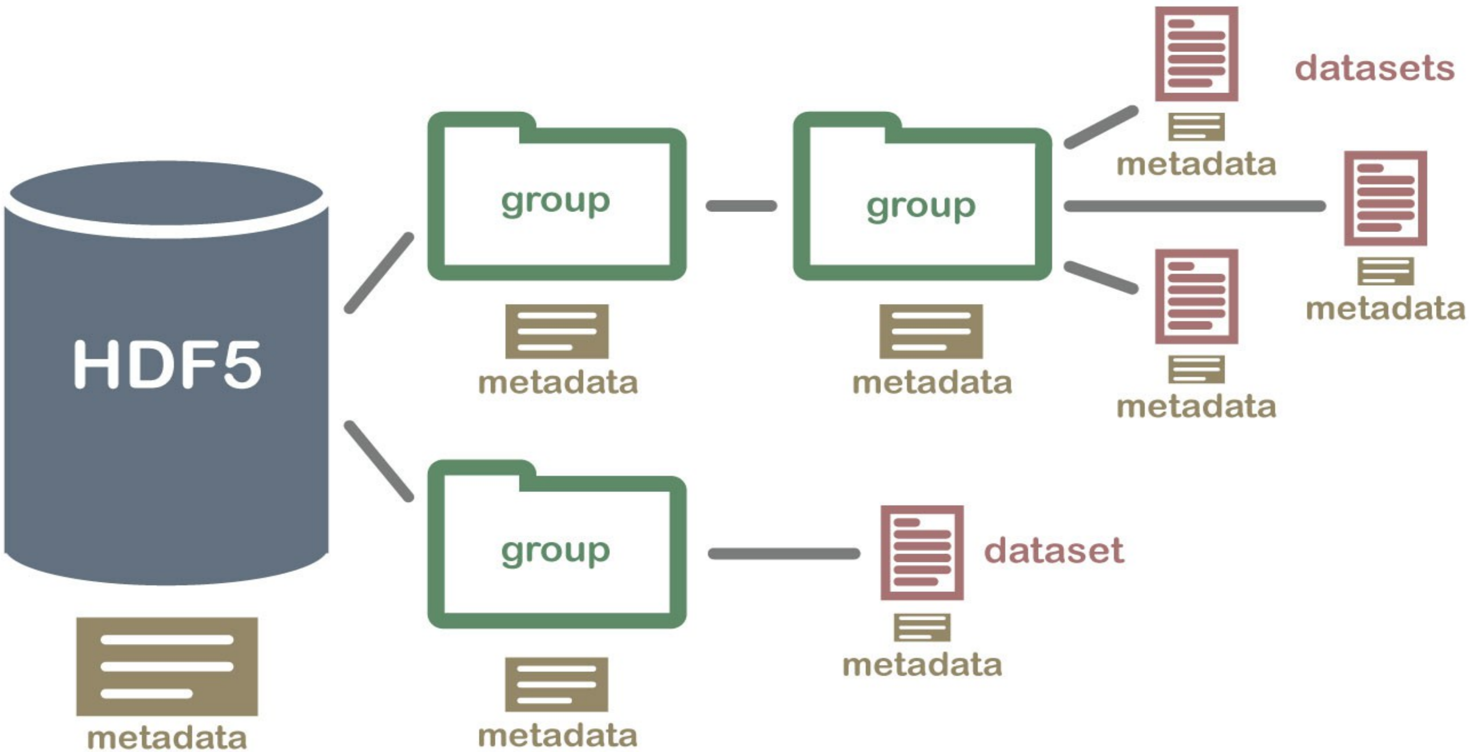
h5 / hdf5

Hierarchical Data Formats
structure, plotting, converting



David Spataro

Example

/nfs/dust/luxe/group/MCProduction/Signal/ptarmigan-v0.7-preview/brem_laser/phase0/5.00_particles.h5

```
import nexusformat.nexus as nx
f = nx.nxload('5.00_particles.h5')
print(f.tree)
```

Items which are not groups themselves are datasets, scalars, arrays etc. possible

Top Level

“build“ is a group structure, has items with labels “branch““, etc.

```
root:NXroot
  build:NXgroup
    branch = 'devel'
    commit-hash = 'f2598613dd16a533fa4152e6650f6e370654ceb5'
    features = 'fits-output,hdf5-output,with-mpi'
    version = '0.6.2'
```

array with single values

```
photon:NXgroup
  a0_at_creation = float64(505743)
  momentum = ('<f8', (4,))(505743)
  n_pos = float64(505743)
  position = ('<f8', (4,))(505743)
  weight = float64(505743)
  xi -> /a0_at_creation
```

2-dim array
little-endian
single-precision float
8 byte

Example

/nfs/dust/luxe/group/MCProduction/Signal/ptarmigan-v0.7-preview/brem_laser/phase0/5.00_particles.h5

```
import nexusformat.nexus as nx
f = nx.nxload('5.00_particles.h5')
print(f.tree)
```

Items which are not groups themselves are datasets, scalars, arrays etc. possible

Top Level

“build“ is a group structure, has items with labels “branch““, etc.

```
root:NXroot
  build:NXgroup
    branch = 'devel'
    commit-hash = 'f2598613dd16a533fa4152e6650f6e370654ceb5'
    features = 'fits-output,hdf5-output,with-mpi'
    version = '0.6.2'
```

array with single values

```
photon:NXgroup
  a0_at_creation = float64(505743)
  momentum = ('<f8', (4,))(505743)
  n_pos = float64(505743)
  position = ('<f8', (4,))(505743)
  weight = float64(505743)
  xi -> /a0_at_creation
```

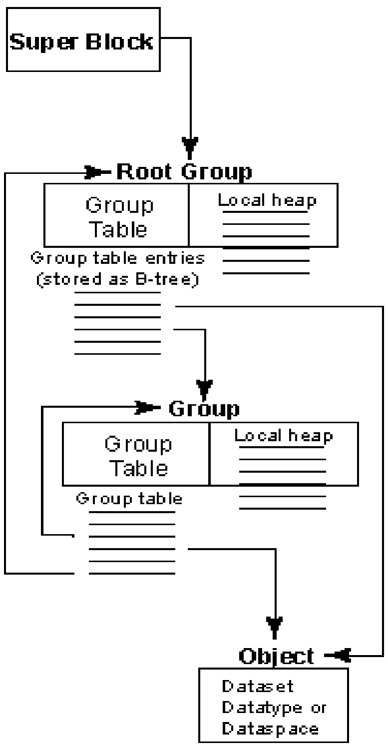
2-dim array
little-endian
single-precision float
8 byte

h5/hdf5 vs. ROOT Trees

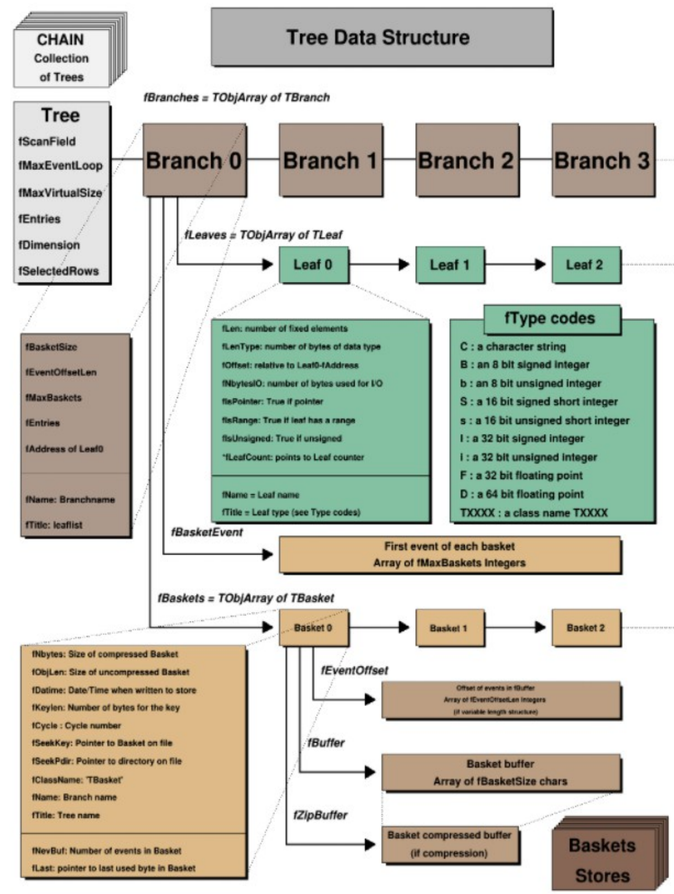
pros and cons of both formats

HDF5:

- groups: grouping structure
- datasets: multidimensional array of record structures



ROOT Trees:



Converting datasets from h5/hdf5 to ROOT I

Using Pyroot with Jupyter on NAF

- Python as “glueing“ language
- Jupyter on NAF: https://confluence.desy.de/display/IS/Jupyter+on+NAF

Seiten / ... / NAF - National Analysis Facility

Jupyter on NAF

Johannes Reppin posted on 21. Aug. 2019 15:00h - last edited by Thomas Hartmann on 16. Jun. 2021 09:50h

Introduction

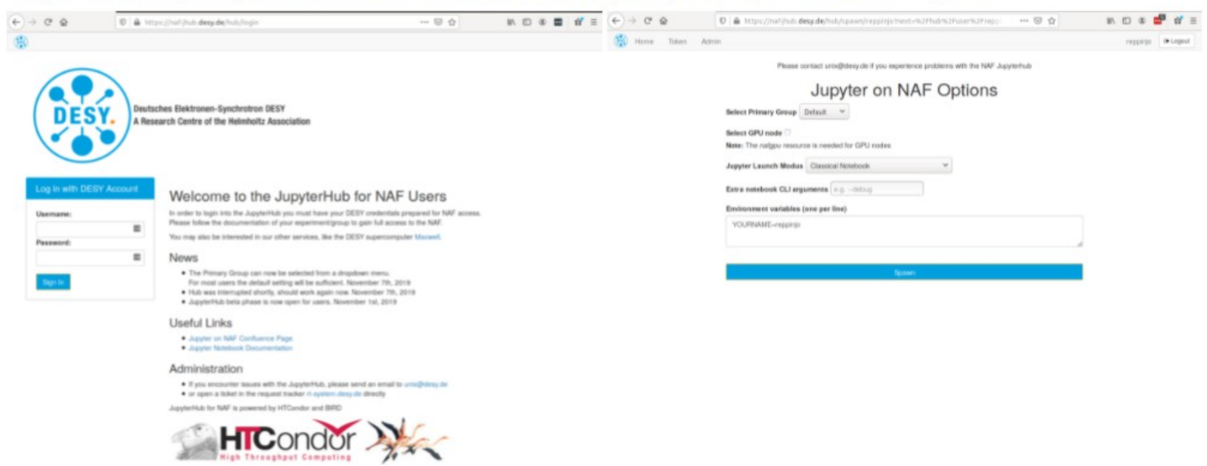
We have a JupyterHub instance on https://naf-jhub.desy.de which is reachable from worldwide IP ranges, not just the DESY network.
In order to use the JupyterHub you need a NAF account, see Getting a NAF account for details.

