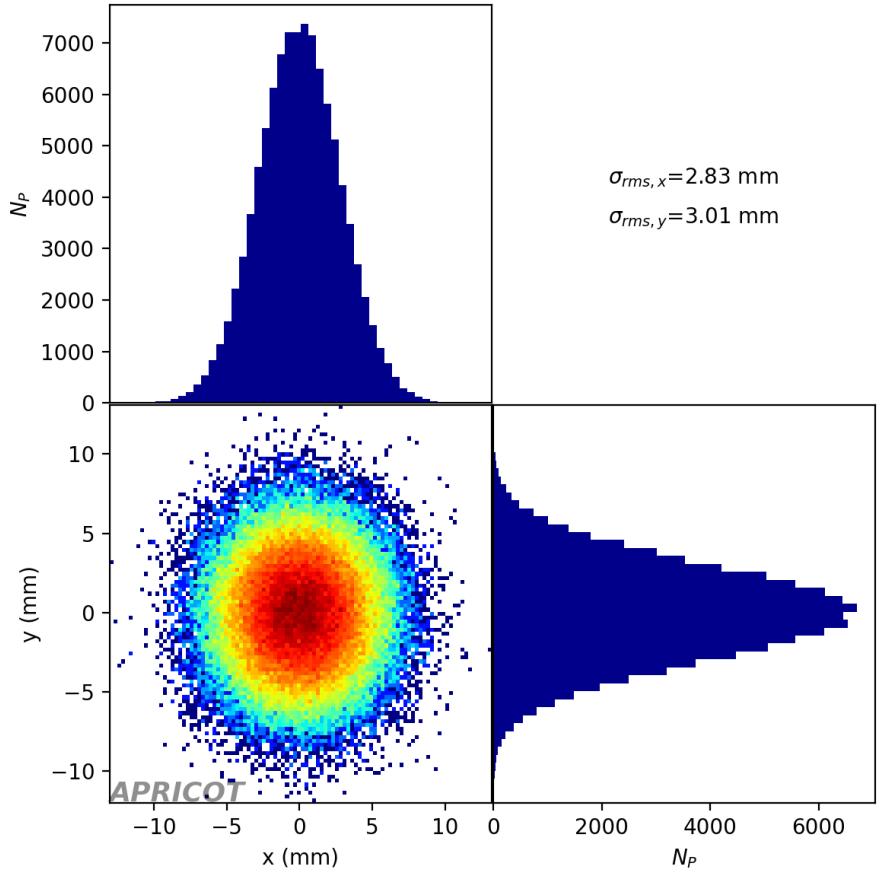
APRICOT

github.com/acanbay/apricot

Ali Can Canbay (AÜ)

Python3 için hazırlanmış **parçacık iz takip (particle tracking)** modülü.

