coupling axion-like particles to photons during propagation: the ALPinist plug-in

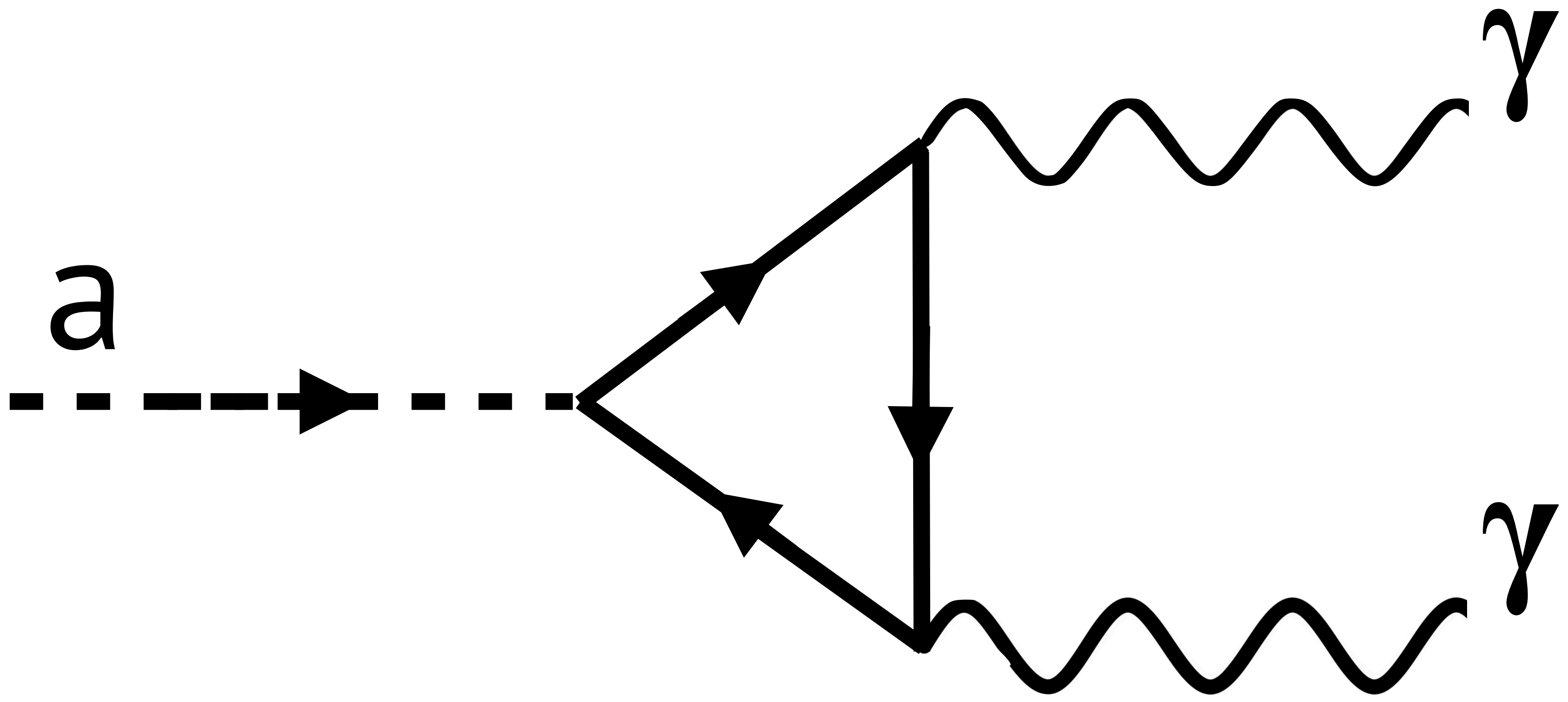
Rafael Alves Batista, Cristina Viviente, Gaetano Di Marco, Miguel A. Sánchez-Conde

Instituto de Física Teórica UAM-CSIC, Madrid, Spain
Departamento de Física Teórica, Universidad Autónoma de Madrid, Madrid, Spain