Starting Jupyter

To start a Jupyter Job on the NAF JupyterHub, login with your DESY credentials and select Job options like your primary group and whether you wish to use a GPU slot.

You then need to wait until HTCondor starts your jupyter job and the notebook server is spawned.
This takesabout 20, so don't be too impatient.

Once your server starts you see your AFS $HOME and can start a new Notebook by clicking on the "New" → "Notebook" → "Python 3"



Converting datasets from h5/hdf5 to ROOT II

Using Pyroot with Jupyter on NAF

- Start at your own afs directory, create new notebook

In [ ]:

```python
import ROOT
import h5py

from ROOT import TFile, TTree, gROOT
from array import array
```

In [ ]:

```python
# specify file
file = h5py.File('/nfs/dust/luxe/group/MCProduction/Signal/ptarmigan-v0.7-preview/brem_laser/phase0/5.00_particles.h5', 'r')
```

In [ ]:

```python
# Inspect file
# hdf5 has two main structures:
# 1) groups
# 2) datasets, e.g arrays, single values also possible

def hierarchy(d):
    for item in d:
        if '0 members' in str(d[item]):
            print(d[item].name, ['empty group'])
        if isinstance(d[item], h5py.Group):
            hierarchy(d[item])
        else:
            print(d[item].name, ['dataset'])

hierarchy(file)
```

Converting datasets from h5/hdf5 to ROOT III

Using Pyroot with Jupyter on NAF

```
/build/branch ['dataset']
/build/commit-hash ['dataset']
/build/features ['dataset']
/build/version ['dataset']
/config/beam/bremsstrahlung_source ['dataset']
/config/beam/collision_angle ['dataset']
/config/beam/gamma ['dataset']
/config/beam/gamma_min ['dataset']
/config/beam/length ['dataset']
/config/beam/longitudinal_distribution_is_normal ['dataset']
/config/beam/n ['dataset']
/config/beam/radius ['dataset']
/config/beam/rms_divergence ['dataset']
/config/beam/sigma ['dataset']
/config/beam/transverse_distribution_is_normal ['dataset']
/config/control/bandwidth_correction ['dataset']
/config/control/dt_multiplier ['dataset']
/config/control/increase_pair_rate_by ['dataset']
/config/control/lcfa ['dataset']
/config/control/rng_seed ['dataset']
/config/control/select_multiplicity ['dataset']
/config/input-file ['dataset']
/config/laser/a0 ['dataset']
/config/laser/chirp_b ['dataset']
/config/laser/focusing ['dataset']
/config/laser/fwhm_duration ['dataset']
/config/laser/polarization ['dataset']
/config/laser/waist ['dataset']
/config/laser/wavelength ['dataset']
/config/mpi-tasks ['dataset']
/config/output/beam_defines_positive_z ['dataset']
/config/output/discard_background_e ['dataset']
/config/output/laser_defines_positive_z ['dataset']
/config/output/min_energy ['dataset']
/final-state/electron/momentum ['dataset']
/final-state/electron/n_gamma ['dataset']
/final-state/electron/position ['dataset']
/final-state/electron/weight ['dataset']
/final-state/photon/a0_at_creation ['dataset']
/final-state/photon/momentum ['dataset']
/final-state/photon/n_pos ['dataset']
/final-state/photon/position ['dataset']
/final-state/photon/weight ['dataset']
/final-state/photon/xi ['dataset']
/final-state/positron/momentum ['dataset']
/final-state/positron/n_gamma ['dataset']
/final-state/positron/position ['dataset']
/final-state/positron/weight ['dataset']
```

- access via address :

address

event number

```
1 print(file['/final-state/photon/momentum'][0])
```

```
[2.97486279e+03 1.23197152e-02 2.74067540e-04 2.97486279e+03]
```

- addresses are keys of their respective supergroups

```
1 print(file['/build'].keys())
```

```
<KeysViewHDF5 ['branch', 'commit-hash', 'features', 'version']>
```

Converting datasets from h5/hdf5 to ROOT IV

Using Pyroot with Jupyter on NAF

```
OUTPUT_FILE_NAME = "hp5_to_ROOT.root"

f = ROOT.TFile(OUTPUT_FILE_NAME, "RECREATE")
tree = ROOT.TTree("MC_Simulated_Example", "Convert_Example")

# build
tree.Branch('build', build, 'branch[50]/C:commit_hash[50]/C:features[50]/C:version[50]/C')

# config/beam
tree.Branch('bremsstrahlung_source', bremsstrahlung_source, 'bremsstrahlung_source/O')
tree.Branch('collision_angle', collision_angle, 'collision_angle/D')
tree.Branch('gamma', gamma, 'gamma/D')
tree.Branch('gamma_min', gamma_min, 'gamma_min/D')
tree.Branch('length', length, 'length/D')
tree.Branch('longitudinal_distribution_is_normal', longitudinal_distribution_is_normal, 'longitudinal_distribution_is_normal
tree.Branch('n', n, 'n/I')
tree.Branch('radius', radius, 'radius/D')
```

```
# Filling the Tree

# build
build.branch = file['/build/branch'][()]
build.commit_hash = file['/build/commit-hash'][()]
build.features = file['/build/features'][()]
build.version = file['/build/version'][()]

# config/beam
bremsstrahlung_source[0] = int(file['/config/beam/bremsstrahlung_source'][()])
collision_angle[0] = file['/config/beam/collision_angle'][()]
gamma[0] = file['/config/beam/gamma'][()]
gamma_min[0] = file['/config/beam/gamma_min'][()]
length[0] =  file['/config/beam/length'][()]
longitudinal_distribution_is_normal[0] = int(file['/config/beam/longitudinal_distribution_is_normal'][()])
n[0] = int(file['/config/beam/n'][()])
radius[0] = file['/config/beam/radius'][()]
```

Single value in hdf5 file
has to be accessed this way

Converting datasets from h5/hdf5 to ROOT V

Using Pyroot with Jupyter on NAF

```
tree.Show(1)
```

```
======> EVENT:1
 branch           = devel
 commit_hash      = f2598613dd16a533fa4152e6650f6e370654ceb5
 features         = fits-output,hdf5-output,with-mpi
 version          = 0.6.2
 bremsstrahlung_source = 1
 collision_angle = -0.300197
 gamma            = 32289.7
 gamma_min        = 3913.9
 length           = 2.4e-05
 longitudinal_distribution_is_normal = 1
 n                = 500000
 radius           = 8.4628e-06
 rms_divergence   = 8.672e-06
 sigma            = 0
 transverse_distribution_is_normal = 0
 bandwidth_correction = 0
 dt_multiplier    = 0.1
 increase_pair_rate_by = 58.5788
 lcfa             = 0
 rng_seed         = 0
 select_multiplicity = 0
 a0               = 5
 chirp_b          = 0
 focusing         = 1
 fwhm_duration    = 2.5e-14
 polarization     = 1
 waist            = 4.2314e-06
 wavelength       = -1698910392
 mpi_tasks        = 28
 beam_defines_positive_z = 1
 discard_background_e = 0
 laser_defines_positive_z = 0
 min_energy       = 0
 e_momentum       = 5982.35,
                   0.853699, 0.0893968, 5982.35
 e_n_gamma        = 1
 e_position       = 3.92521e-05,
                   -4.72542e-06, -6.01045e-07, -1.60652e-05
 e_weight         = 0.000314957
 p_momentum       = 6824.93,
                   -1.17599, -0.0570599, 6824.93
 p_n_gamma        = 1
 p_position       = 3.92521e-05,
                   -4.7287e-06, -6.01263e-07, -1.60652e-05
```

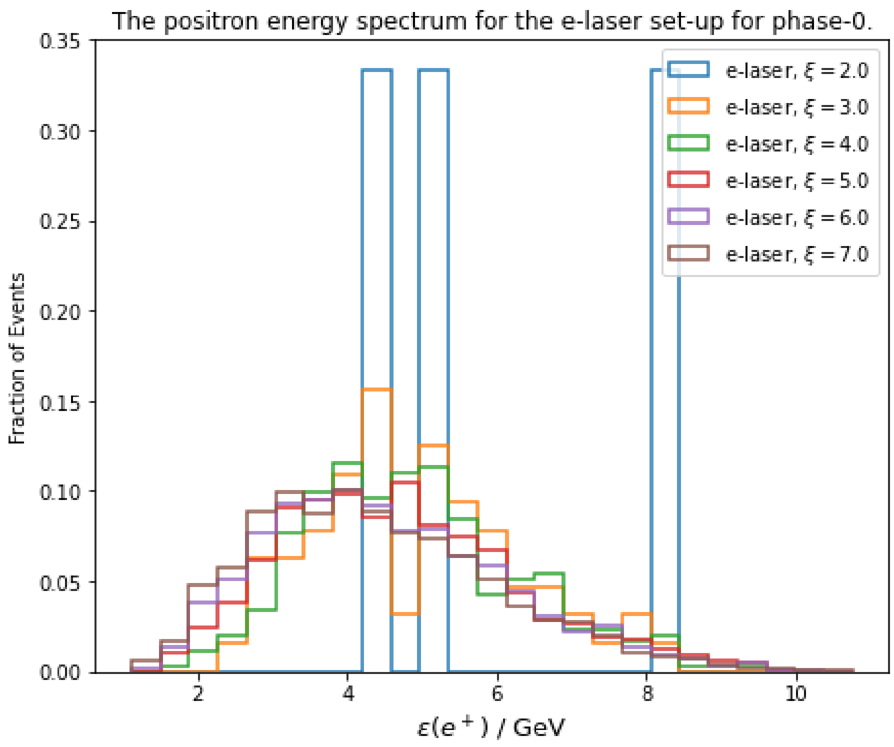
```
# save config file separately
text = file['/config/input-file'][()]
outfile = open('config_file', 'w')
text_separated = text.split('\n')
for line in text_separated:
    print(line)
    outfile.write(line)
outfile.close()
```