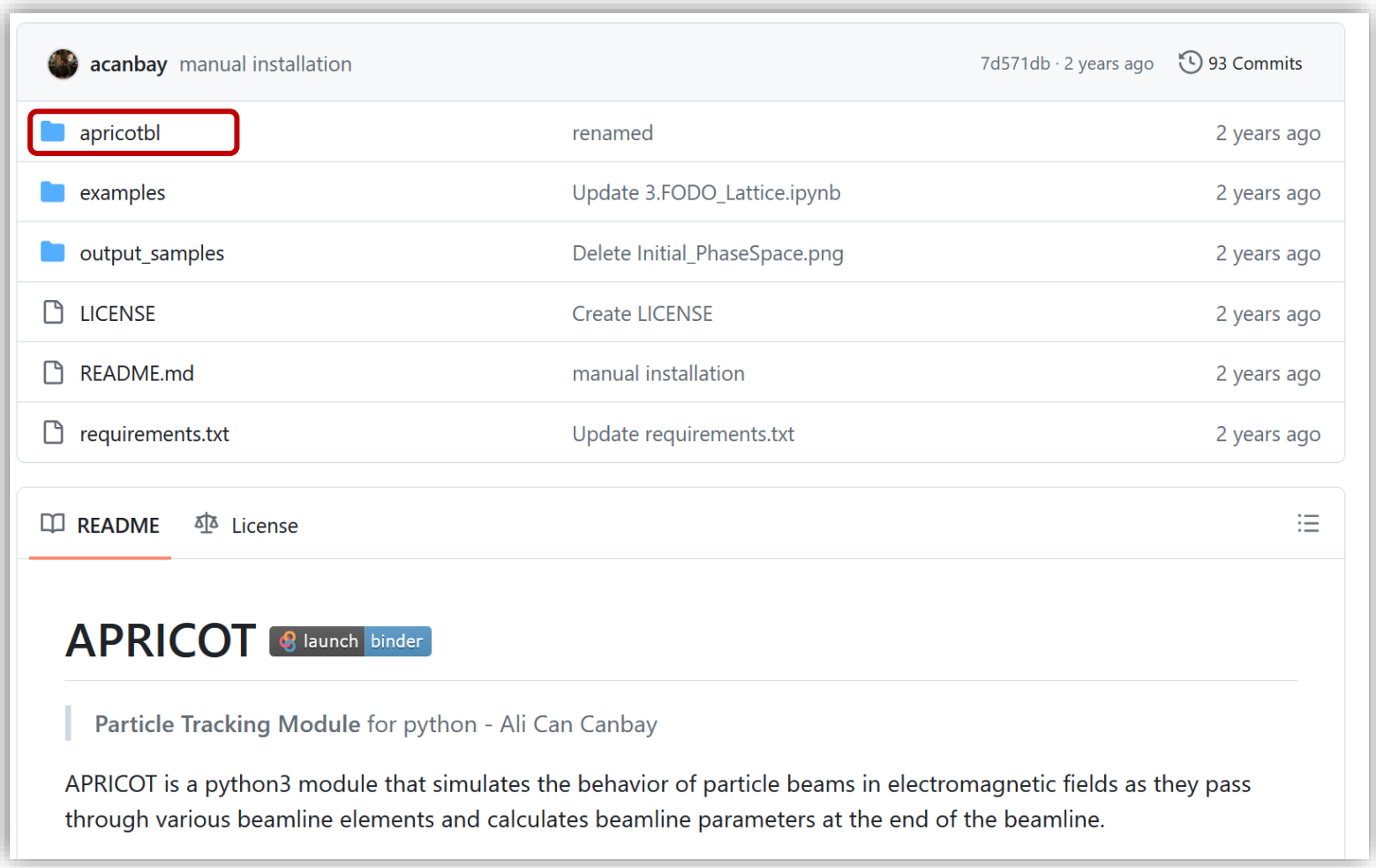
- Demet üretimi
  - Rastgele üreteç
  - Elektron tabancası
- Demet hattı elemanları
  - Sürüklenme tüpü
  - Kuadropol mıknatıs
  - Dipol mıknatıs
  - Solenoid
- FODO örgüleri



Geliştirme aşaması devam ediyor (Eğitim için kullanılabilir).