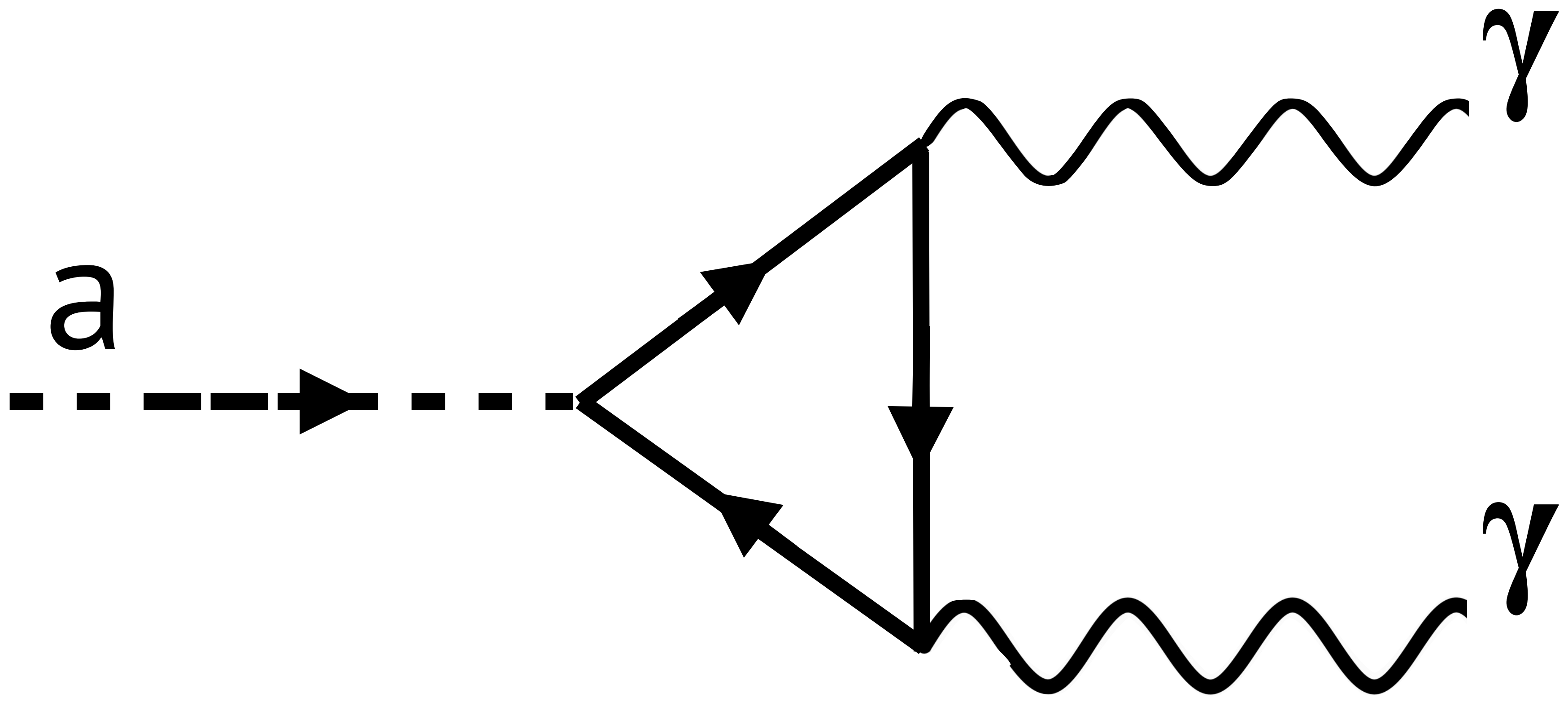
rafael.alvesbatista@uam.es

axion-like particles

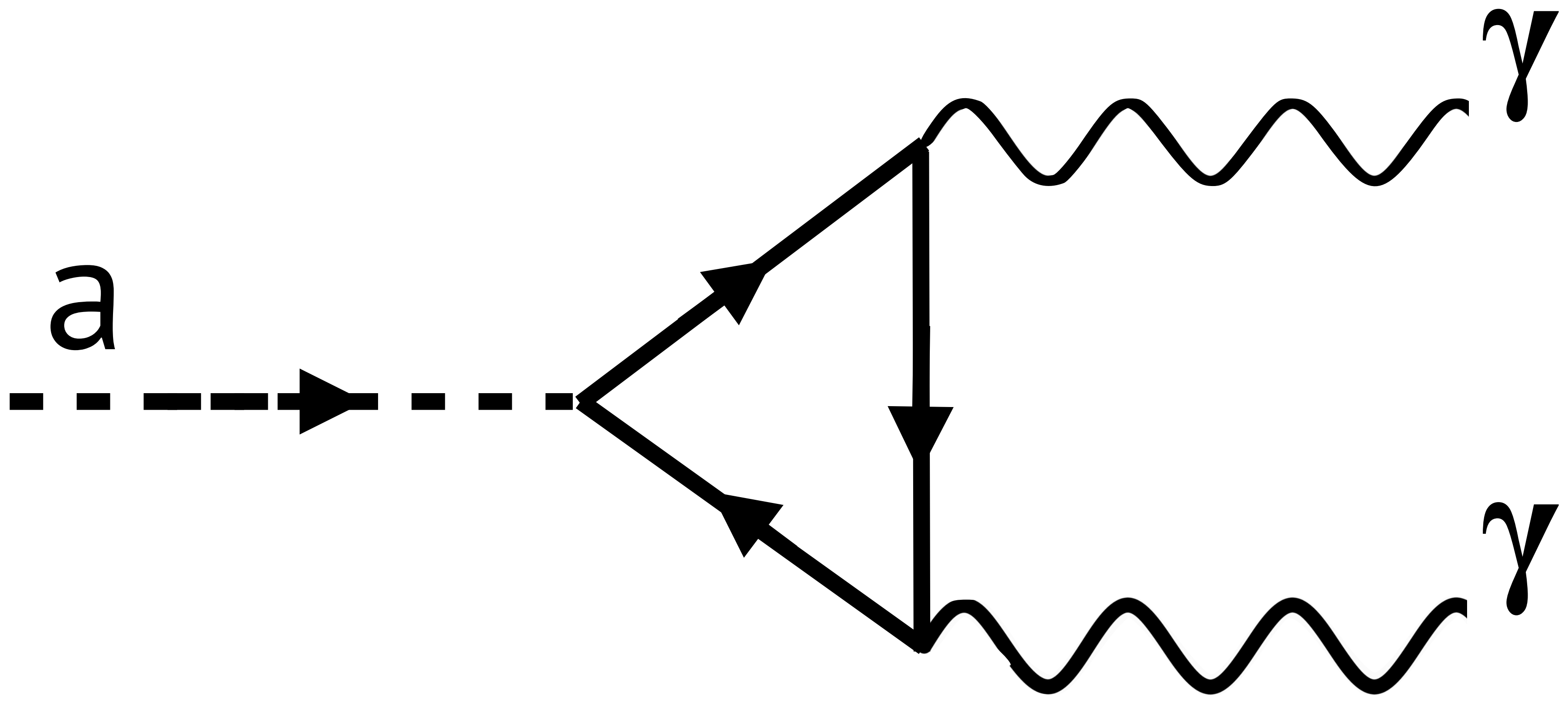
axion-like particles



axion-like particles



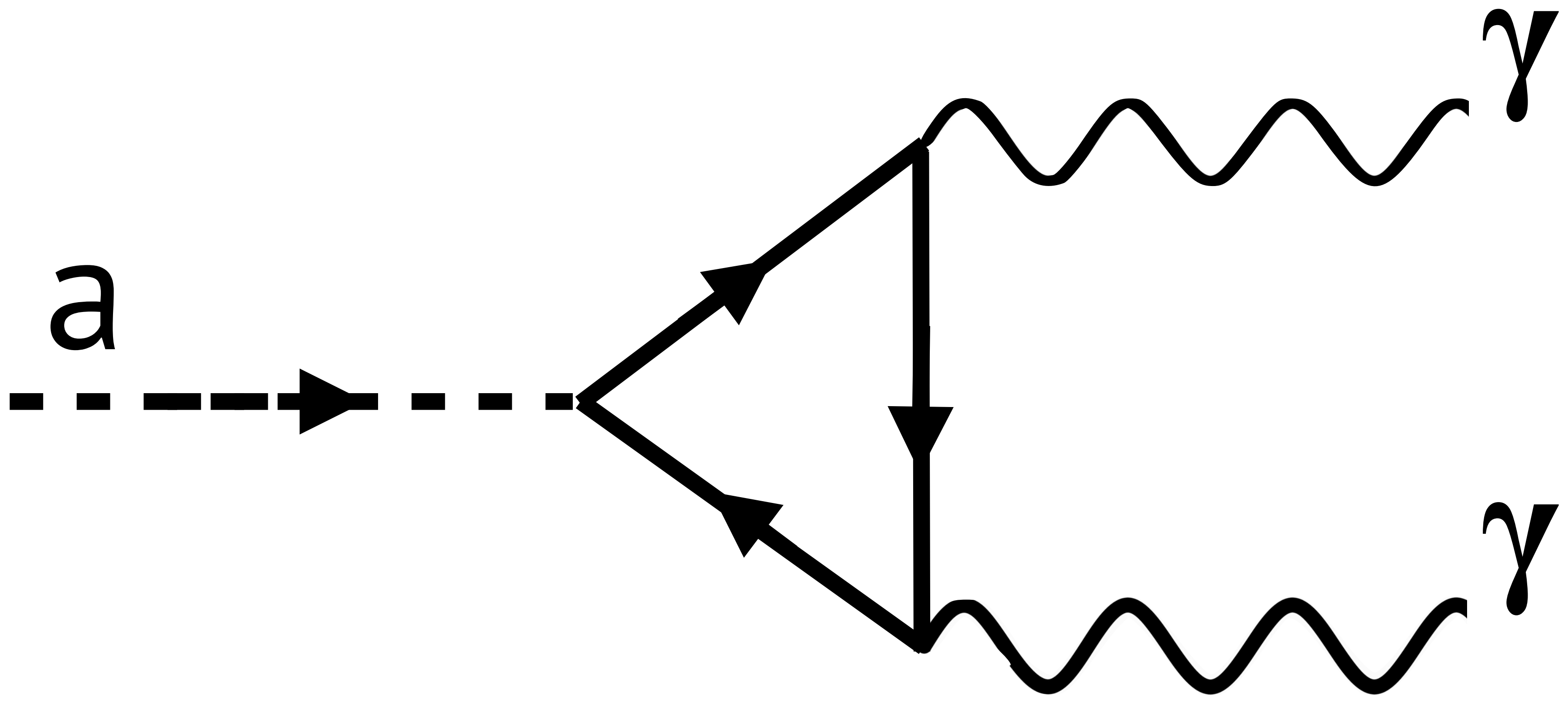
Lagrangian ALP-photon mixing



Lagrangian ALP-photon mixing

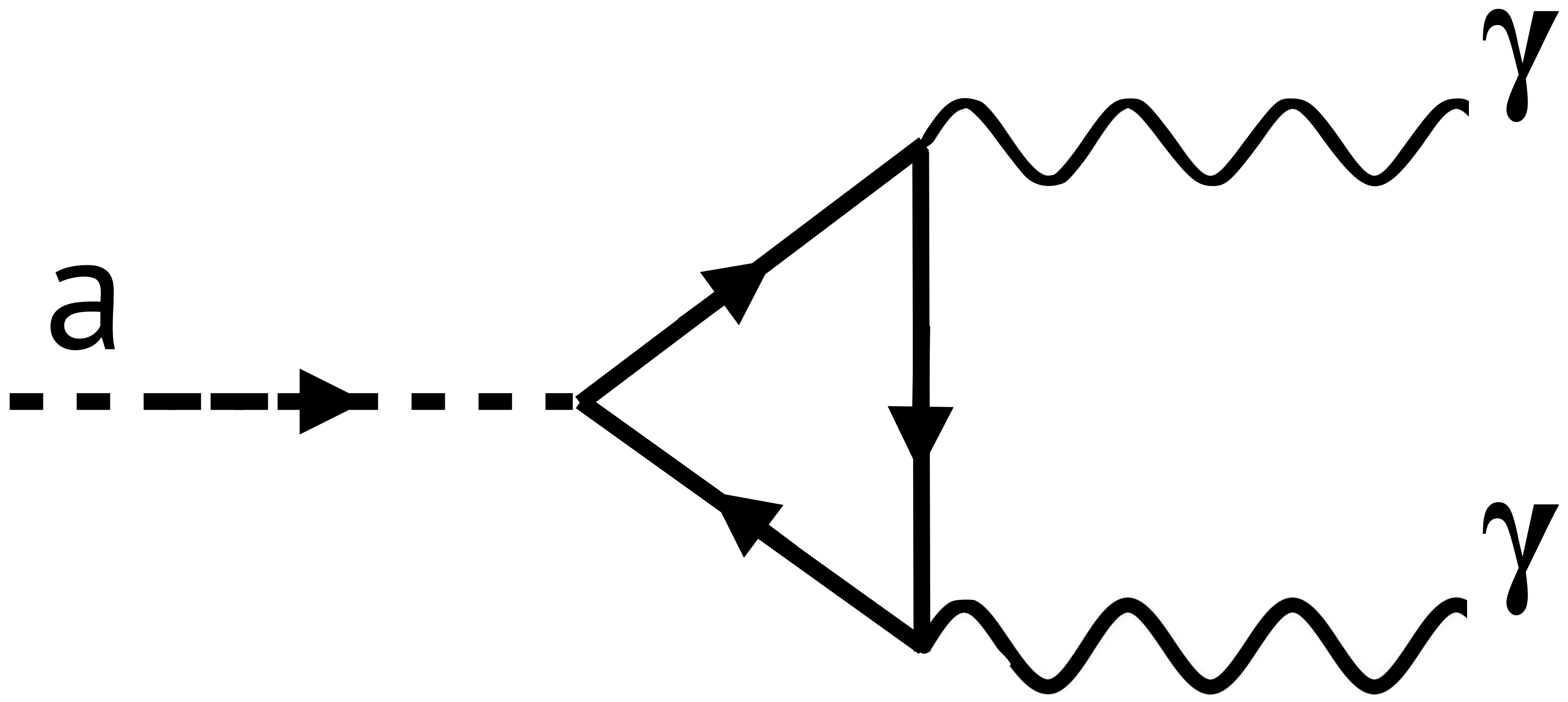
$$\mathscr{L}_{a\gamma} = g_{a\gamma} \overrightarrow{E}_{\gamma} \cdot \overrightarrow{B} \, a$$

axion-like particles



Lagrangian ALP-photon mixing

$$\mathscr{L}_{a\gamma} = \underbrace{g_{a\gamma}}_{\text{coupling constant}} \overrightarrow{E}_{\gamma} \cdot \overrightarrow{B}\, a$$



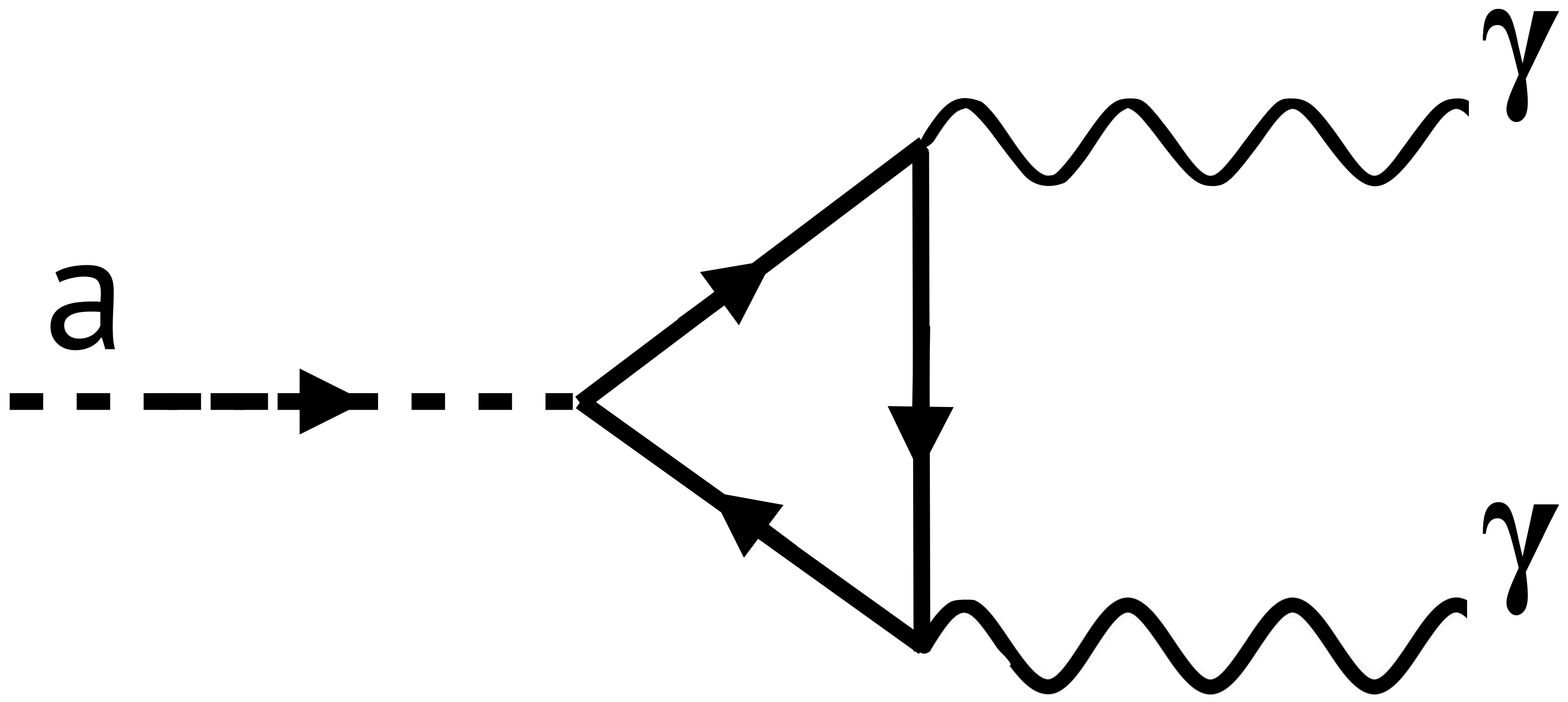
Lagrangian ALP-photon mixing

$$\mathscr{L}_{a\gamma} = g_{a\gamma} \vec{E}_{\gamma} \cdot \vec{B}\, a$$

coupling constant

photon polarisation

axion-like particles



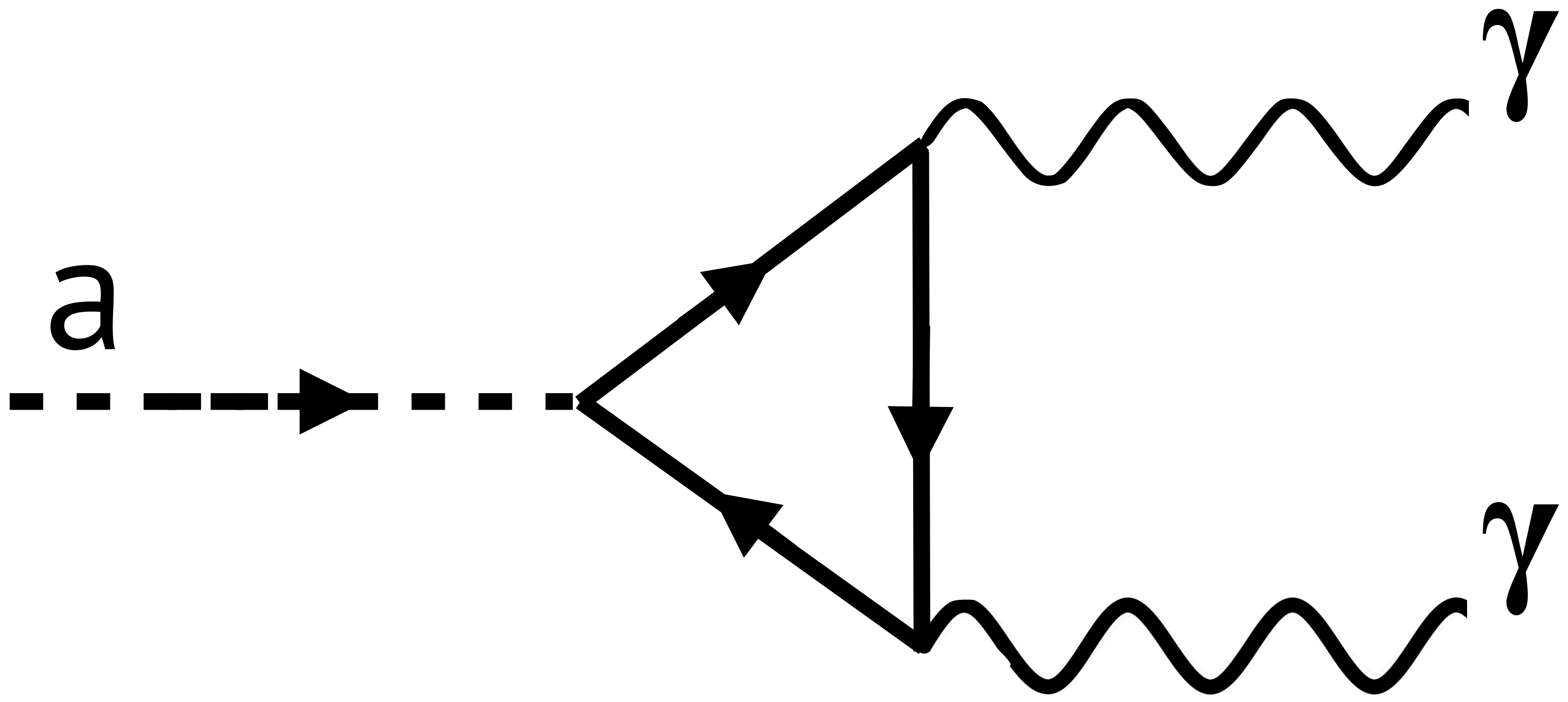
Lagrangian ALP-photon mixing

$$\mathscr{L}_{a\gamma} = g_{a\gamma} \vec{E}_{\gamma} \cdot \vec{B}\, a$$

coupling constant

photon polarisation

external magnetic field



Lagrangian ALP-photon mixing

$$\mathscr{L}_{a\gamma} = g_{a\gamma} \overrightarrow{E}_{\gamma} \cdot \overrightarrow{B}\, a$$