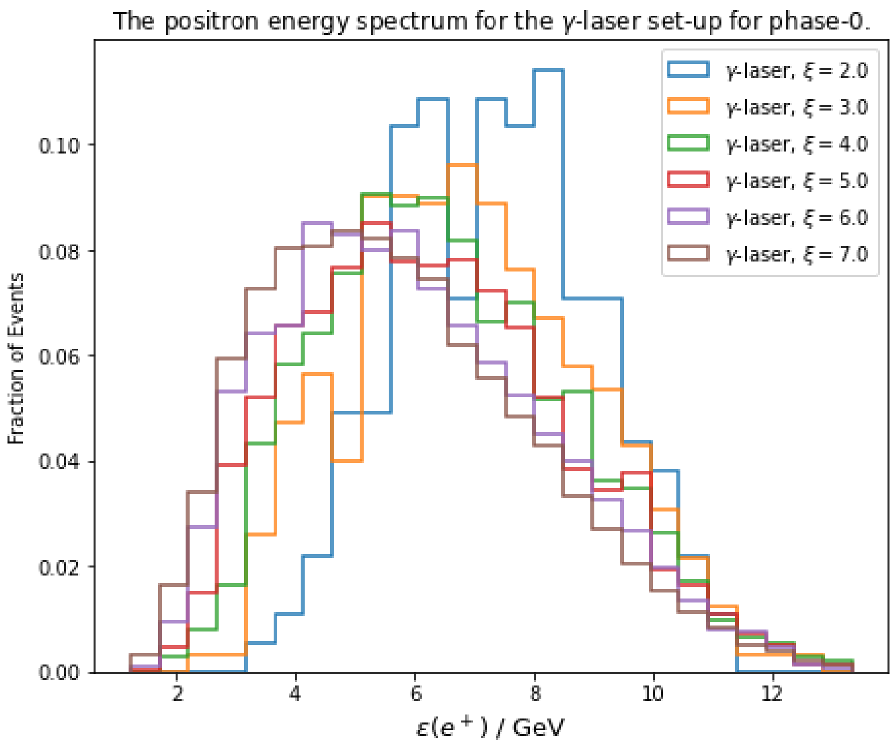
```
---
control:
  dt_multiplier: 0.1
  increase_pair_rate_by: auto

laser:
  a0: 5.00
  wavelength: 0.8 * micro
  fwhm_duration: 25.0 * femto
  waist: waist

beam:
  ne: 500000
  species: photon
  charge: 41.0 * pi * (2.0 * waist / micro)^2 * e # 41 photons / um^2 in focal plane
  gamma: 16.5 * GeV / (m * c^2)
  gamma_min: 2.0 * GeV / (m * c^2)
  bremsstrahlung_source: true
  radius: [2.0 * waist, uniformly_distributed]
  collision_angle: -17.2 * degree
  length: 24.0 * micro
  rms_divergence: 8.672 * micro

output:
  ident: 5.00
  dump_all_particles: hdf5
  coordinate_system: beam
  photon: [energy]
```

Plotting data with Python I

Example plot for photon

- basic plots from hd5 file to plot in few lines
- address handles to make it look nicer, latex syntax for strings supported

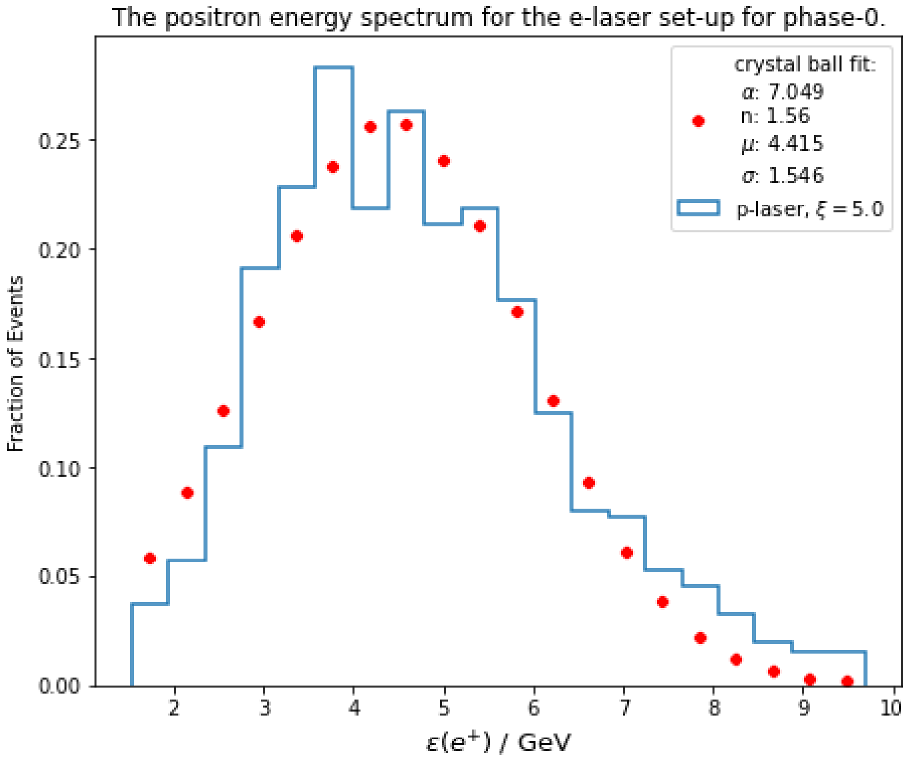


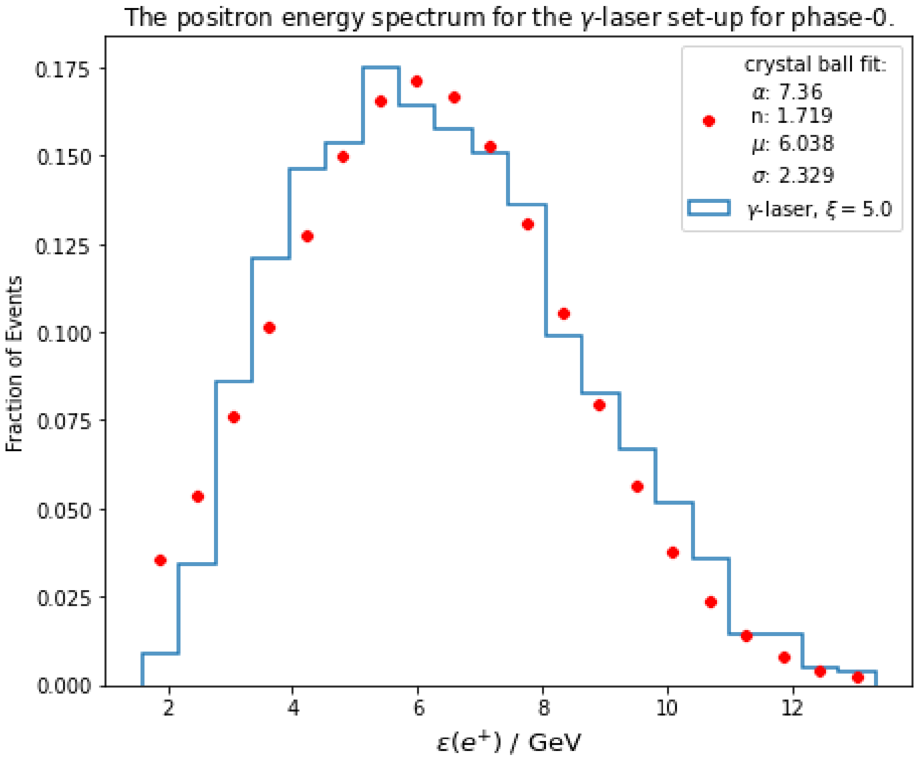


Plotting data with Python II

Fitting is alos very easy – if you import the correct packages :)

- easy to fit even quite complicated functions





Arguments for hdf5 /h5

viable solution in the future (?)

- hdf5 / h5 are very easy to to store, access, plot via python
- python is still on the rise, could be in favour of opting for hdf5 / h5
- structure is of a nested dictionary like in Python, varying size and numbers of subgroups/datasets are implemented / read easily
- good readability of the code (if done correctly)
- viable for big datasets

Converting is bound to this hdf5 structure, but can be adapted for other ones easily

---

- copy nb to your afs, start Jupyter on NAF and load it
- Jupyter Notebook example for converting hdf5 to ROOT:
  - /nfs/dust/luxe/user/spatarod/jupyter-notebooks/h5_to_ROOT.ipynb
  - /nfs/dust/luxe/group/MCProduction/Signal/ptarmigan-v0.7-preview/brem_laser/phase0/5.00_particles.h5

- Jupyter Notebook example for plotting hdf5 content:
  - /nfs/dust/luxe/user/spatarod/jupyter-notebooks/plot_hdf5.ipyn

Sources

https://raw.githubusercontent.com/NEONScience/NEON-Data-Skills/dev-aten/graphics/HDF5-general/hdf5_structure4.jpg