Ana dosyalar

https://github.com/acanbay/apricot

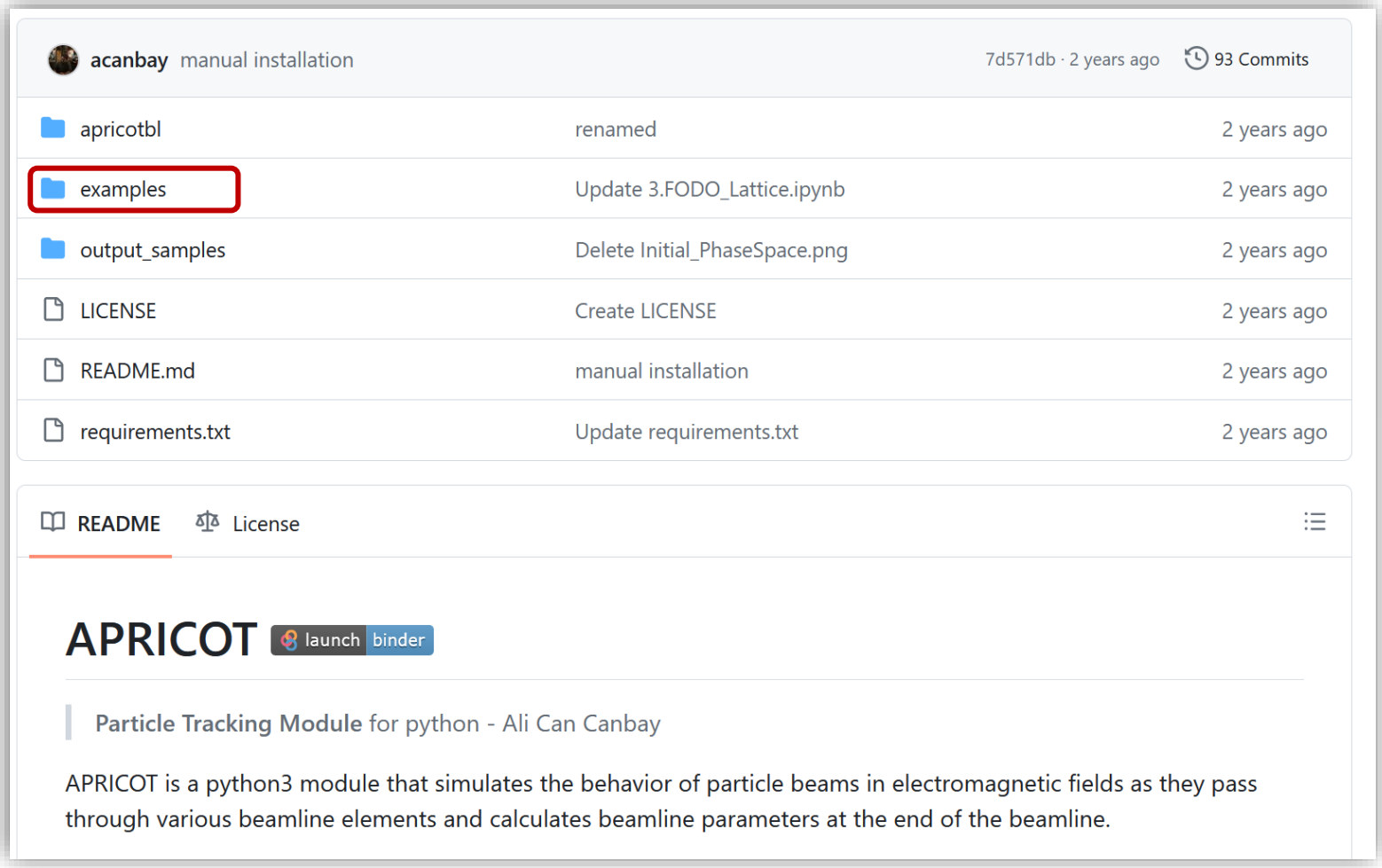


Modül dosyalarını içerir

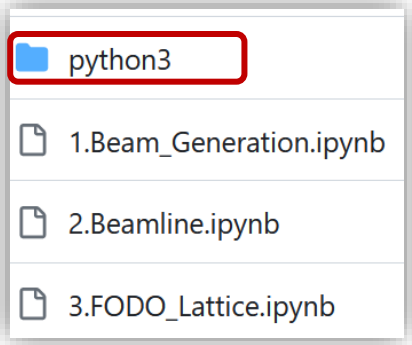
- Beamline.py
- BeamlineComponents.py
- Functions.py
- Graphs.py
- Gun.py
- Matrices.py
- Outputs.py
- Statistics.py