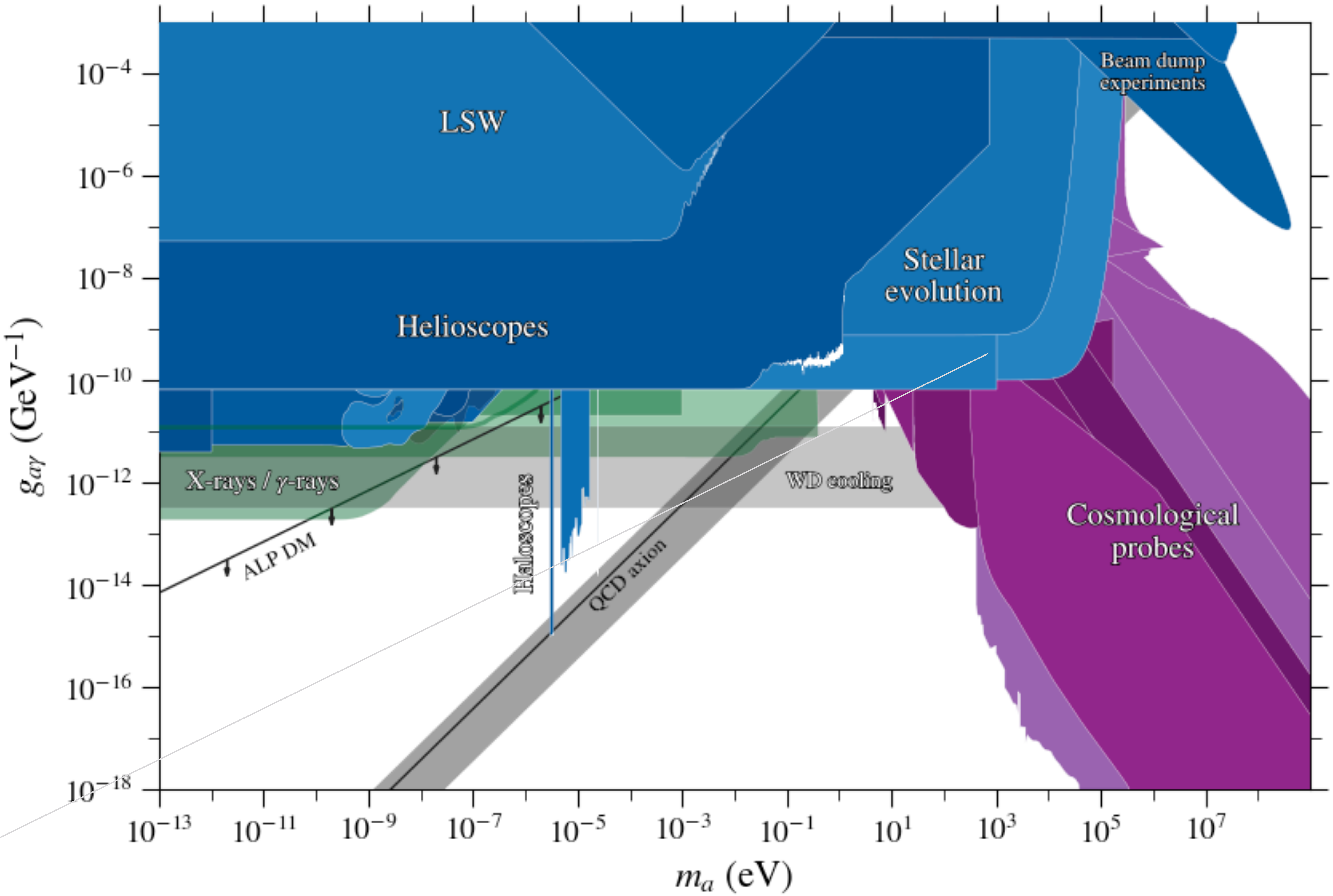
coupling constant

photon polarisation

external magnetic field

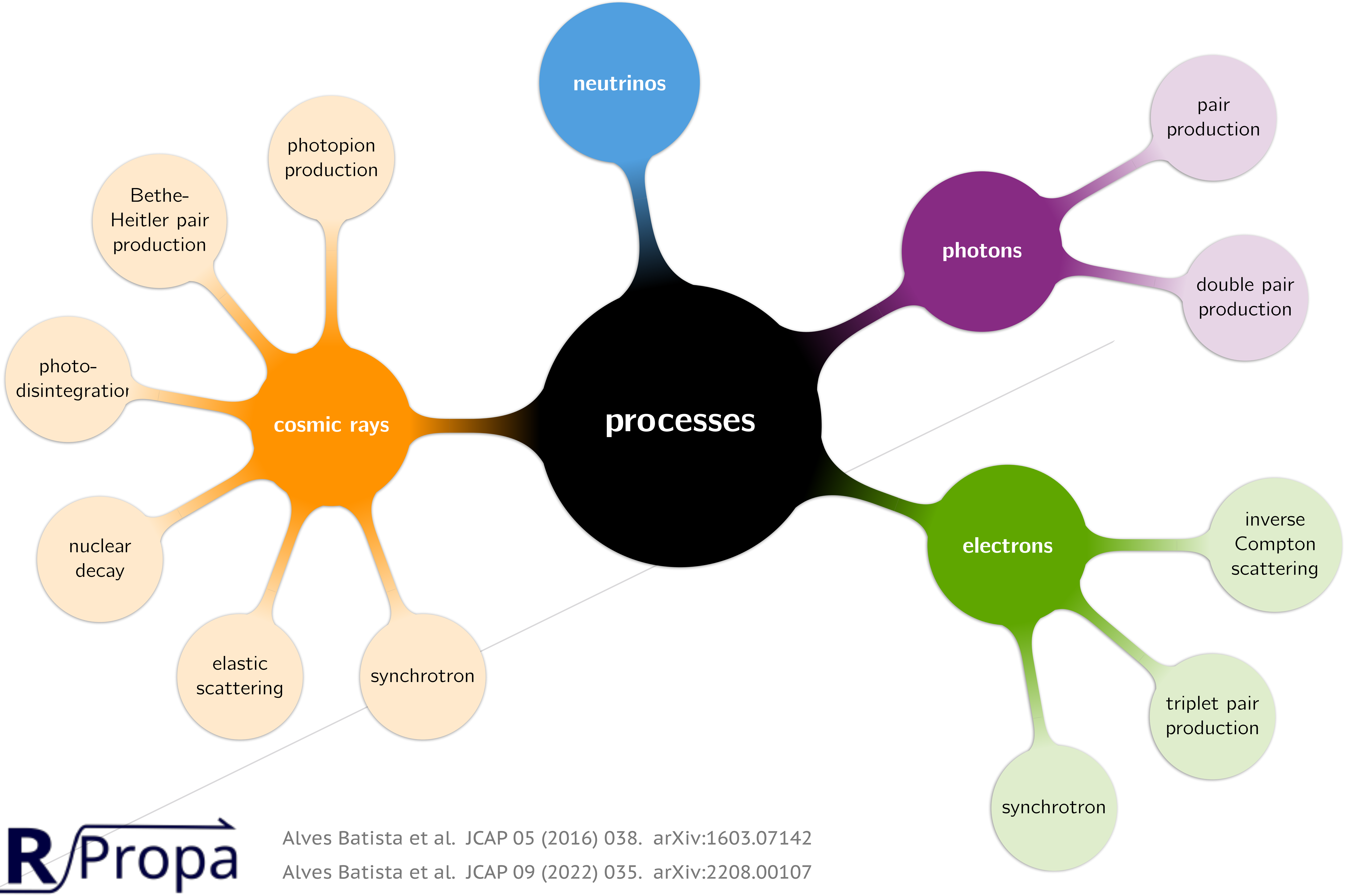
ALP field

axion-like particles: parameter space



https://github.com/me-manu/gammaALPsPlot

Monte Carlo simulations of ALP-photon mixing

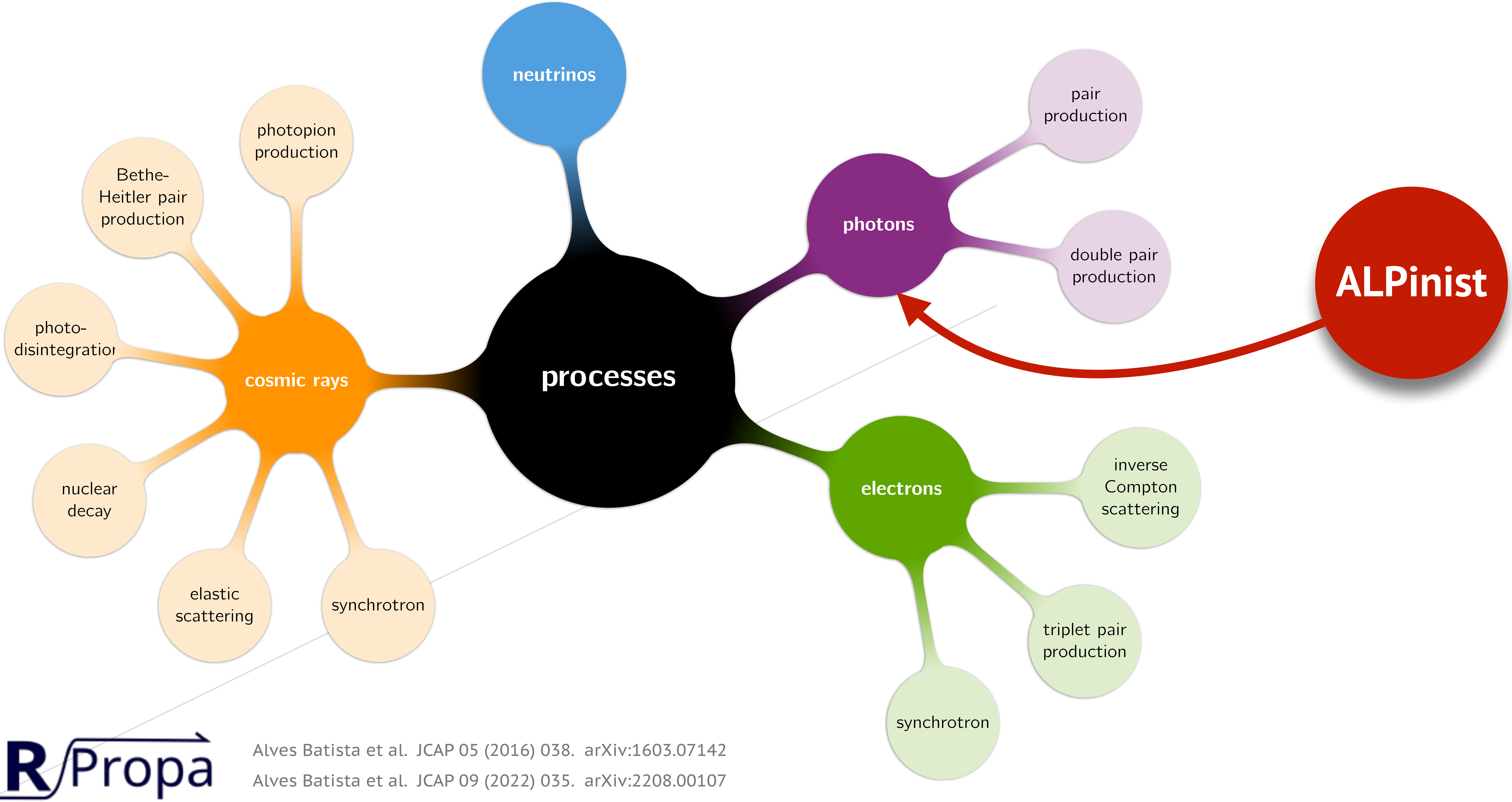


CRPropa

Alves Batista et al. JCAP 05 (2016) 038. arXiv:1603.07142

Alves Batista et al. JCAP 09 (2022) 035. arXiv:2208.00107

Monte Carlo simulations of ALP-photon mixing

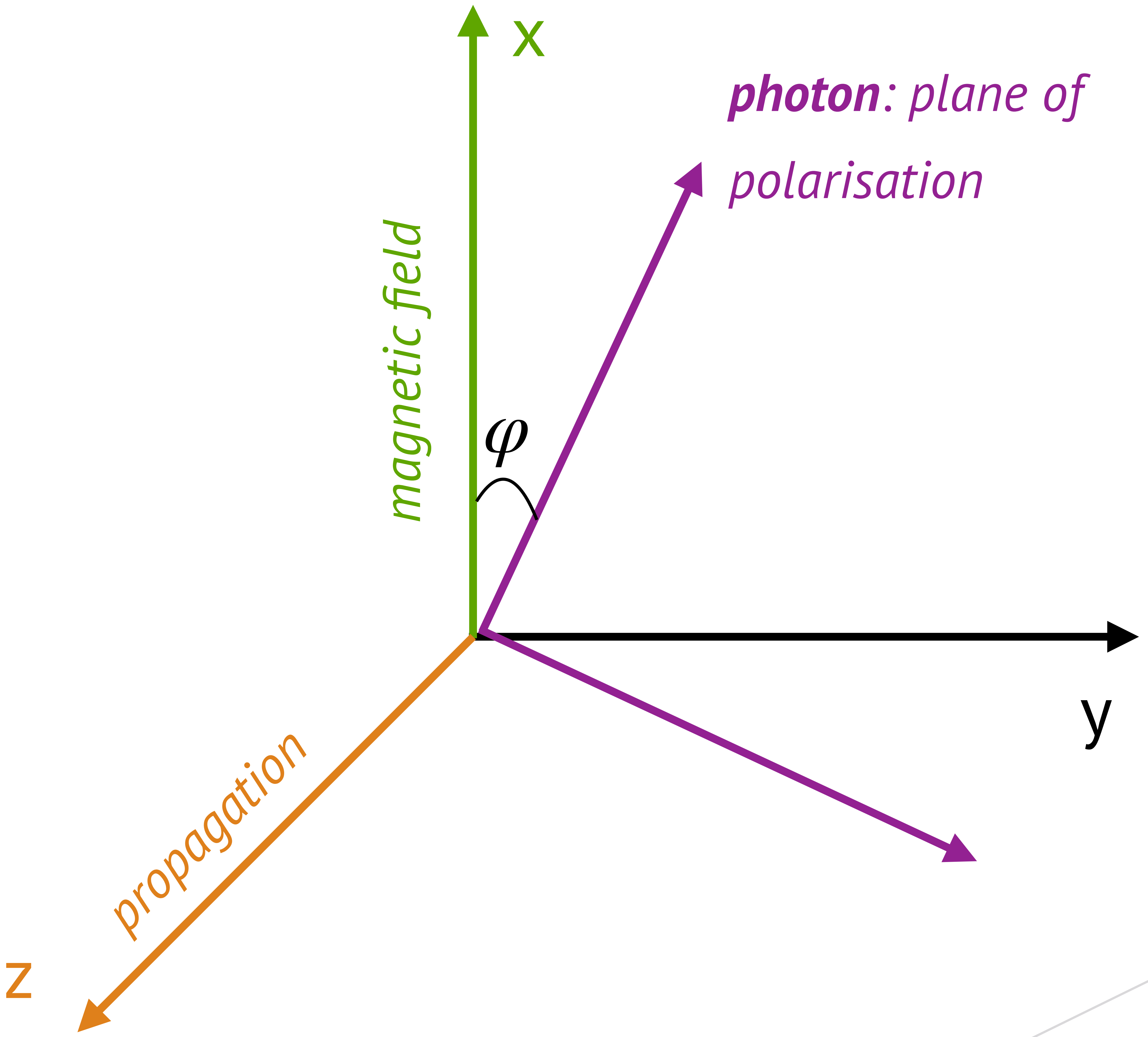


CRPropa