https://root.cern.ch/root/htmldoc/guides/users-guide/Trees.html

https://support.hdfgroup.org/HDF5/doc/H5.intro.html#Intro-OAttributes

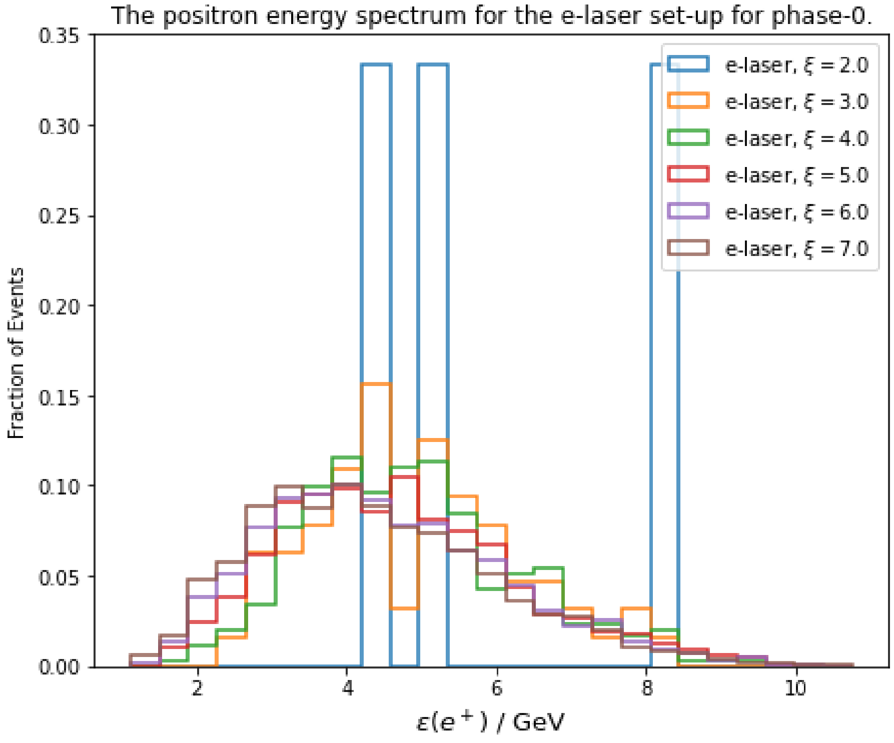
Contact

DESY. Deutsches Elektronen-Synchrotron

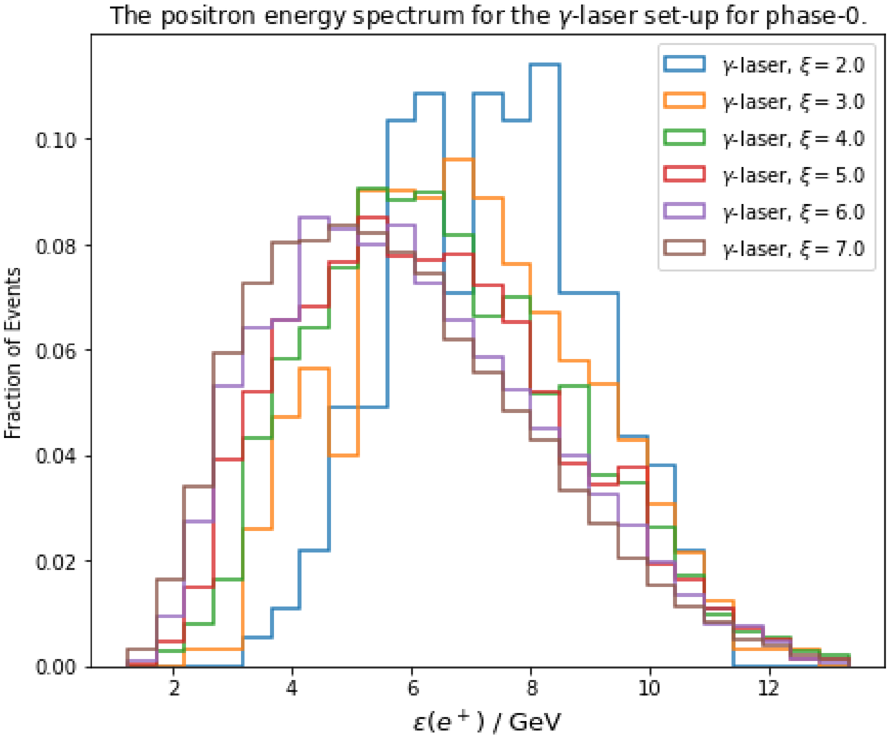
www.desy.de

- David Spataro
- FTX
- david.spataro@desy.de

PLOTS I

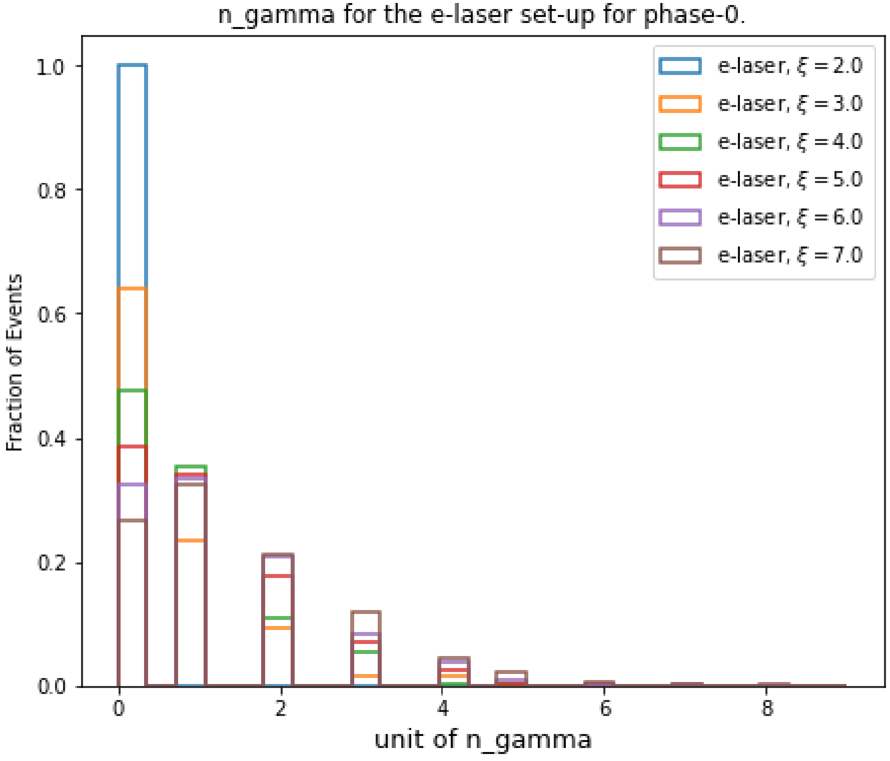


The positron energy spectrum for the e-laser set-up for phase-0.
Fraction of Events
$\varepsilon(e^+)$ / GeV
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
2
4
6
8
10
e-laser, $\xi = 2.0$
e-laser, $\xi = 3.0$
e-laser, $\xi = 4.0$
e-laser, $\xi = 5.0$
e-laser, $\xi = 6.0$
e-laser, $\xi = 7.0$

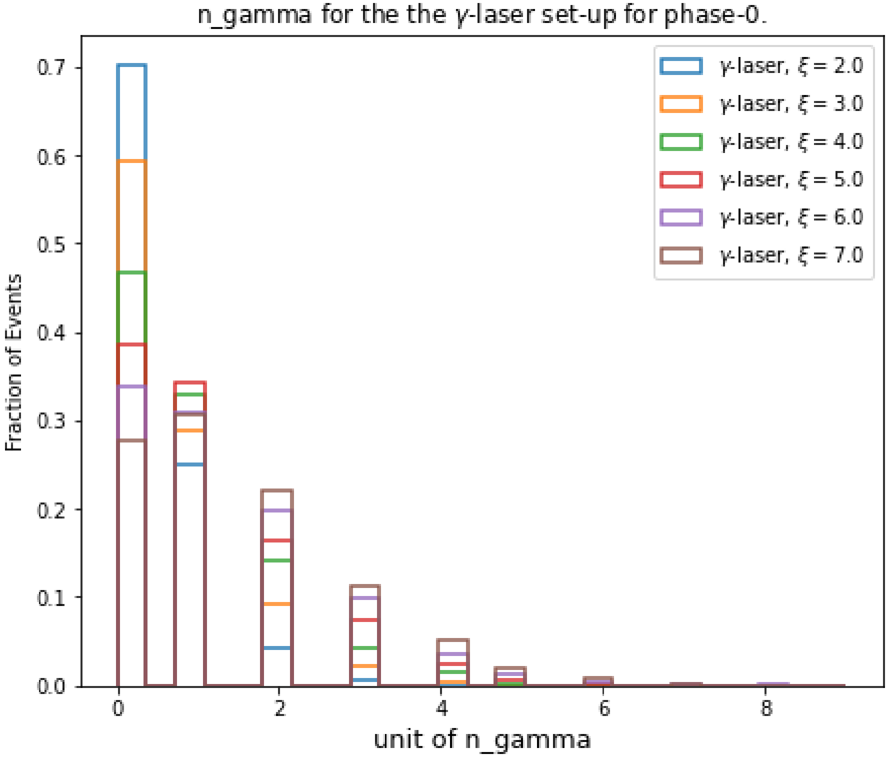


The positron energy spectrum for the γ-laser set-up for phase-0.
Fraction of Events
$\varepsilon(e^+)$ / GeV
0.00
0.02
0.04
0.06
0.08
0.10
2
4
6
8
10
12
γ-laser, $\xi = 2.0$
γ-laser, $\xi = 3.0$
γ-laser, $\xi = 4.0$
γ-laser, $\xi = 5.0$
γ-laser, $\xi = 6.0$
γ-laser, $\xi = 7.0$

PLOTS II

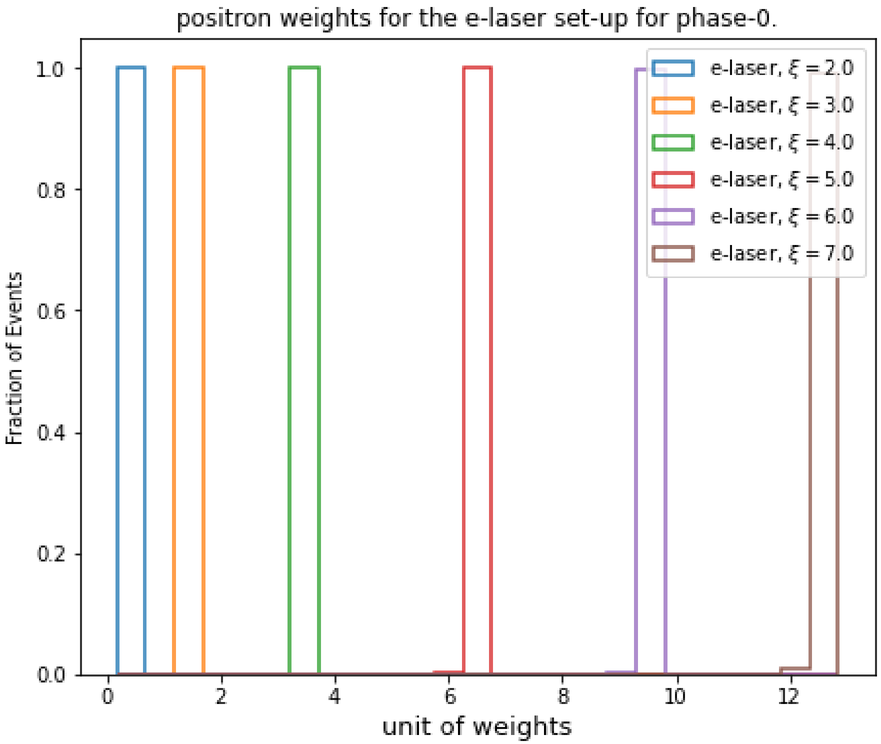


n_gamma for the e-laser set-up for phase-0.
Fraction of Events
unit of n_gamma
e-laser, ξ = 2.0
e-laser, ξ = 3.0
e-laser, ξ = 4.0
e-laser, ξ = 5.0
e-laser, ξ = 6.0
e-laser, ξ = 7.0