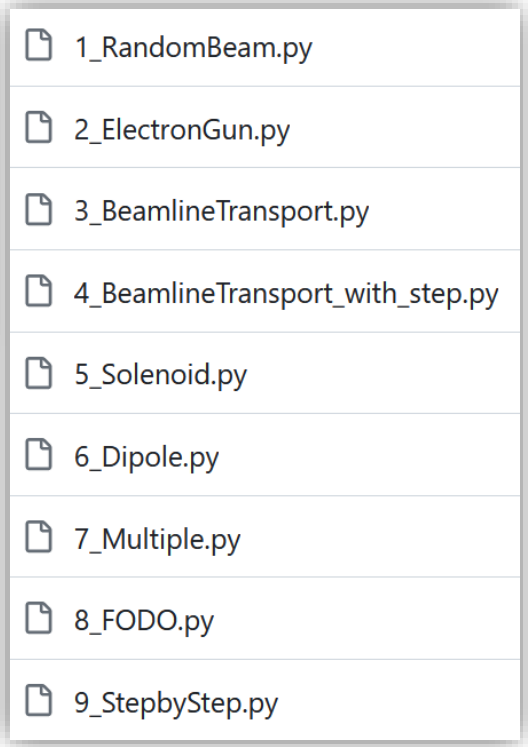
Yardımcı örnekler

https://github.com/acanbay/apricot



Örnekleri içerir





Kurulum

https://github.com/acanbay/apricot

APRICOT launch binder

> **Particle Tracking Module** for python - Ali Can Canbay

APRICOT is a python3 module that simulates the behavior of particle beams in electromagnetic fields as they pass through various beamline elements and calculates beamline parameters at the end of the beamline.

Installing via pip:

APRICOT can be installed by running the following command in the console (all necessary modules will be installed automatically).

```
pip install apricotbl
```

Manual Installation:

APRICOT requires the **numpy**, **scipy**, and **matplotlib** modules. If these modules are not installed, you can install them with the following command:

```
pip install -r requirements.txt
```

requirements.txt is available on APRICOT's github page.

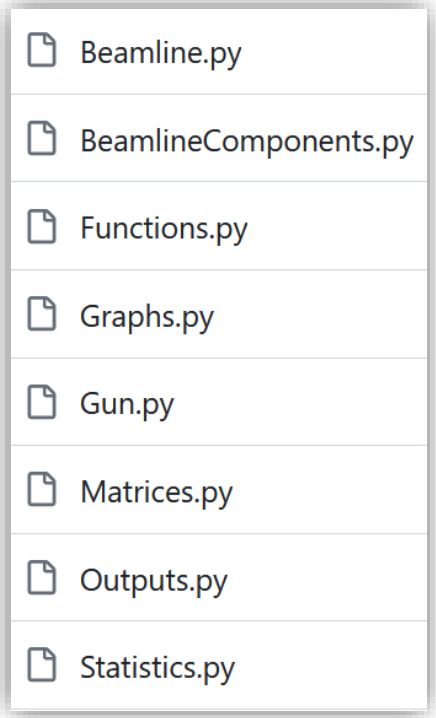
Then, download the latest release and extract it. Enter the extracted file and run the following command via console:

```
python setup.py install
```

Kullanım

```
import apricotbl
```

APRICOT modülü betik dosyasının başında tanıtılmalıdır.




```
import apricotbl.BeamlineComponents as blc

import apricotbl.Beamline as bl

import apricotbl.Functions as fn

import apricotbl.Gun as gun

import apricotbl.Graphs as gp

import apricotbl.Outputs as out
```

Kullanım: Demet üretimi

Rastgele üreteç:

Verilen parametrelerle uyumlu *rastgele* üretilmiş bir demet oluşturmak için **Functions** modülündeki **RandomBeam** fonksiyonu kullanılır.

ParticleTpye	'electron', 'proton' or 'muon'
NumberOfParticles	Parçacık sayısı
BeamEnergy	Demet enerjisi [keV]
x_rms	x-eksenindeki RMS demet genişliği
y_rms	y-eksenindeki RMS demet genişliği [m]
z_rms	z-eksenindeki RMS demet genişliği [m] (opsiyonel)
Emittance_x	x-eksenindeki emittans [m-rad]
Emittance_y	y-eksenindeki emittans [m-rad]
Alpha_x	
Alpha_y	
dE	Enerji yayılımı [keV] (opsiyonel)

```
fn.RandomBeam(ParticleTpye, NumberOfParticles, BeamEnergy, x_rms, y_rms, Emittance_x, Emittance_y, Alpha_x, Alpha_y, z_rms, dE)
```

Kullanım: Demet üretimi

Elektron Tabancası:

Bir elektron tabancasına uygun demet üretmek için **Gun** modülündeki **ElectronGun** sınıfı kullanılır.