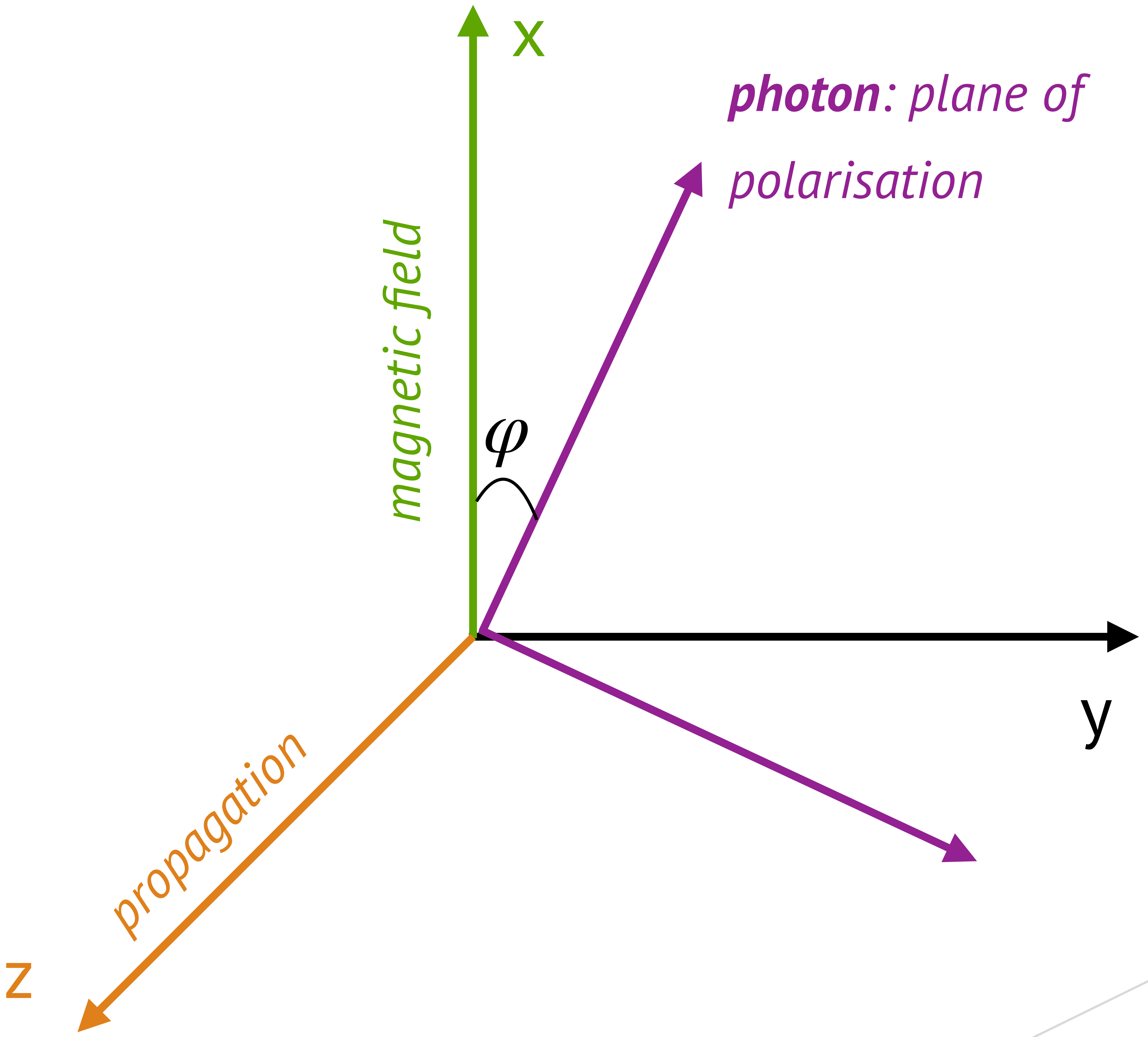
Alves Batista et al. JCAP 05 (2016) 038. arXiv:1603.07142

Alves Batista et al. JCAP 09 (2022) 035. arXiv:2208.00107

ALPinist: implementation



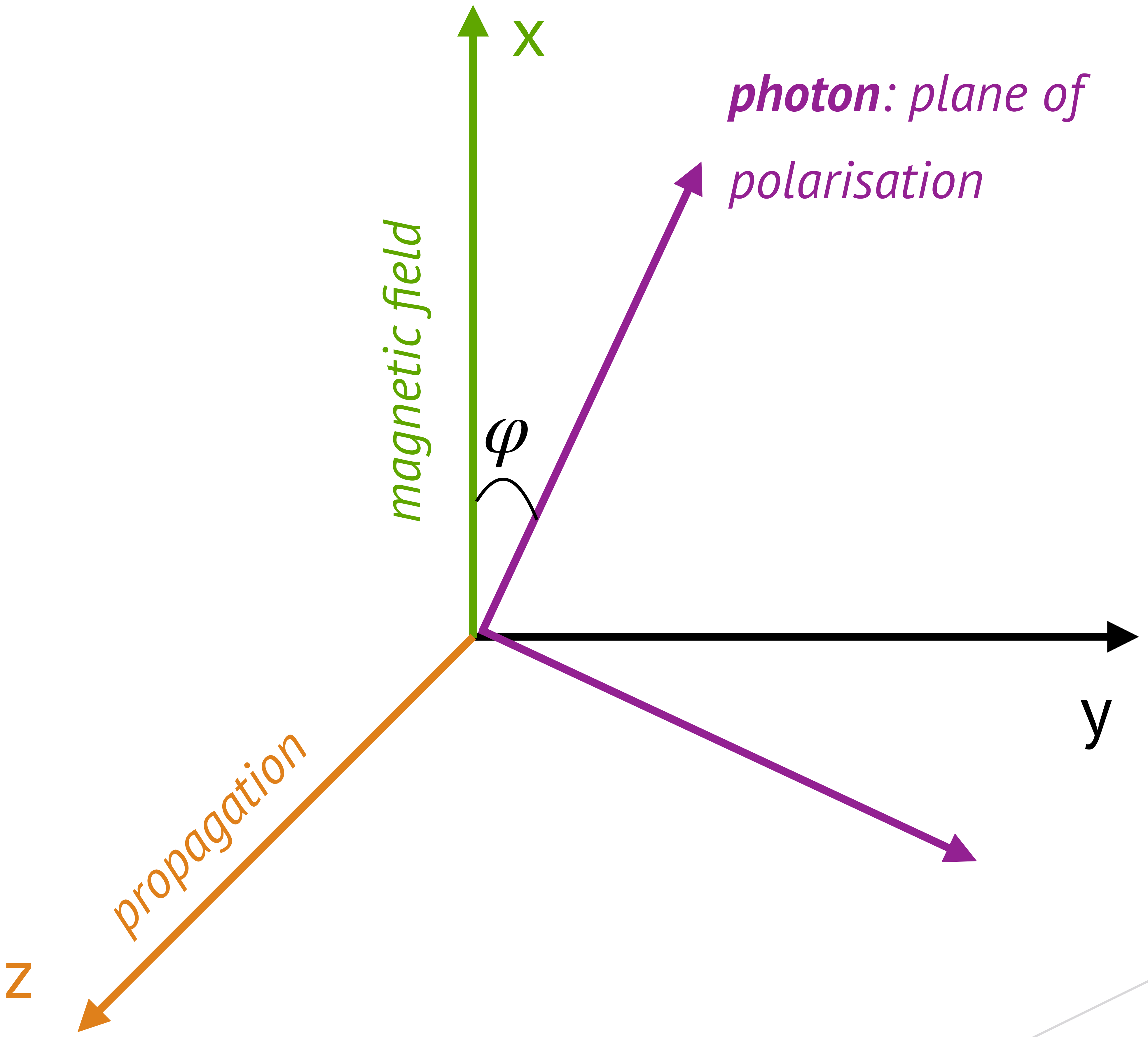
ALPinist: implementation



equation of motion

$$\left(E - i\frac{\partial}{\partial z} + \frac{1}{2E}\mathbb{M}\right)\vec{\mathscr{A}} = 0$$

ALPinist: implementation

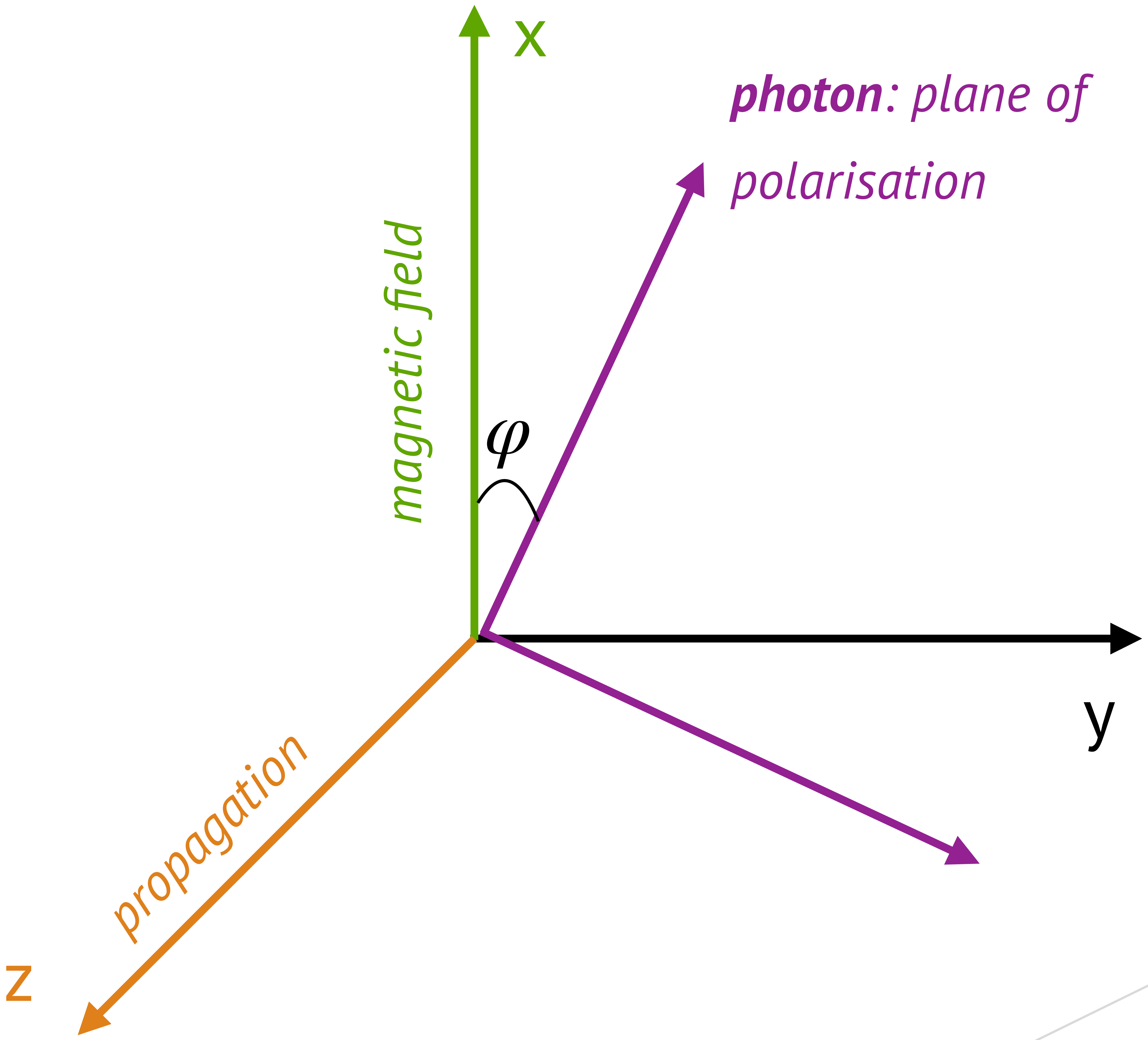


three-state quantum system

equation of motion

$$\left(E - i\frac{\partial}{\partial z} + \frac{1}{2E}\mathbb{M}\right)\vec{\mathscr{A}} = 0$$