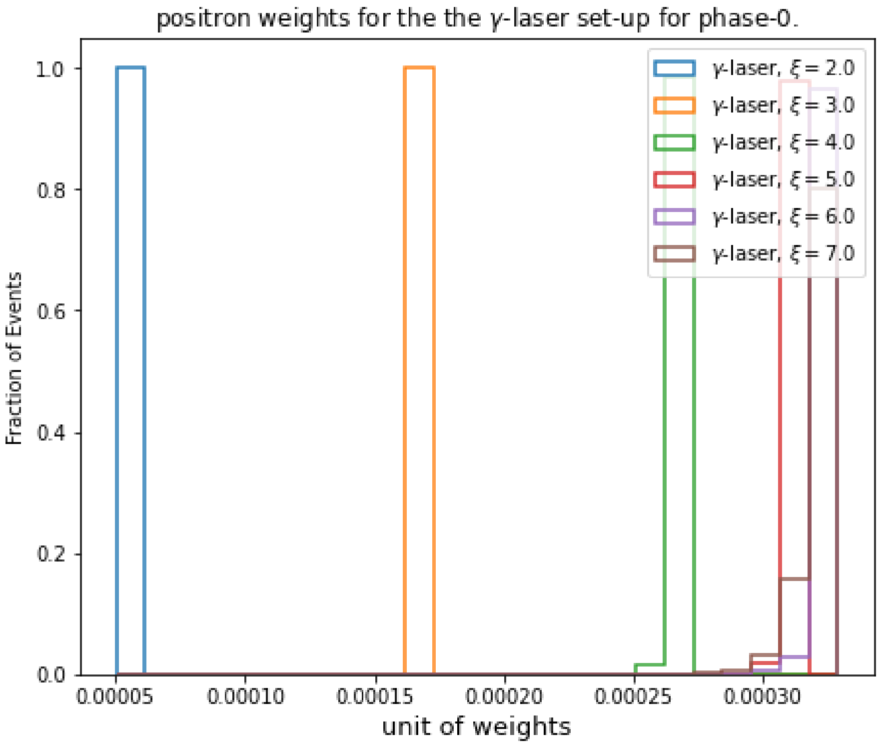


n_gamma for the the γ-laser set-up for phase-0.
Fraction of Events
unit of n_gamma
γ-laser, ξ = 2.0
γ-laser, ξ = 3.0
γ-laser, ξ = 4.0
γ-laser, ξ = 5.0
γ-laser, ξ = 6.0
γ-laser, ξ = 7.0

PLOTS III

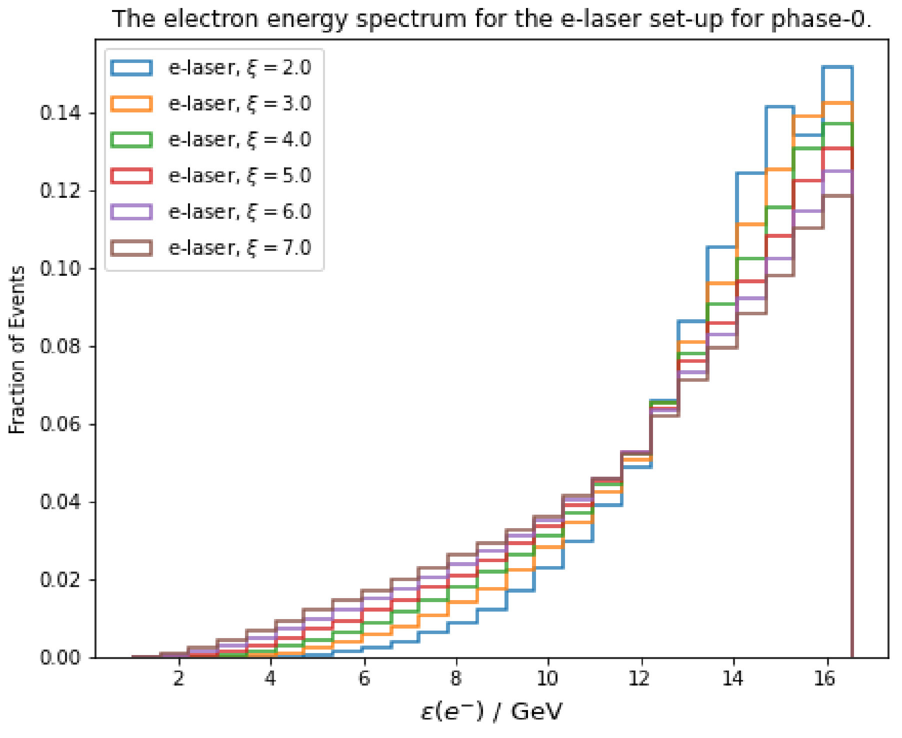


positron weights for the e-laser set-up for phase-0.
Fraction of Events
unit of weights
e-laser, $\xi = 2.0$
e-laser, $\xi = 3.0$
e-laser, $\xi = 4.0$
e-laser, $\xi = 5.0$
e-laser, $\xi = 6.0$
e-laser, $\xi = 7.0$

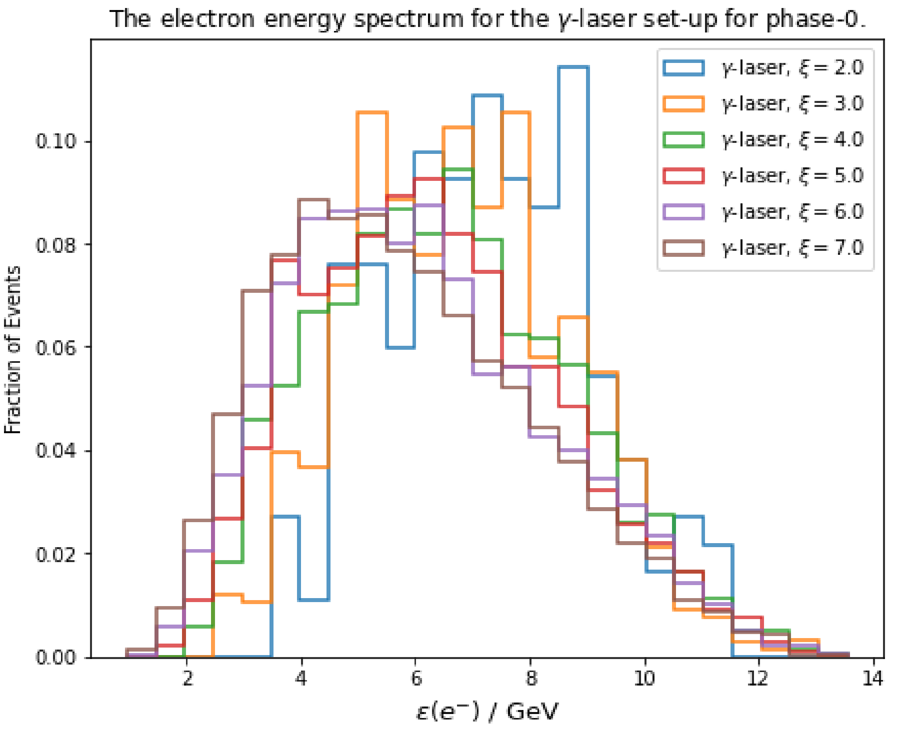


positron weights for the the γ-laser set-up for phase-0.
Fraction of Events
unit of weights
γ-laser, $\xi = 2.0$
γ-laser, $\xi = 3.0$
γ-laser, $\xi = 4.0$
γ-laser, $\xi = 5.0$
γ-laser, $\xi = 6.0$
γ-laser, $\xi = 7.0$

PLOTS IV

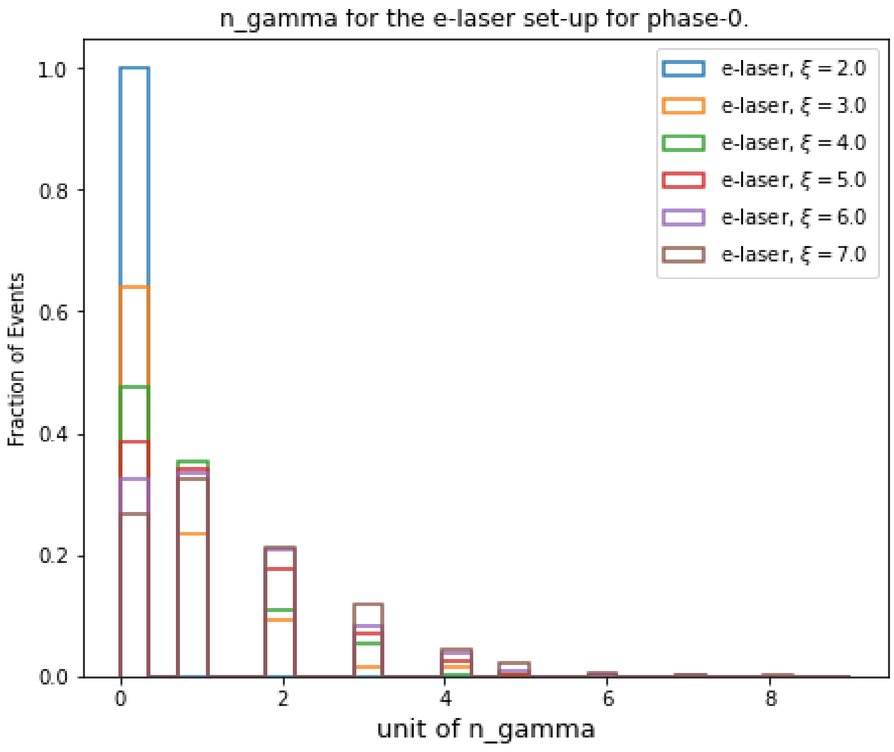


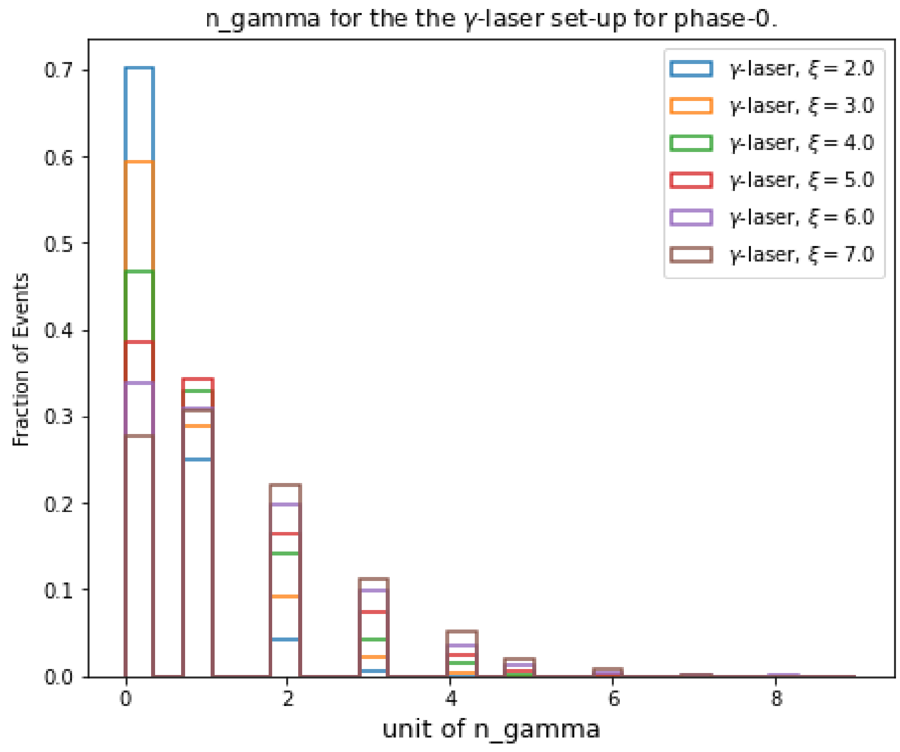
The electron energy spectrum for the e-laser set-up for phase-0.
e-laser, $\xi = 2.0$
e-laser, $\xi = 3.0$
e-laser, $\xi = 4.0$
e-laser, $\xi = 5.0$
e-laser, $\xi = 6.0$
e-laser, $\xi = 7.0$
Fraction of Events
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
2
4
6
8
10
12
14
16
$\varepsilon(e^-)$ / GeV