NumberOfParticles	Parçacık sayısı
r_Cathode	Katot yarıçapı [m]
Temperature	Katot sıcaklığı [°C]
Voltage	Katot gerilimi [V]

Elektron tabancasını tanımlama:

```
egun = gun.ElectronGun(r_Cathode, Temperature, Voltage)
```

Demet üretme:

```
egun.GenerateBeam(NumberOfParticles)
```

Kullanım: Demet hattı elemanları

Sürüklenme Tüpü:

BeamLineComponent modülündeki **DriftTube** sınıfı kullanılır.

Name	Eleman ismi
Length	Elemanın uzunluğu [m]

```
blc.DriftTube(Name, Length)
```

Kuadrupol Mıknatıs:

BeamLineComponent modülündeki **QuadrupoleMagnet** sınıfı kullanılır.

Name	Eleman ismi
Length	Elemanın uzunluğu [m]
Strength	Mıknatıs gücü

```
blc.QuadrupoleMagnet(Name, Length, Strength)
```

Kullanım: Demet hattı elemanları

Dipol Mıknatıs:

BeamLineComponent modülündeki **DipoleMagnet** sınıfı kullanılır.

Name	Eleman ismi
Length	Elemanın uzunluğu [m]
Angle	Bükme açısı
ybend	Y-ekseninde bükmek için 1 yazılır (opsiyonel)

```
blc.DipoleMagnet(Name, Length, Angle, ybend)
```

Solenoid:

BeamLineComponent modülündeki **Solenoid** sınıfı kullanılır.

Name	Eleman ismi
Length	Elemanın uzunluğu [m]
Strength	Mıknatıs gücü

```
blc.Solenoid(Name, Length, Strength)
```

Kullanım: Demet hattı oluşturma

Demet Hattı:

BeamLine modülündeki **BeamLine** sınıfı kullanılır.

Name	Demet hattının ismi
Elements	Demet hattındaki elemanlar (sırasıyla ve liste olarak verilir)

```
bl.Beamline(Name, Elements)
```

Demeti demet hattına gönderme:

Functios modülündeki **TransportBeam** fonksiyonu kullanılır.

```
fn.TransportBeam( Beam, beamline.Elements )
```

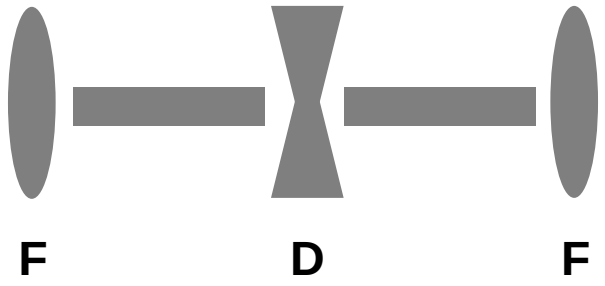
Elemanların içerisinde neler olduğu görülmek istenirse:

```
fn.TransportBeam( Beam, beamline.Elements, dz )
```

Adım uzunluğunun (dz) eleman uzunluklarının tam böleni olması gerekmektedir. Adım uzunluğunun minimum değeri milimetre cinsinden olması önerilir (10 mikrona kadar indirilebilir).

Kullanım: FODO örgüsü

FODO Örgüsü:



BeamLine modülündeki **FODO** sınıfı kullanılır.

Name	Demet hattının ismi
DriftLength	Sürüklenme tüpü uzunluğu
QuadrupoleMagnetLength	Kuadrupol mıknatıs uzunluğu
QuadrupoleMagnetStrength	Kuadrupol mıknatısın gücü

```
bl.FODO(Name, DriftLength, QuadrupoleMagnetLength, QuadrupoleMagnetStrength)
```