ALPinist: implementation



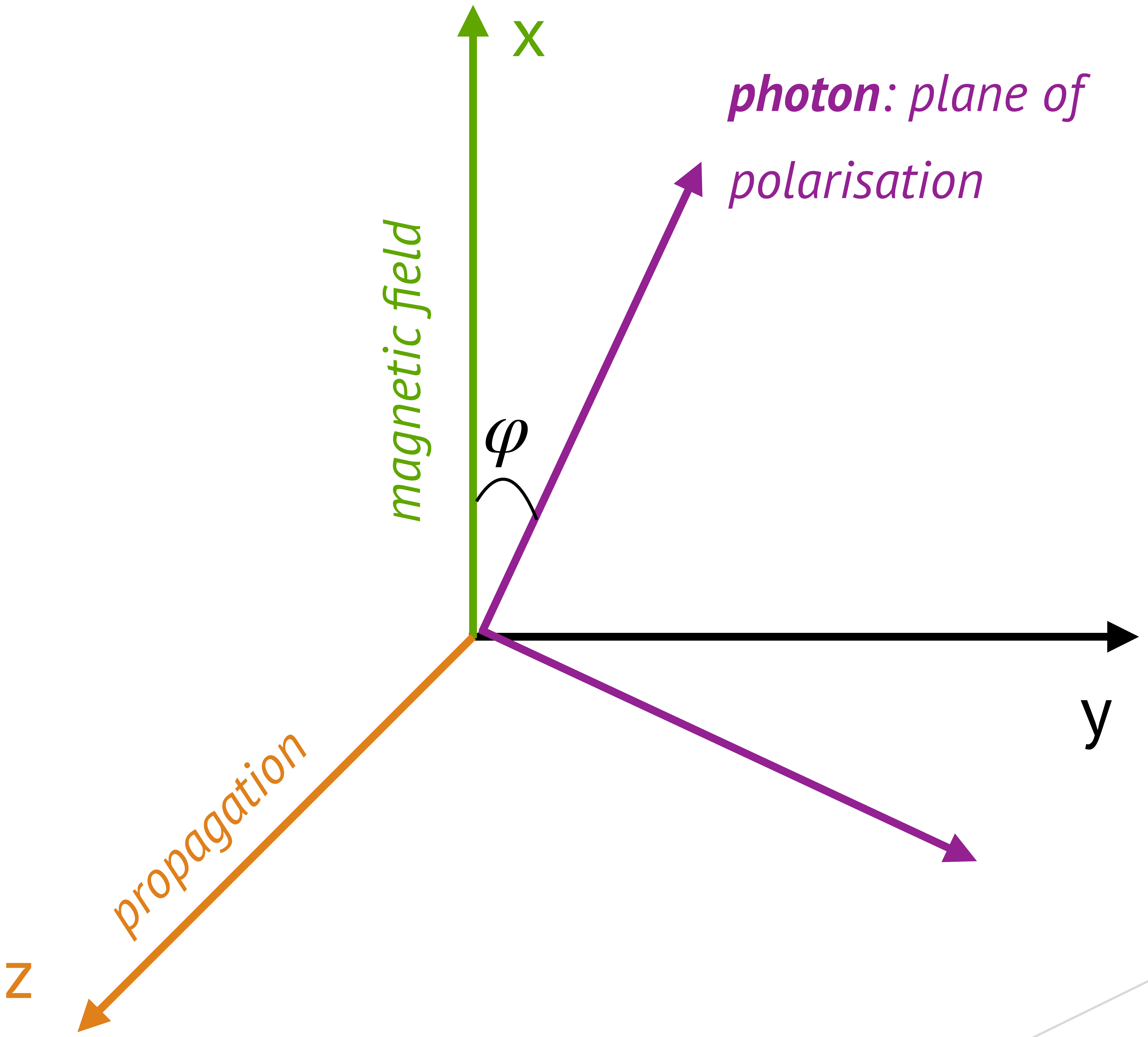
three-state quantum system

$$\mathscr{A}(z) = \begin{pmatrix} A_1(z) \\ A_2(z) \\ a(z) \end{pmatrix}$$

equation of motion

$$\left(E - i\frac{\partial}{\partial z} + \frac{1}{2E}\mathbb{M} \right) \vec{\mathscr{A}} = 0$$

ALPinist: implementation



three-state quantum system

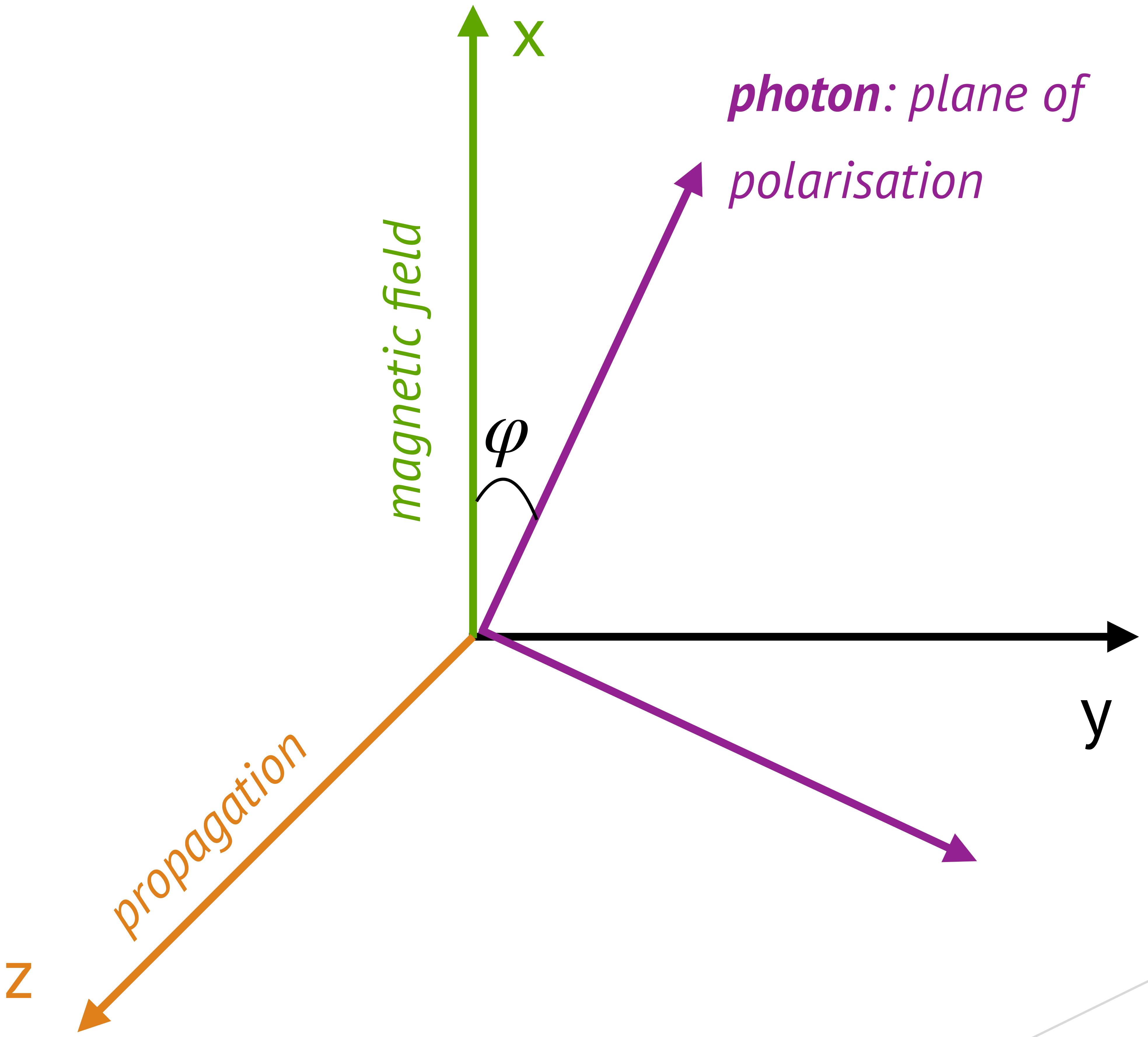
$$\mathscr{A}(z) = \begin{pmatrix} A_1(z) \\ A_2(z) \\ a(z) \end{pmatrix}$$

photon polarisations

equation of motion

$$\left(E - i\frac{\partial}{\partial z} + \frac{1}{2E}\mathbb{M} \right) \vec{\mathscr{A}} = 0$$

ALPinist: implementation



three-state quantum system

$$\mathscr{A}(z) = \begin{pmatrix} A_1(z) \\ A_2(z) \\ a(z) \end{pmatrix}$$

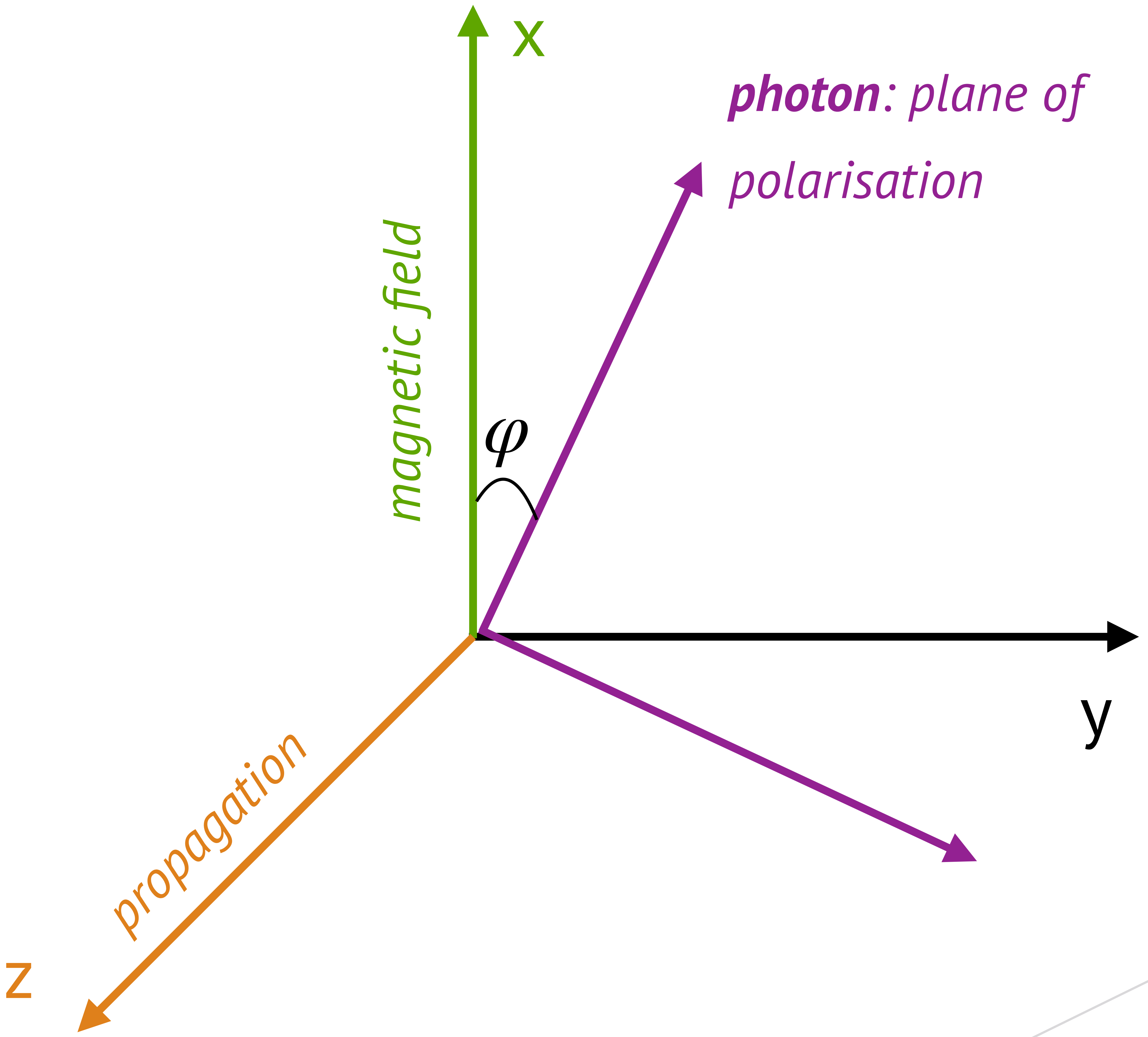
photon polarisations

ALP field

equation of motion

$$\left(E - i\frac{\partial}{\partial z} + \frac{1}{2E}\mathbb{M} \right) \vec{\mathscr{A}} = 0$$