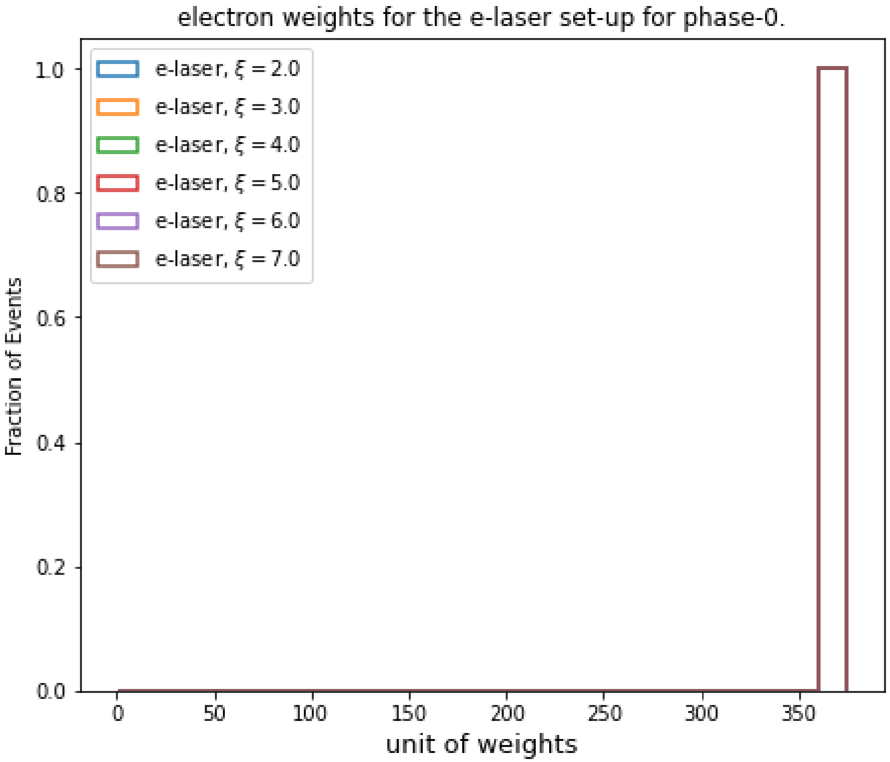
The electron energy spectrum for the γ-laser set-up for phase-0.
γ-laser, $\xi = 2.0$
γ-laser, $\xi = 3.0$
γ-laser, $\xi = 4.0$
γ-laser, $\xi = 5.0$
γ-laser, $\xi = 6.0$
γ-laser, $\xi = 7.0$
Fraction of Events
0.00
0.02
0.04
0.06
0.08
0.10
2
4
6
8
10
12
14
$\varepsilon(e^-)$ / GeV

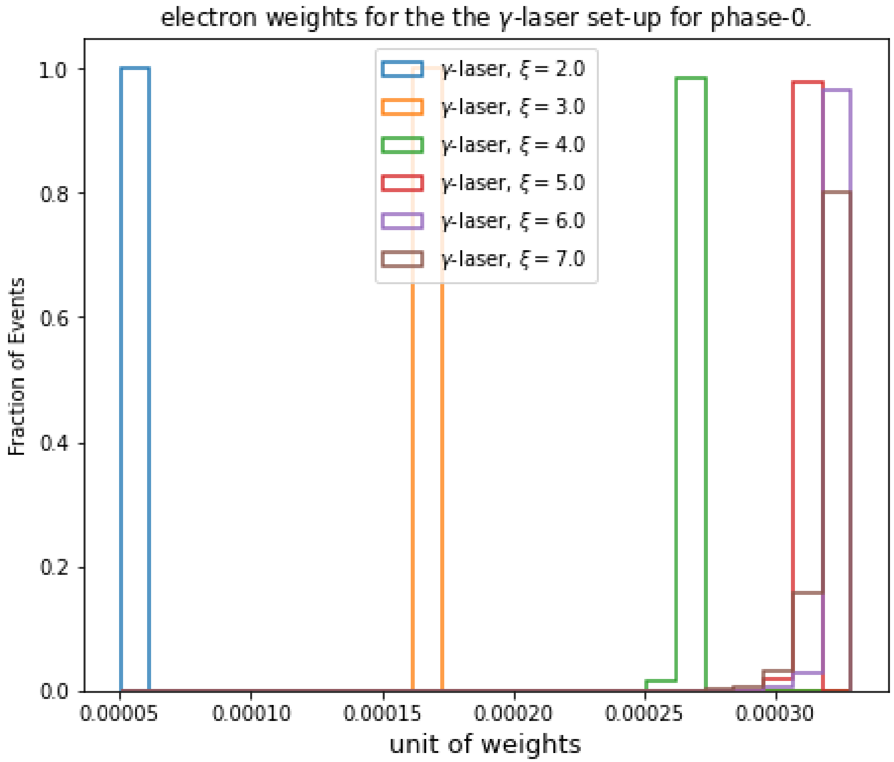
PLOTS V


n_gamma for the e-laser set-up for phase-0.
e-laser, $\xi = 2.0$
e-laser, $\xi = 3.0$
e-laser, $\xi = 4.0$
e-laser, $\xi = 5.0$
e-laser, $\xi = 6.0$
e-laser, $\xi = 7.0$
Fraction of Events
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
unit of n_gamma

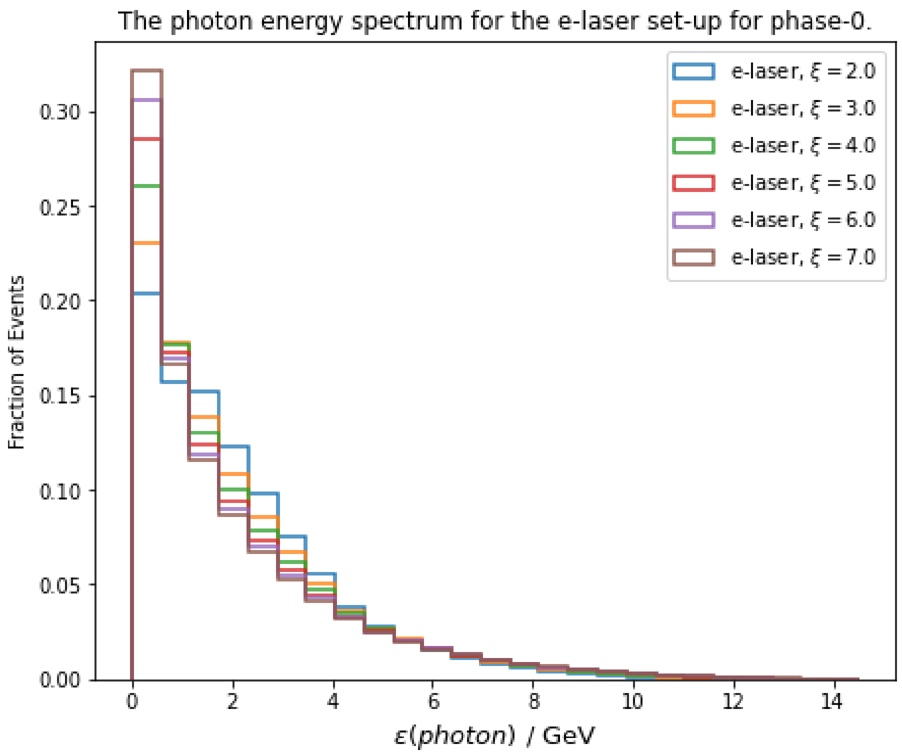

n_gamma for the the γ-laser set-up for phase-0.
γ-laser, $\xi = 2.0$
γ-laser, $\xi = 3.0$
γ-laser, $\xi = 4.0$
γ-laser, $\xi = 5.0$
γ-laser, $\xi = 6.0$
γ-laser, $\xi = 7.0$
Fraction of Events
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2
4
6
8
unit of n_gamma

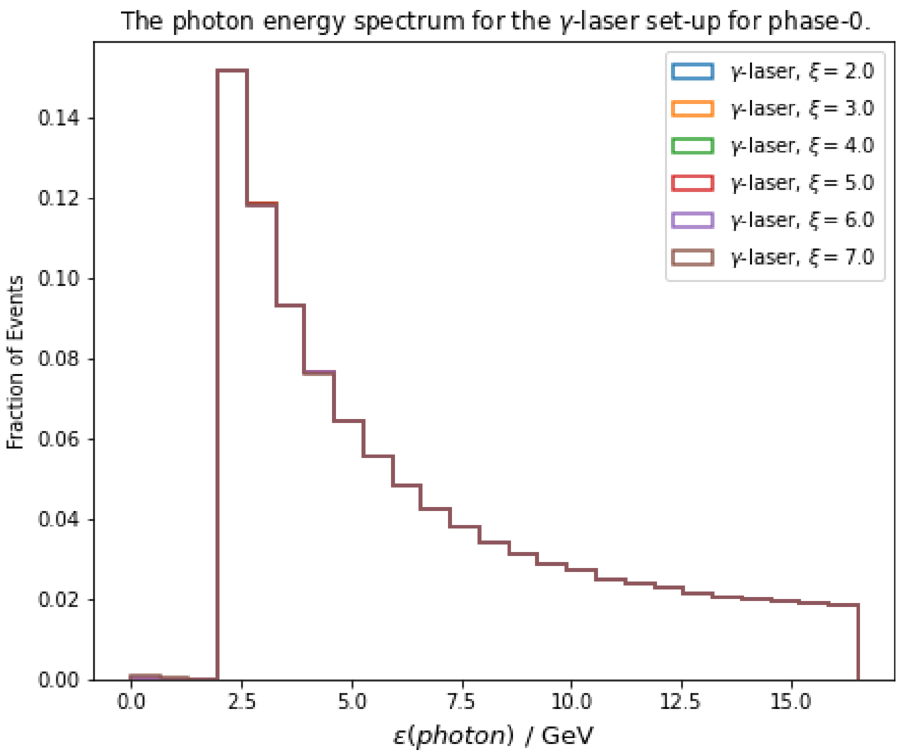
PLOTS VI


electron weights for the e-laser set-up for phase-0.
e-laser, $\xi = 2.0$
e-laser, $\xi = 3.0$
e-laser, $\xi = 4.0$
e-laser, $\xi = 5.0$
e-laser, $\xi = 6.0$
e-laser, $\xi = 7.0$
Fraction of Events
unit of weights
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
250
300
350