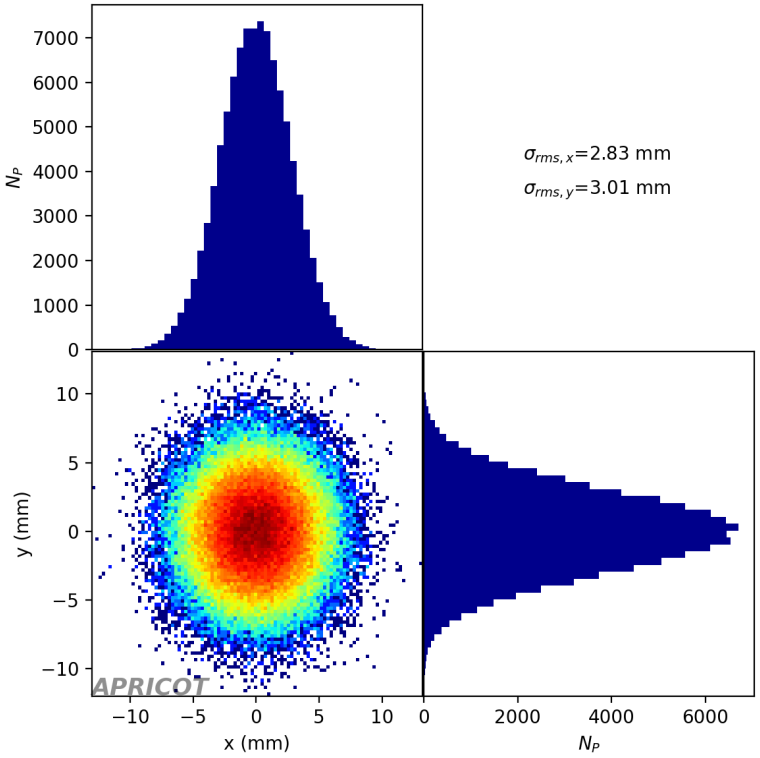
FODO örgüsü de demet hattına eleman olarak dahil edilebilir.

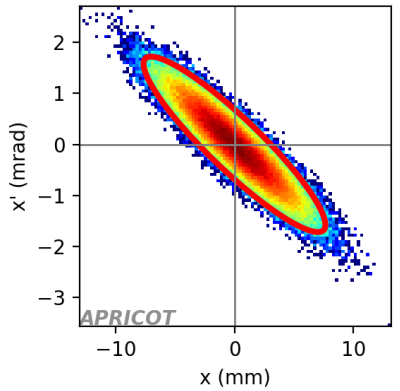
Çıktılar: Demet şekillenimleri

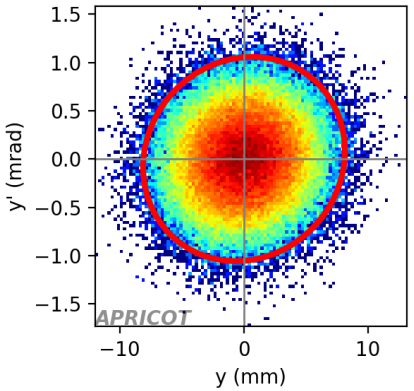
Outputs modülündeki **getBeam** fonksiyonu kullanılır.

`Beam`	Demet sınıfı
`path`	Çıktıların kaydedileceği dizin
`tag`	Çıktı etiketi

```
out.getBeam(Beam, path, tag)
```





