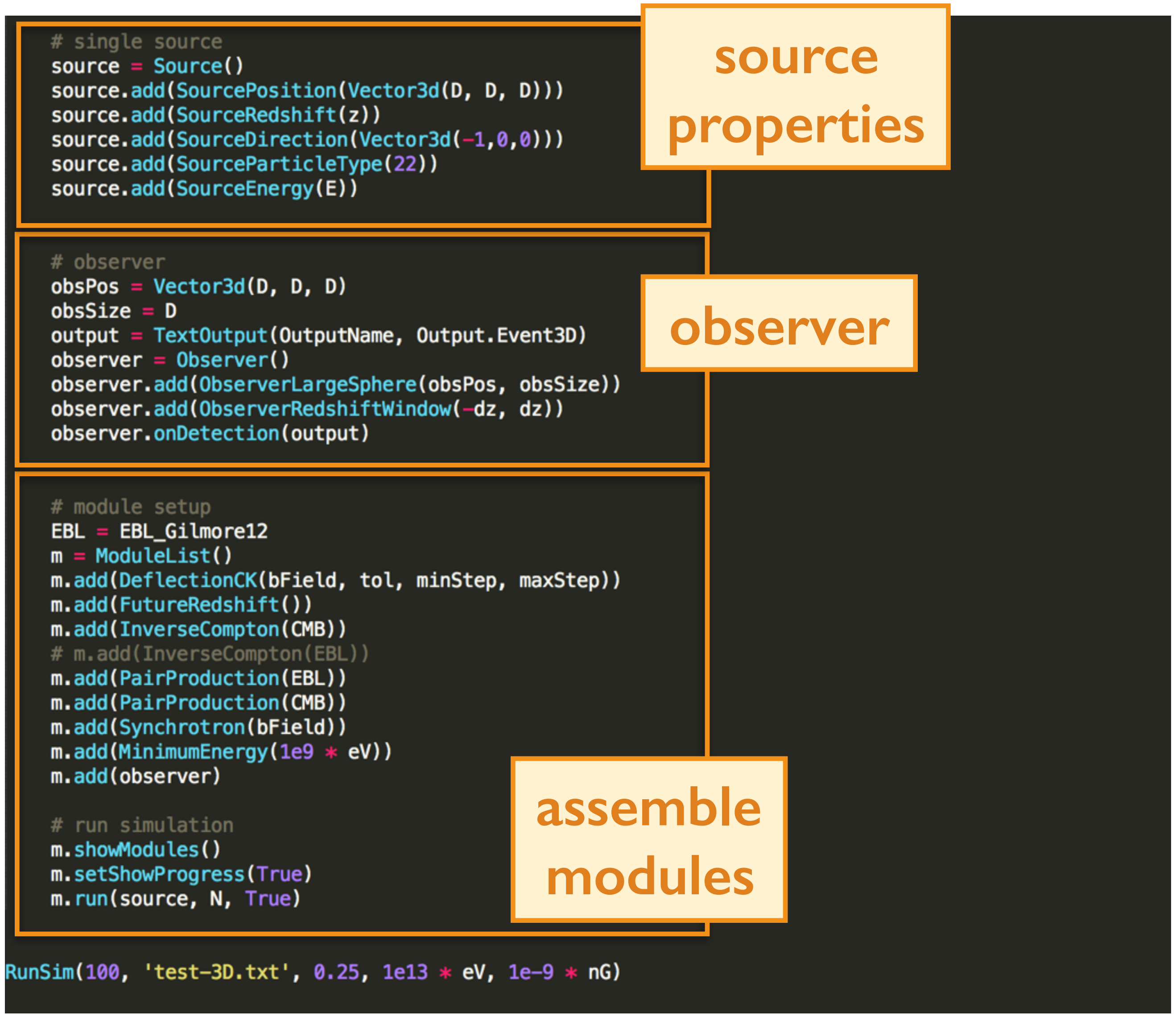
The illustration of the concept of the Large Sphere Observer. The global coordinate system S' is represented by coordinates $\{x', y', z'\}$ which is located at the centre of the sphere. The reference coordinate system S_{ref} is represented by coordinates $\{x_{ref}, y_{ref}, z_{ref}\}$ with origin at $(x', y', z') = (x'_0, y'_0, z'_0)$. The observer coordinate system S with its origin being placed at the hit position $(x', y', z') = (x'_l, y'_l, z'_l)$ is represented by coordinates $\{x, y, z\}$. The solid line represents the equator of the Large Sphere, while the dashed line is the geodesic along which the parallel transport takes place.