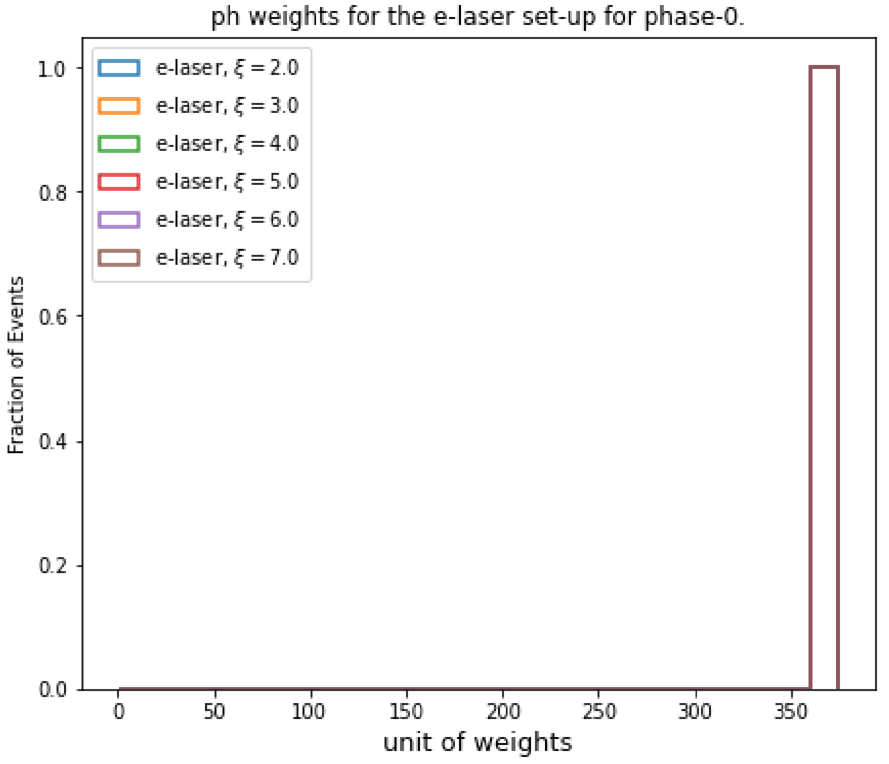

electron weights for the the γ-laser set-up for phase-0.
γ-laser, $\xi = 2.0$
γ-laser, $\xi = 3.0$
γ-laser, $\xi = 4.0$
γ-laser, $\xi = 5.0$
γ-laser, $\xi = 6.0$
γ-laser, $\xi = 7.0$
Fraction of Events
unit of weights
0.0
0.2
0.4
0.6
0.8
1.0
0.00005
0.00010
0.00015
0.00020
0.00025
0.00030

PLOTS VII

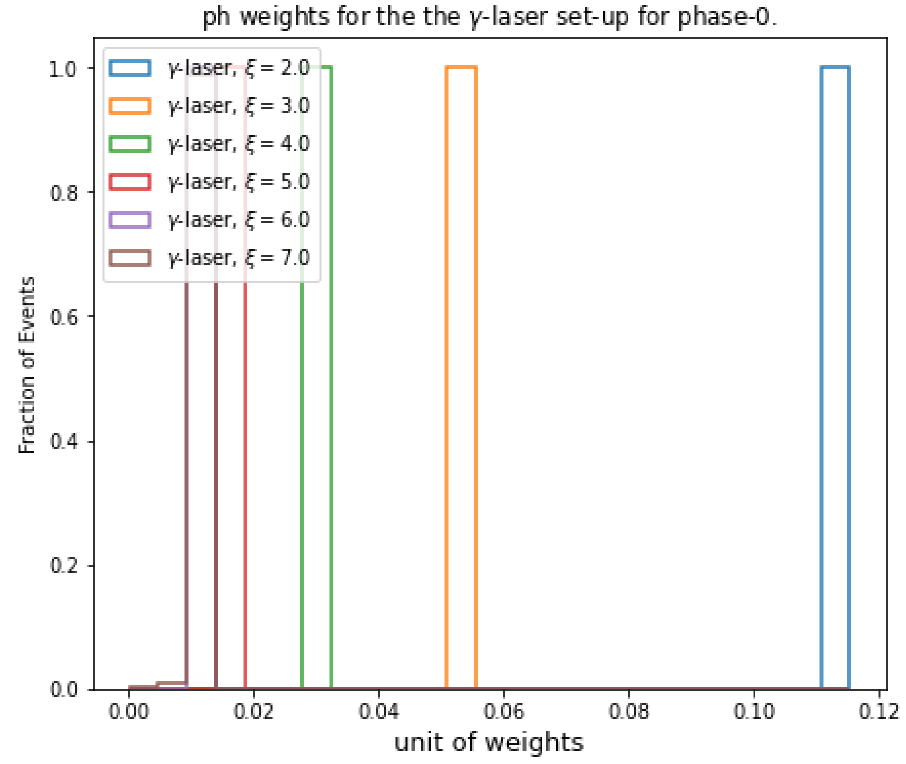

The photon energy spectrum for the e-laser set-up for phase-0.
Fraction of Events
ε(photon) / GeV
e-laser, ξ = 2.0
e-laser, ξ = 3.0
e-laser, ξ = 4.0
e-laser, ξ = 5.0
e-laser, ξ = 6.0
e-laser, ξ = 7.0


The photon energy spectrum for the γ-laser set-up for phase-0.
Fraction of Events
ε(photon) / GeV
γ-laser, ξ = 2.0
γ-laser, ξ = 3.0
γ-laser, ξ = 4.0
γ-laser, ξ = 5.0
γ-laser, ξ = 6.0
γ-laser, ξ = 7.0

PLOTS VIII

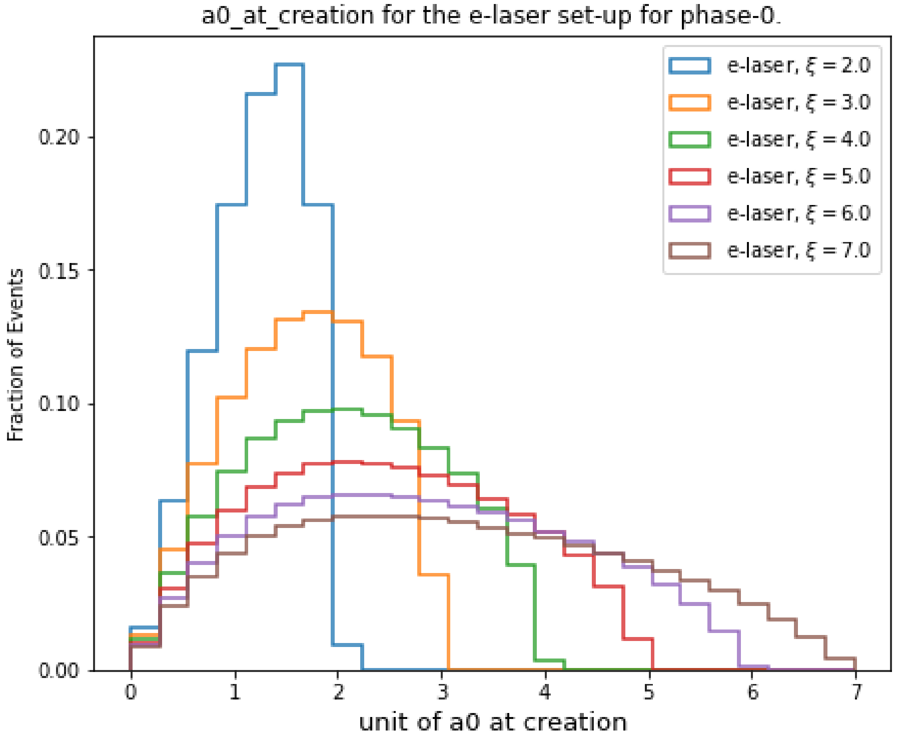


ph weights for the e-laser set-up for phase-0.
e-laser, $\xi = 2.0$
e-laser, $\xi = 3.0$
e-laser, $\xi = 4.0$
e-laser, $\xi = 5.0$
e-laser, $\xi = 6.0$
e-laser, $\xi = 7.0$
Fraction of Events
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
250
300
350
unit of weights

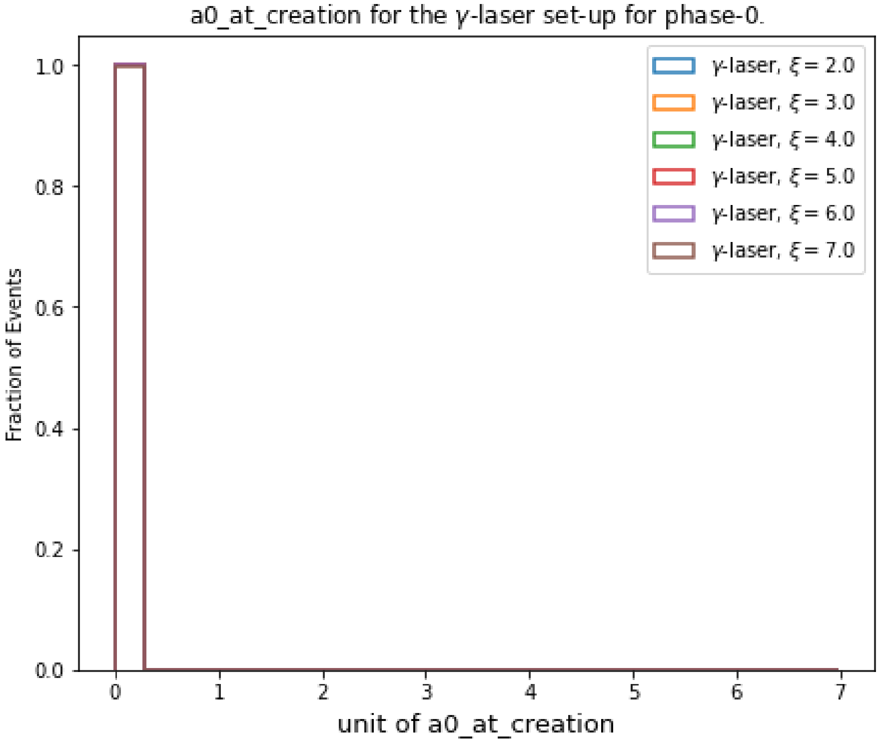


ph weights for the the γ-laser set-up for phase-0.
γ-laser, $\xi = 2.0$
γ-laser, $\xi = 3.0$
γ-laser, $\xi = 4.0$
γ-laser, $\xi = 5.0$
γ-laser, $\xi = 6.0$
γ-laser, $\xi = 7.0$
Fraction of Events
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
unit of weights