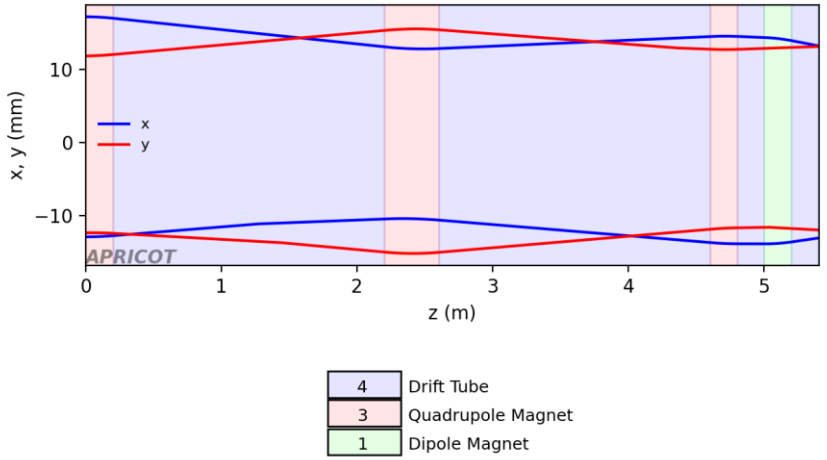


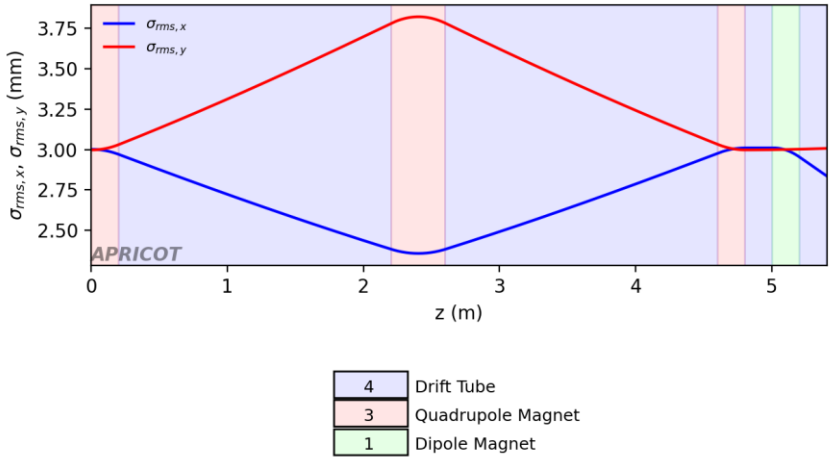
Çıktılar: Demet pozisyon grafikleri

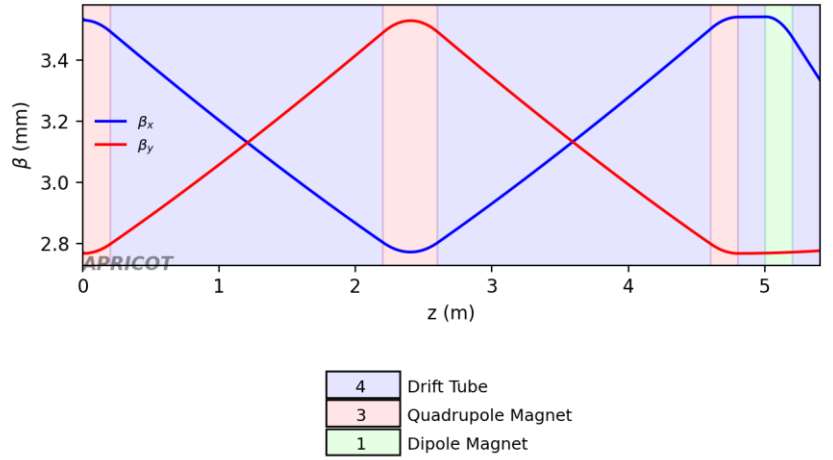
Outputs modülündeki **`getBeamPositions`** fonksiyonu kullanılır.

`Beam`	Demet sınıfı
`beamline.Element`	Demet elemanlarının listesi (sıralı)
`path`	Çıktıların kaydedileceği dizin
`tag`	Çıktı etiketi

```
out.getBeamPositions(Beam, beamline.Elements, path, tag)
```







Çıktılar

Graph modülündeki bazı fonksiyonlar, kaydetmeden görsel üretmek için kullanılabilir.

Demet şekillenimi: xy, yz ve zx eksenlerindeki demet şekillenimlerini verir.

```
gp.plotBeamShape_xy(Beam)
```

Phase Space:

```
gp.plotPhaseSpace(Beam)
```

Pozisyon grafiği: Demetin, demet hattı içindeki davranışını verir.

```
gp.plotPositionGraph( Beam, beamline.Elements )
```

Pozisyon grafiği (RMS): Demetin, demet içindeki davranışını RMS uzunluklara göre verir.

```
gp.plotPositionGraph_RMSsize( Beam, beamline.Elements )
```

Beta Fonksiyonu:

```
gp.plotBetaFunctions( Beam, beamline.Elements )
```