ALPinist: implementation



three-state quantum system

$$\mathscr{A}(z) = \begin{pmatrix} A_1(z) \\ A_2(z) \\ a(z) \end{pmatrix}$$

photon polarisations

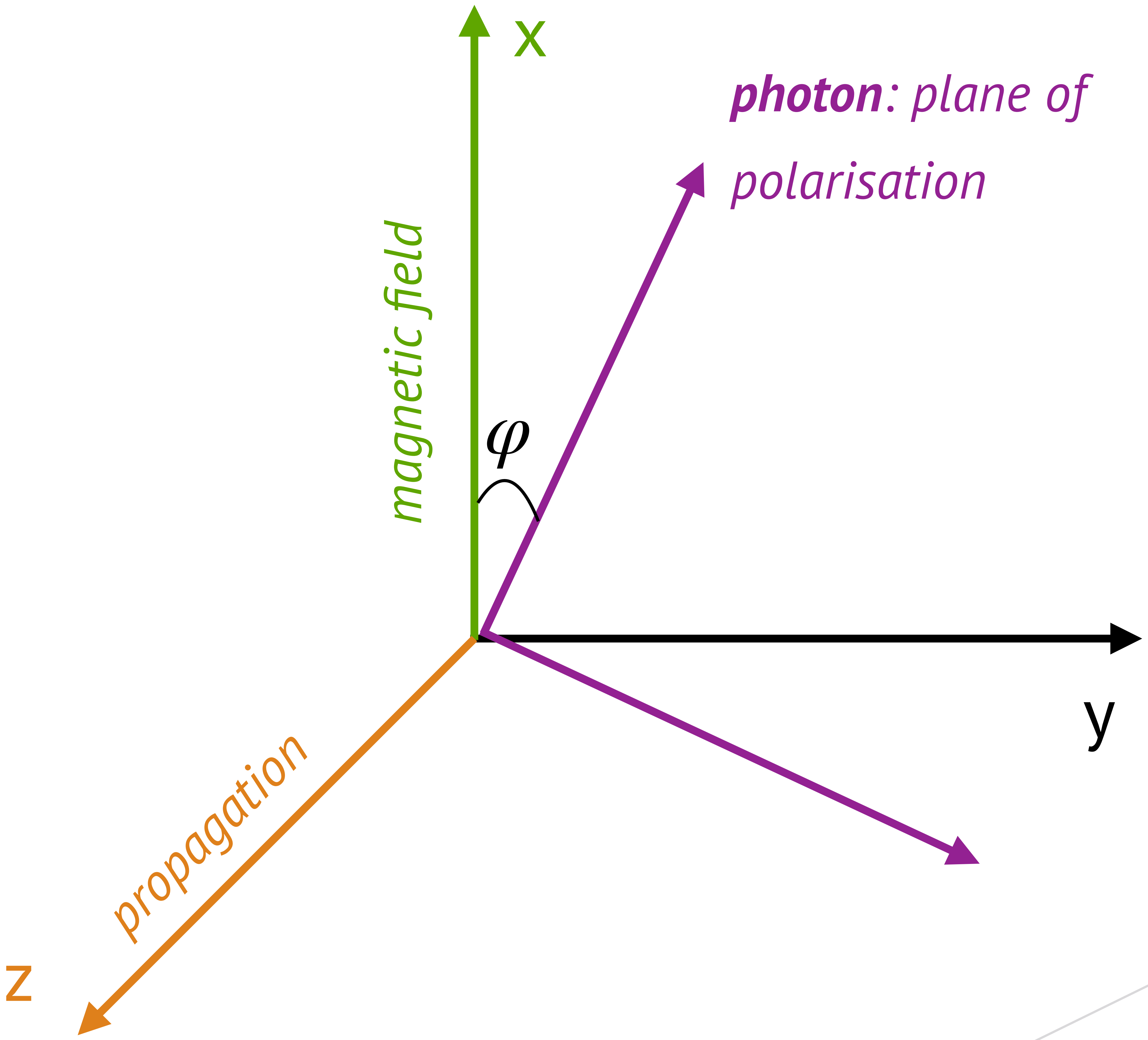
ALP field

source

equation of motion

$$\left(E - i\frac{\partial}{\partial z} + \frac{1}{2E}\mathbb{M}\right)\vec{\mathscr{A}} = 0$$

ALPinist: implementation



three-state quantum system

$$\mathscr{A}(z) = \begin{pmatrix} A_1(z) \\ A_2(z) \\ a(z) \end{pmatrix}$$

photon polarisations

ALP field

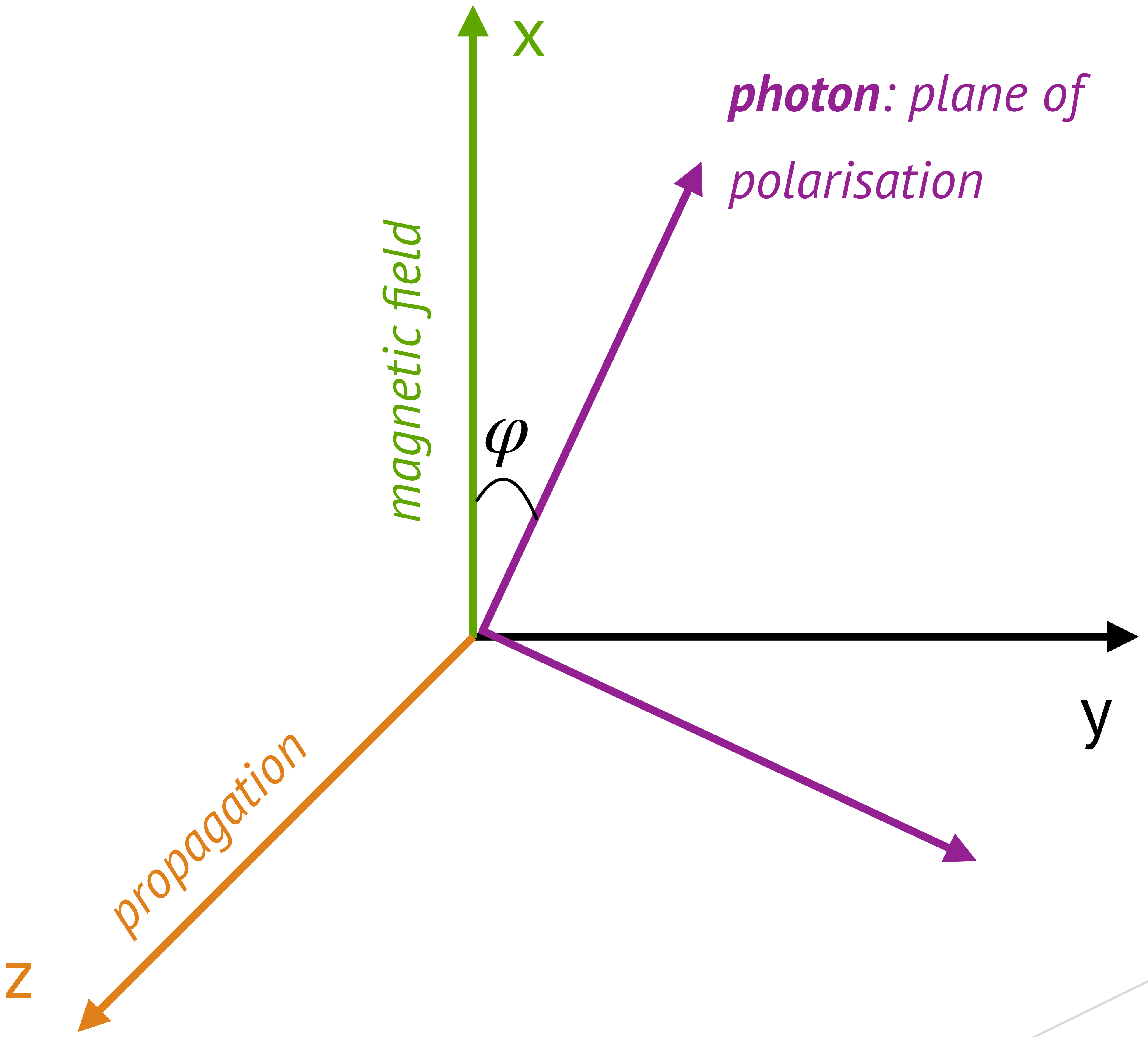
source

- initial ALP state needed to build wave function
- initial photon polarisation also required (A_1, A_2)
- wavefunction built at source

equation of motion

$$\left(E - i\frac{\partial}{\partial z} + \frac{1}{2E}\mathbb{M} \right) \vec{\mathscr{A}} = 0$$

ALPinist: implementation



three-state quantum system

$$\mathscr{A}(z) = \begin{pmatrix} A_1(z) \\ A_2(z) \\ a(z) \end{pmatrix}$$

photon polarisations

ALP field

equation of motion

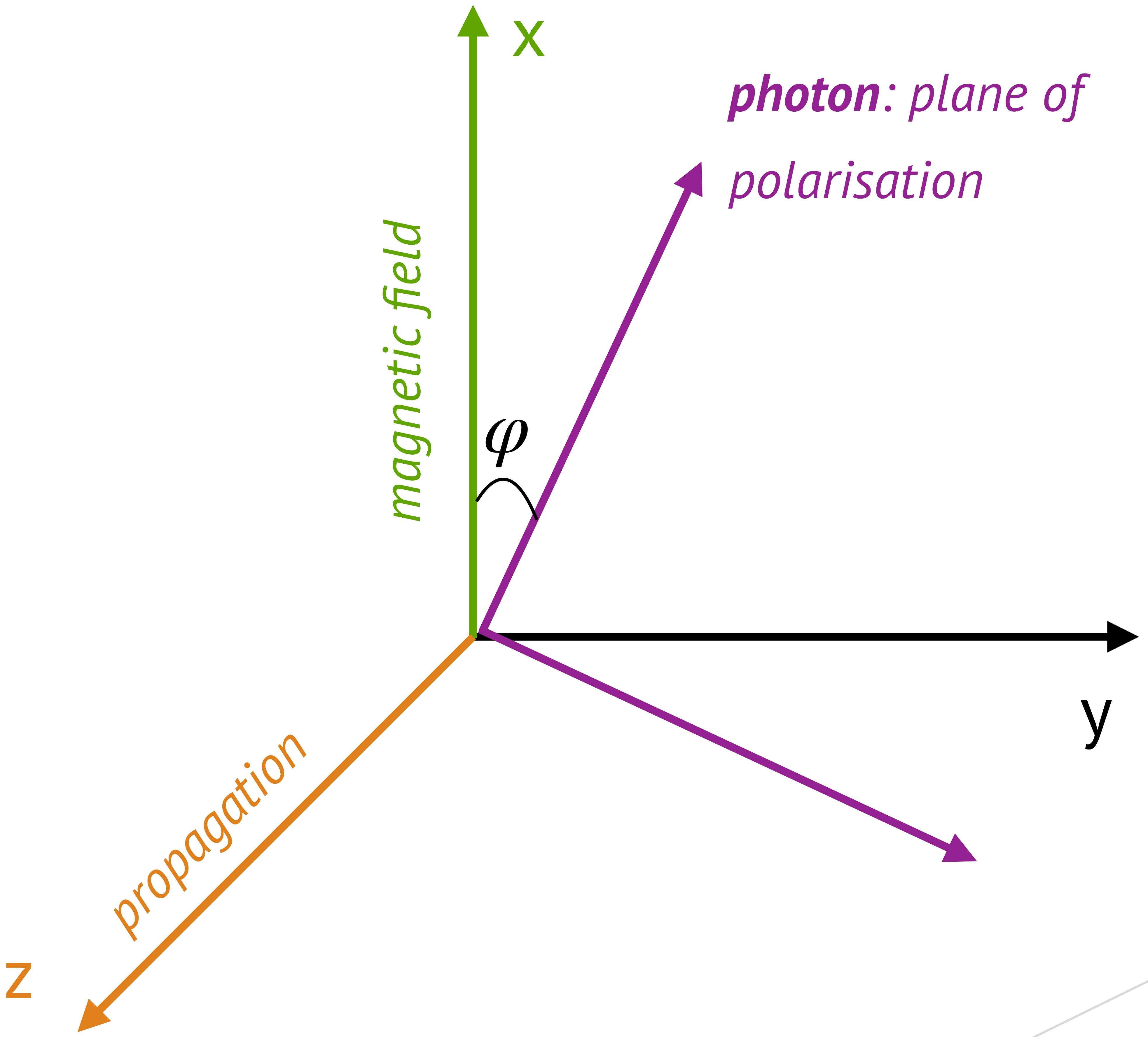
$$\left(E - i\frac{\partial}{\partial z} + \frac{1}{2E}\mathbb{M} \right) \vec{\mathscr{A}} = 0$$

source

- initial ALP state needed to build wave function
- initial photon polarisation also required (A_1, A_2)
- wavefunction built at source

```
source.add(alp.SourceALPState(0., 0.))
source.add(alp.SourceNoPolarisation())
```