example of steering file



```python
from grpropa import *

def RunSim(N, OutputName, z, E, B):
    """
    Simulates the three-dimensional propagation of an electromagnetic cascade
    in the intergalactic medium.
    Magnetic field is assumed to be turbulent (Kolmogorov).

    Input:
      N: number of events
      OutputName: name of output file
      z: redshift of source
      E: energy of source
      B: magnetic field strength
    """

    # parameters
    D = redshift2ComovingDistance(z)
    minStep = 1e-4 * kpc
    maxStep = 500 * kpc
    tol = 1e-3
    sz = D
    dz = 0.0005
    print 'source distance = %.1f Mpc ' % (D / Mpc)
    print 'magnetic field strength = %.0e nG' % (B / nG)

    # magnetic field
    randomSeed = 2308
    gridPoints = 100
    gridSize = 50 * Mpc
    gridSpacing = gridSize / gridPoints
    boxOrigin = Vector3d(0,0,0)
    boxSize = Vector3d(gridSize, gridSize, gridSize)
    minScale = 2 * gridSpacing
    maxScale = 25 * Mpc
    powerSpectralIndex = -11. / 3.
    lc = turbulentCorrelationLength(minScale, maxScale, powerSpectralIndex)
    print 'coherence length: %.1f Mpc ' % (lc / Mpc)
    grid = VectorGrid(boxOrigin, gridPoints, gridSpacing)
    initTurbulence(grid, B, minScale, maxScale, powerSpectralIndex, randomSeed, False)
    bField = PeriodicMagneticField(MagneticFieldGrid(grid), Vector3d(1.2 * D))
```

magnetic field and box geometry



```python
    # single source
    source = Source()
    source.add(SourcePosition(Vector3d(D, D, D)))
    source.add(SourceRedshift(z))
    source.add(SourceDirection(Vector3d(-1,0,0)))
    source.add(SourceParticleType(22))
    source.add(SourceEnergy(E))

    # observer
    obsPos = Vector3d(D, D, D)
    obsSize = D
    output = TextOutput(OutputName, Output.Event3D)
    observer = Observer()
    observer.add(ObserverLargeSphere(obsPos, obsSize))
    observer.add(ObserverRedshiftWindow(-dz, dz))
    observer.onDetection(output)

    # module setup
    EBL = EBL_Gilmore12
    m = ModuleList()
    m.add(DeflectionCK(bField, tol, minStep, maxStep))
    m.add(FutureRedshift())
    m.add(InverseCompton(CMB))
    # m.add(InverseCompton(EBL))
    m.add(PairProduction(EBL))
    m.add(PairProduction(CMB))
    m.add(Synchrotron(bField))
    m.add(MinimumEnergy(1e9 * eV))
    m.add(observer)

    # run simulation
    m.showModules()
    m.setShowProgress(True)
    m.run(source, N, True)

RunSim(100, 'test-3D.txt', 0.25, 1e13 * eV, 1e-9 * nG)
```