PLOTS IX

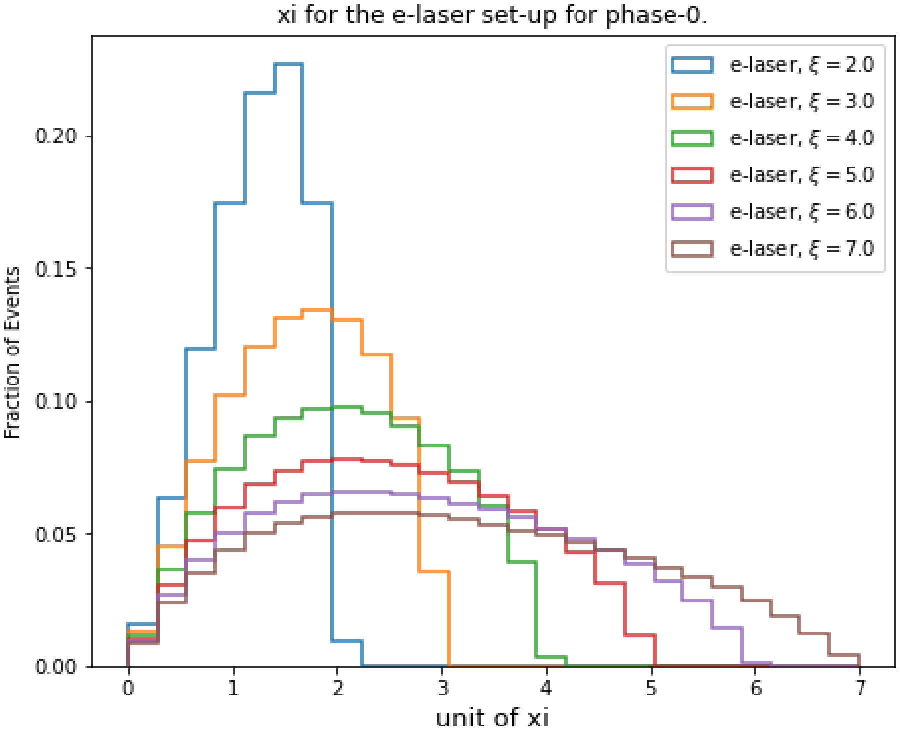


a0_at_creation for the e-laser set-up for phase-0.
e-laser, ξ = 2.0
e-laser, ξ = 3.0
e-laser, ξ = 4.0
e-laser, ξ = 5.0
e-laser, ξ = 6.0
e-laser, ξ = 7.0
Fraction of Events
0.00
0.05
0.10
0.15
0.20
0 1 2 3 4 5 6 7
unit of a0 at creation

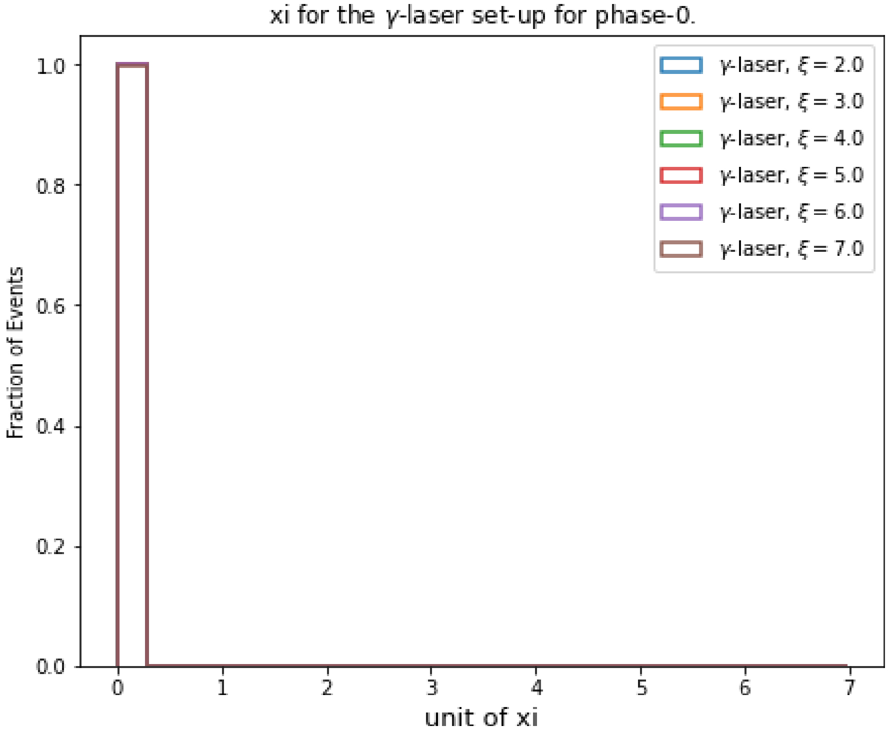


a0_at_creation for the γ-laser set-up for phase-0.
γ-laser, ξ = 2.0
γ-laser, ξ = 3.0
γ-laser, ξ = 4.0
γ-laser, ξ = 5.0
γ-laser, ξ = 6.0
γ-laser, ξ = 7.0
Fraction of Events
0.0
0.2
0.4
0.6
0.8
1.0
0 1 2 3 4 5 6 7
unit of a0_at_creation

PLOTS X

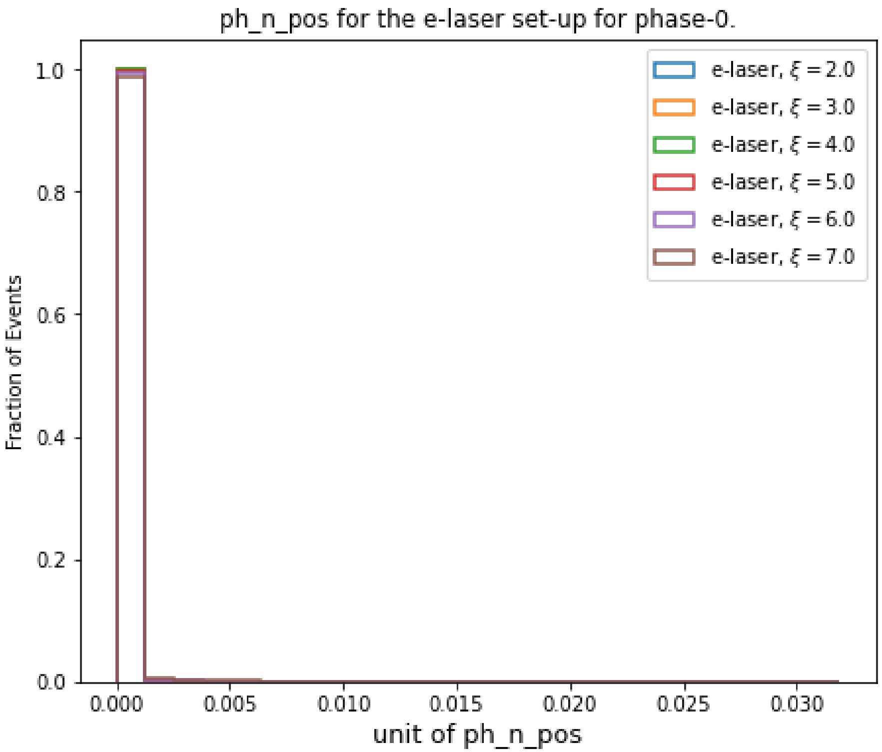


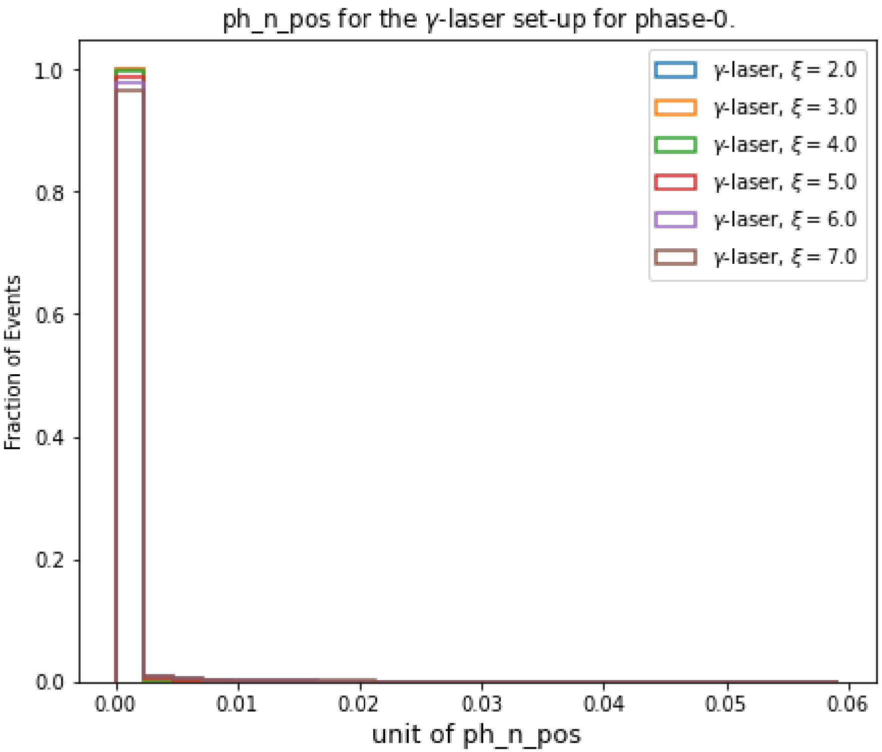
xi for the e-laser set-up for phase-0.
e-laser, $\xi = 2.0$
e-laser, $\xi = 3.0$
e-laser, $\xi = 4.0$
e-laser, $\xi = 5.0$
e-laser, $\xi = 6.0$
e-laser, $\xi = 7.0$
Fraction of Events
0.00
0.05
0.10
0.15
0.20
0
1
2
3
4
5
6
7
unit of xi



xi for the γ-laser set-up for phase-0.
γ-laser, $\xi = 2.0$
γ-laser, $\xi = 3.0$
γ-laser, $\xi = 4.0$
γ-laser, $\xi = 5.0$
γ-laser, $\xi = 6.0$
γ-laser, $\xi = 7.0$
Fraction of Events
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6
7
unit of xi

PLOTS XI


ph_n_pos for the e-laser set-up for phase-0.
e-laser, $\xi = 2.0$
e-laser, $\xi = 3.0$
e-laser, $\xi = 4.0$
e-laser, $\xi = 5.0$
e-laser, $\xi = 6.0$
e-laser, $\xi = 7.0$
Fraction of Events
1.0
0.8
0.6
0.4
0.2
0.0
0.000
0.005
0.010
0.015
0.020
0.025
0.030
unit of ph_n_pos


ph_n_pos for the γ-laser set-up for phase-0.
γ-laser, $\xi = 2.0$
γ-laser, $\xi = 3.0$
γ-laser, $\xi = 4.0$
γ-laser, $\xi = 5.0$
γ-laser, $\xi = 6.0$
γ-laser, $\xi = 7.0$
Fraction of Events
1.0
0.8
0.6
0.4
0.2
0.0
0.00
0.01
0.02
0.03
0.04
0.05
0.06
unit of ph_n_pos