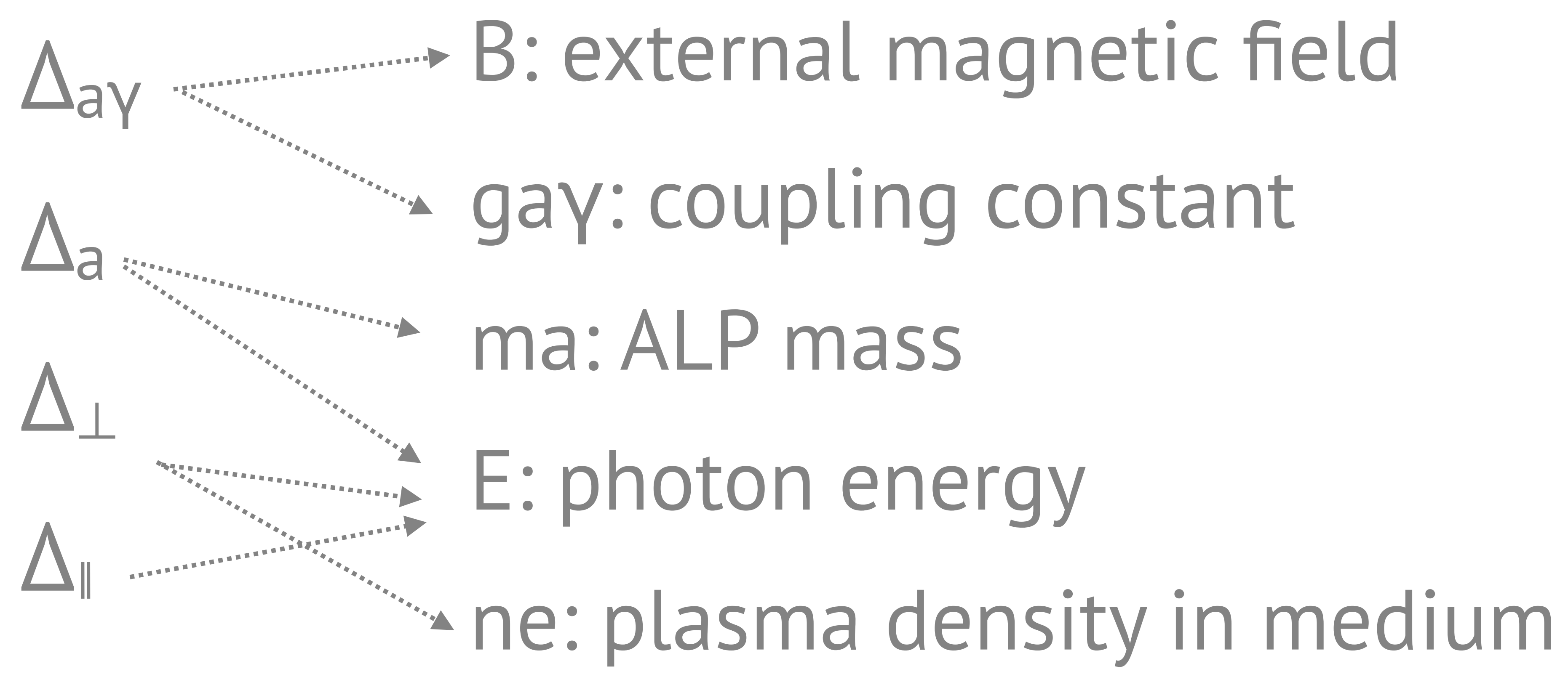
ALPinist: implementation



mixing matrix

$$
\mathbb{M} = \begin{pmatrix} \Delta_{\parallel} \cos^2 \varphi + \Delta_{\perp} \sin^2 \varphi & \left(\Delta_{\parallel} - \Delta_{\perp}\right) \sin \varphi \cos \varphi & \Delta_{a\gamma} \cos \varphi \\ \left(\Delta_{\parallel} - \Delta_{\perp}\right) \sin \varphi \cos \varphi & \Delta_{\parallel} \sin^2 \varphi + \Delta_{\perp} \cos^2 \varphi & \Delta_{a\gamma} \sin \varphi \\ \Delta_{a\gamma} \cos \varphi & \Delta_{a\gamma} \sin \varphi & \Delta_{a} \end{pmatrix}
$$

depends on



B: external magnetic field

gaγ: coupling constant

ma: ALP mass

E: photon energy

ne: plasma density in medium

equation of motion

$$
\left(E - i \frac{\partial}{\partial z} + \frac{1}{2E} \mathbb{M} \right) \vec{\mathscr{A}} = 0
$$

ALPinist: implementation

ALPinist: implementation

plasma density

ALPinist: implementation

plasma density

- influences the ALP-photon coupling

ALPinist: implementation

plasma density

- influences the ALP-photon coupling
- available types:

ALPinist: implementation

plasma density

- influences the ALP-photon coupling
- available types:
    - uniform

ALPinist: implementation

plasma density

- influences the ALP-photon coupling
- available types:
  - uniform
  - grid

ALPinist: implementation

plasma density

- influences the ALP-photon coupling
- available types:
  - uniform
  - grid
  - turbulent *(to-do list)*

ALPinist: implementation

plasma density

- influences the ALP-photon coupling
- available types:
    - uniform
    - grid
    - turbulent *(to-do list)*

```
density = alp.PlasmaDensityUniform(densityIGM)
```

ALPinist: implementation

plasma density

- influences the ALP-photon coupling
- available types:
    - uniform
    - grid
    - turbulent *(to-do list)*

```
density = alp.PlasmaDensityUniform(densityIGM)
```

ALP-photon mixing

ALPinist: implementation

plasma density

- influences the ALP-photon coupling
- available types:
    - uniform
    - grid
    - turbulent *(to-do list)*

```
density = alp.PlasmaDensityUniform(densityIGM)
```

ALP-photon mixing

```
alpPhoton = alp.ALPPhotonMixing(axionMass * eV / c_light ** 2, coupling / eV, bField, density, .5)
```

ALPinist: implementation

ALPinist: implementation

- mixing matrix is diagonalised at each step (using Eigen)

ALPinist: implementation

- mixing matrix is diagonalised at each step (using Eigen)
- the wave function is collapsed at each step to decide on interaction

ALPinist: implementation

- mixing matrix is diagonalised at each step (using Eigen)
- the wave function is collapsed at each step to decide on interaction
- ALP is internally coded 51 in CRPropa

ALPinist: implementation

- mixing matrix is diagonalised at each step (using Eigen)
- the wave function is collapsed at each step to decide on interaction
- ALP is internally coded 51 in CRPropa
- random numbers compared to probability of photon-ALP conversion to decide if oscillation occurs

ALPinist: code validation

For a homogeneous magnetic field and medium, the equations of motion can be solved analytically to determine the conversion probability.

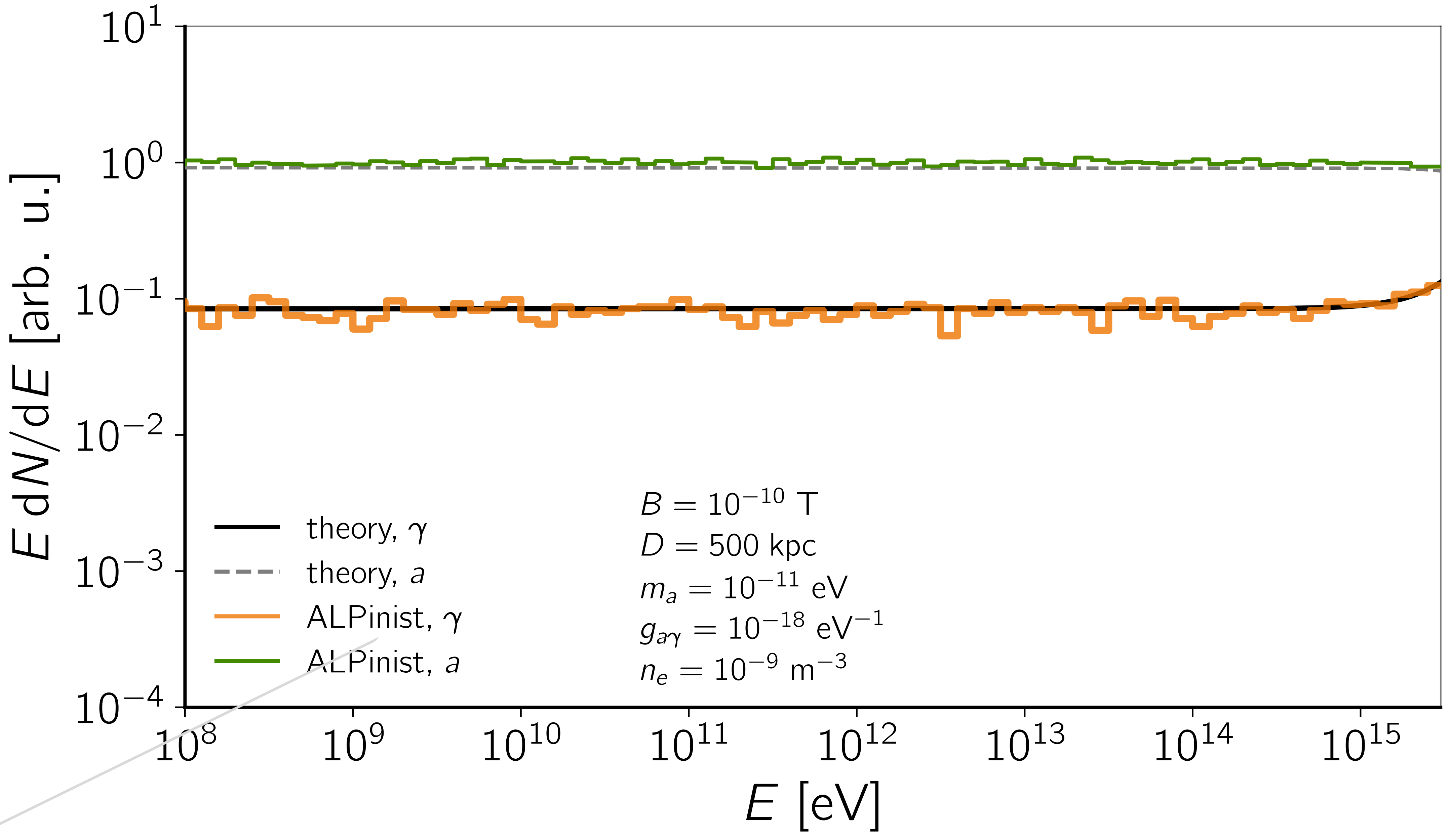
conversion probability

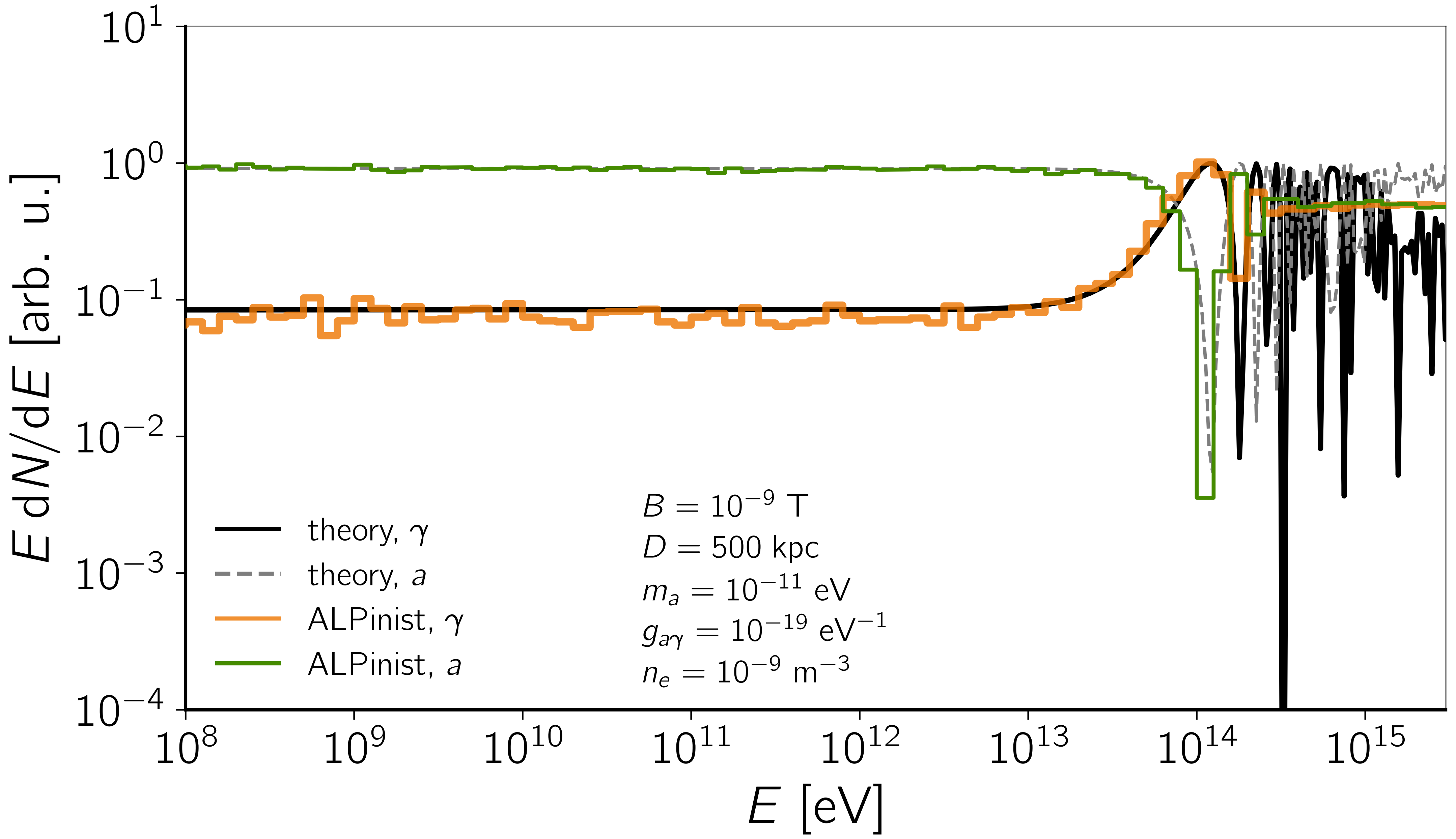
$$P_{a\gamma}(z) = \sin^2\theta \sin^2\left(\frac{z}{2}\sqrt{\left(\Delta_a - \Delta_{\rm pl}\right)^2 + 4\Delta_{a\gamma}}\right)$$