source properties

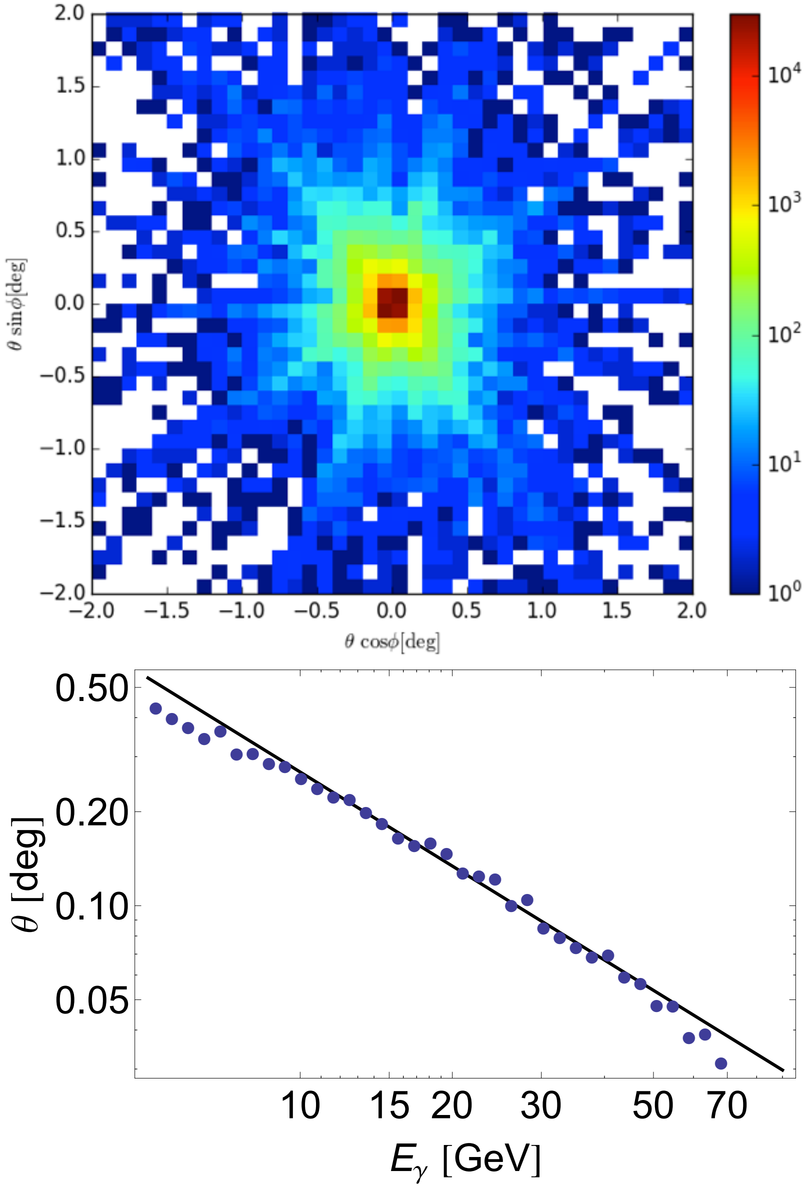
observer

assemble modules

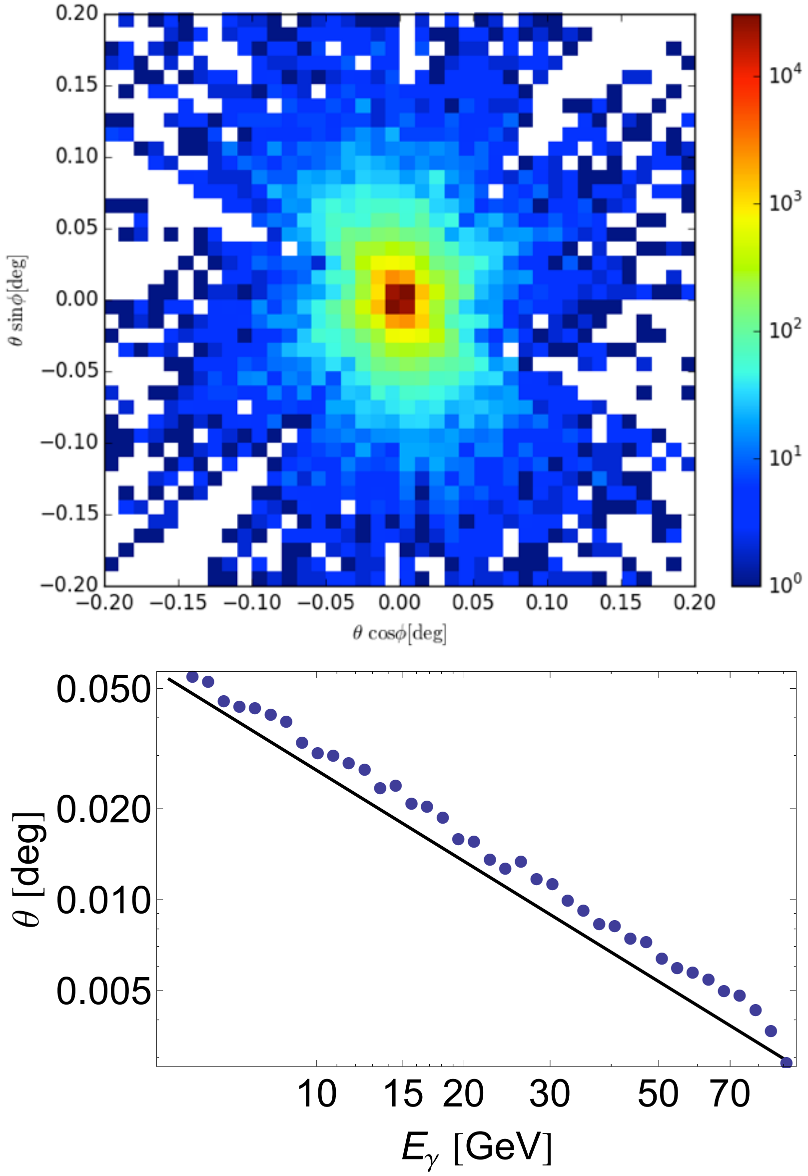
3D simulations

RAB, A. Saveliev, G. Sigl, T. Vachaspati. PRD 94 (2016) 083005. arXiv:1607.00320

B=10^{-15} G



B=10^{-16} G



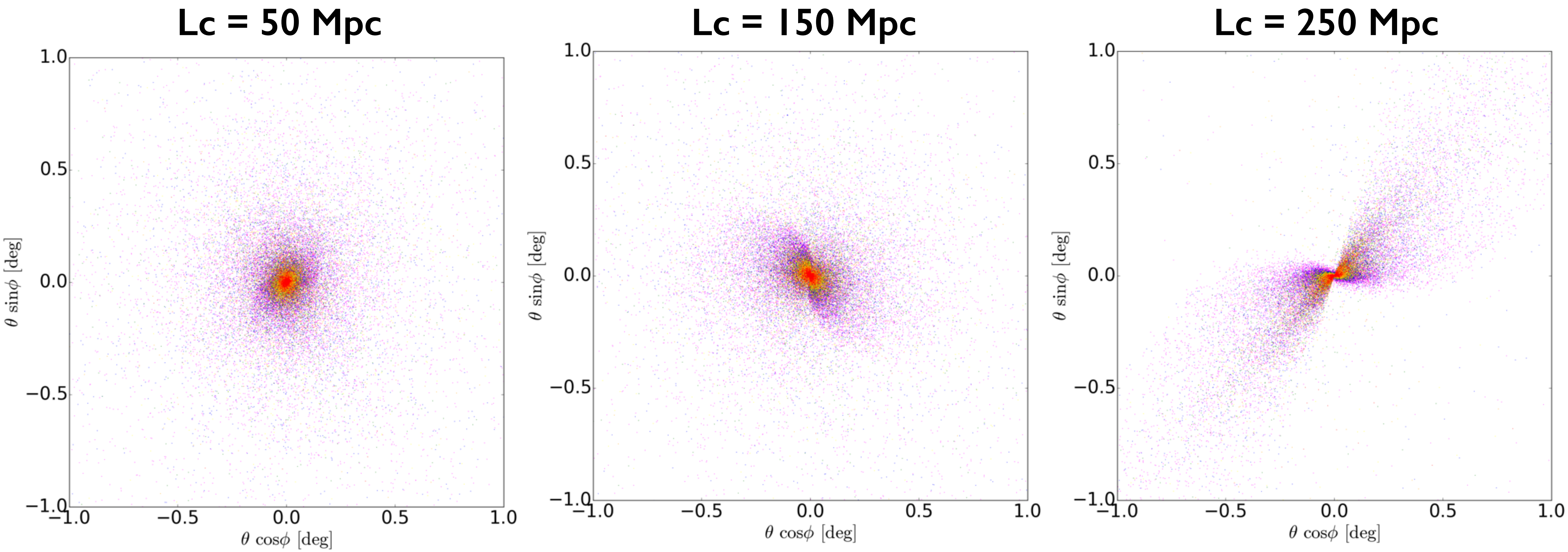
- stochastic magnetic field with Batcherlor spectrum
- blazar located at D=1 Gpc
- performance: 10^5 initial photons , without thinning, take about 8 hours on 64 cores at 2.3 GHz

theoretical prediction

$$\theta(E_\gamma) \simeq 0.05^\circ \kappa (1+z_\mathrm{s})^{-4} \left(\frac{B}{\mathrm{fG}}\right)\left(\frac{E_\gamma}{0.1\,\mathrm{TeV}}\right)^{-1}\left(\frac{D_\mathrm{s}}{\mathrm{Gpc}}\right)^{-1}\left(\frac{E_\mathrm{TeV}}{10\,\mathrm{TeV}}\right)^{-1}$$

Neronov & Semikoz. PRD 80 (2009) 123012.

morphology of blazar pair haloes



Lc = 50 Mpc

Lc = 150 Mpc

Lc = 250 Mpc

effects of the coherence length for helicity = +1

RAB, A. Saveliev, G. Sigl, T. Vachaspati. PRD 94 (2016) 083005. arXiv:1607.00320

to-do list and others

- improve agreement with Elmag and understand potential differences
- immediate problems that need fixing: inelasticity of ICS requires energy threshold to be very low (it is taking too long)
- particle-by-particle MC propagation → computationally inefficient 😞
- particles weighted according to cross section → thinning (still being tested, seems to be working) 😀
- implement relevant interactions above 10 PeV (double and triplet pair production)
- magnetic fields are tested and working
- implement photon-ALP conversion
- the code will be open for contributions and enhancements (e.g. LIV)
- write output modules to interface with ctools, gammapy, etc (probably easily done)
- estimated time for release: ~january/2017