mixing angle

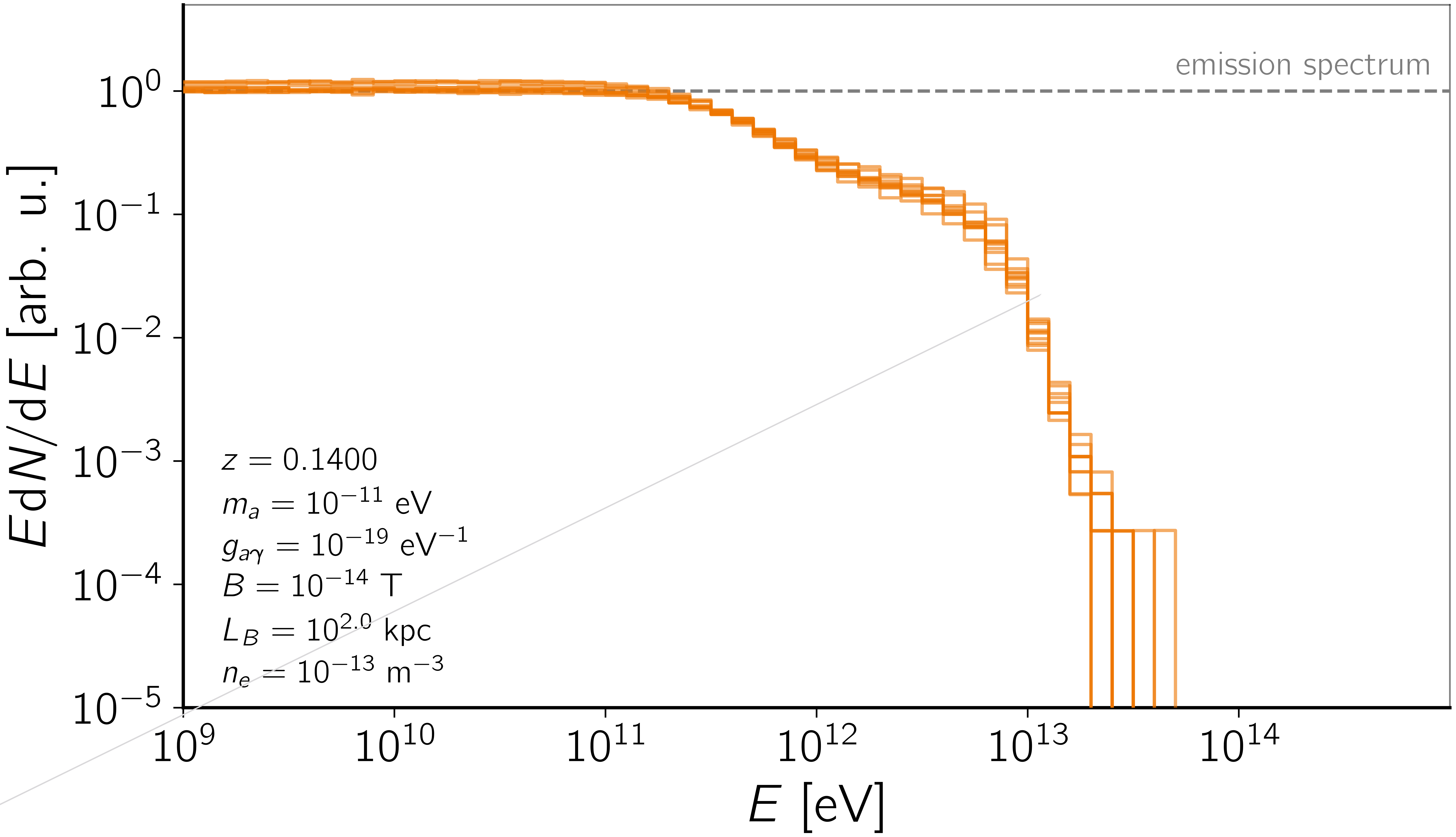
$$\theta = \frac{1}{2}\arcsin\left(\frac{2\Delta_{a\gamma}}{\sqrt{\left(\Delta_a - \Delta_{\rm pl}\right)^2 + 4\Delta_{a\gamma}}}\right)$$

For θ=45° mixing is maximal and an oscillatory behaviour sets in.

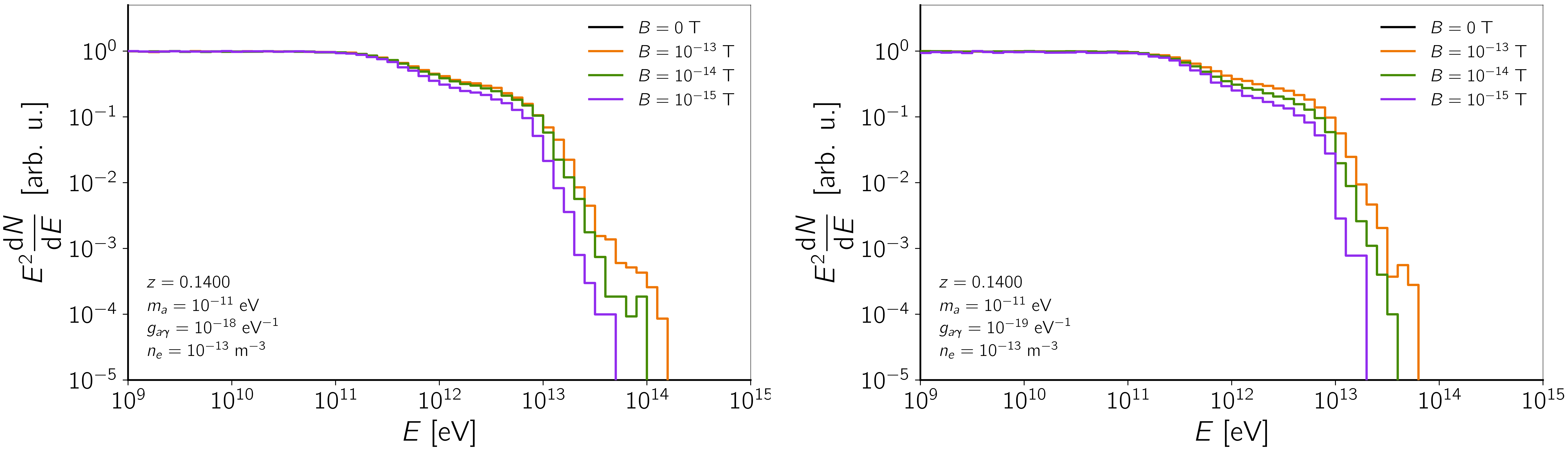




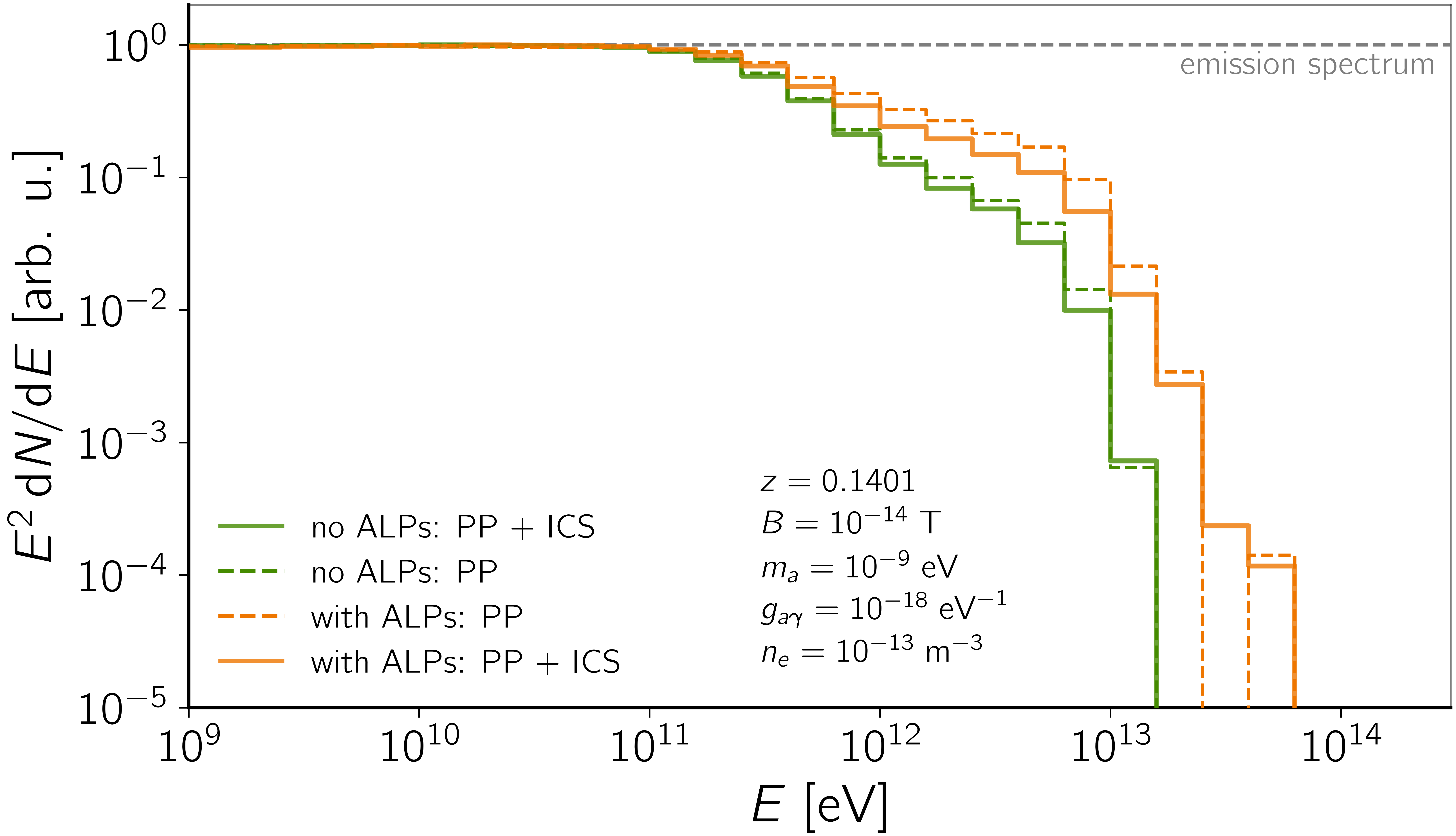
simulations with turbulent magnetic fields



simulations with pair production and turbulent magnetic fields



adding inverse Compton scattering



conclusions and outlook

conclusions and outlook

- new Monte Carlo code for ALP-photon mixing (plugin for CRPropa): **ALPinist**

conclusions and outlook

- new Monte Carlo code for ALP-photon mixing (plugin for CRPropa): **ALPinist**
- advantages:
  - wealth of magnetic-field models available in CRPropa
  - easy treatment of non-uniform media
  - seamless *interaction with CRPropa modules* for particle propagation
  - enables *3D simulations*

conclusions and outlook

- new Monte Carlo code for ALP-photon mixing (plugin for CRPropa): **ALPinist**
- advantages:
  - wealth of magnetic-field models available in CRPropa
  - easy treatment of non-uniform media
  - seamless *interaction with CRPropa modules* for particle propagation
  - enables *3D simulations*
- limitations:
  - the Monte Carlo approach not efficient for the strong mixing regime
  - inverse Compton scattering can only be treated if the initial photons are *unpolarised*

conclusions and outlook

- new Monte Carlo code for ALP-photon mixing (plugin for CRPropa): **ALPinist**
- advantages:
  - wealth of magnetic-field models available in CRPropa
  - easy treatment of non-uniform media
  - seamless *interaction with CRPropa modules* for particle propagation
  - enables *3D simulations*
- limitations:
  - the Monte Carlo approach not efficient for the strong mixing regime
  - inverse Compton scattering can only be treated if the initial photons are *unpolarised*
- first-ever simulations of electromagnetic cascades including all effects: pair production + inverse Compton scattering + ALP-photon mixing
  - cascades initiated via inverse Compton can provide an additional contribution to the gamma-ray flux