JÜLICH FORSCHUNGSZENTRUM

Proposition for the Beam Smearing Implementation inside Pandaroot

December 10, 2009 | T. Randriamalala, J. Ritman and T. Stockmanns

Outiline

Introduction and Motivation

FairPrimaryGenerator Vertex Smearing

HESR Beam Properties

Beam Smearing

Tentative Implementation

Conclusion

Introduction and Motivation

Introduction and Motivation

- Beam smearing can be done by acting on the outgoing generated particles (cf. PCM 09.09 (Smearing of the DPM outpout)).

Introduction and Motivation

- Beam smearing can be done by acting on the outgoing generated particles (cf. PCM 09.09 (Smearing of the DPM outpout)).
- Similarly, one can operate by using FairPrimaryGenerator (modified).

Introduction and Motivation

- Beam smearing can be done by acting on the outgoing generated particles (cf. PCM 09.09 (Smearing of the DPM outpout)).
- Similarly, one can operate by using FairPrimaryGenerator (modified).
- But, the current implementation FairPrimaryGenerator can smear only vertices
  $\Longrightarrow$ *Angular smearing* should be implemented (beam particles do not have identical 4-momemtum)

Introduction and Motivation

- Beam smearing can be done by acting on the outgoing generated particles (cf. PCM 09.09 (Smearing of the DPM outpout)).
- Similarly, one can operate by using FairPrimaryGenerator (modified).
- But, the current implementation FairPrimaryGenerator can smear only vertices
  $\Longrightarrow$ *Angular smearing* should be implemented (beam particles do not have identical 4-momemtum)
- Also, the vertex smearing should be modified according to the HESR beam properties.

FairPrimaryGenerator Vertex Smearing

- In the simulation macro:

```
FairPrimaryGenerator* primGen = new FairPrimaryGenerator();

primGen->SetTarget(0.0, 0.2);
primGen->SmearVertexZ(kTRUE);


primGen->SetBeam(0.0, 0.0, 0.08, 0.08);
primGen->SmearVertexXY(kTRUE);
```

FairPrimaryGenerator Vertex Smearing

- In the simulation macro:

```
FairPrimaryGenerator* primGen = new FairPrimaryGenerator();
```

```
primGen->SetTarget(0.0, 0.2);
primGen->SmearVertexZ(kTRUE);
```

a flat distribution for the Z-position of the vertex with a width of $0.2 cm$ centered on 0.0

```
primGen->SetBeam(0.0, 0.0, 0.08, 0.08);
primGen->SmearVertexXY(kTRUE);
```

X and Y will be smeared both by a gaussian centered on 0.0, with $\sigma = 0.08 cm$.

FairPrimaryGenerator Vertex Smearing

- In the simulation macro:

```
FairPrimaryGenerator* primGen = new FairPrimaryGenerator();
```

```
primGen->SetTarget(0.0, 0.2);
primGen->SmearVertexZ(kTRUE);
```

a flat distribution for the Z-position of the vertex with a width of 0.2*cm* centered on 0.0

```
primGen->SetBeam(0.0, 0.0, 0.08, 0.08);
primGen->SmearVertexXY(kTRUE);
```

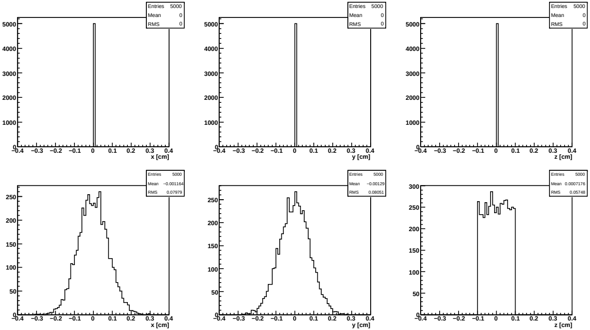
X and Y will be smeared both by a gaussian centered on 0.0, with $\sigma = 0.08cm$.

- Run a simulation:

```
fRun->SetGenerator(primGen);
FairBoxGenerator* boxGen =...
boxGen->SetXYZ(0., 0., 0.);
primGen->AddGenerator(boxGen);
```

Smearing Output

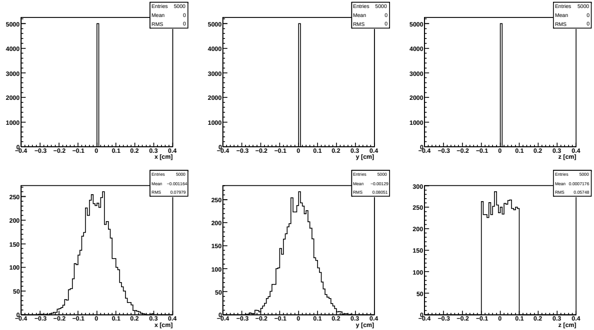
- Before vertex smearing
- After vertex smearing



[1] D. Reistad, B. Galnander, K. Rathsman, A. Sidorin, "Calculations on High-Energy Electron Cooling in the HESR", Proceedings of COOL 2007, Bad Kreuznach. and V.Ziemann, NIMA 556 (2006) 641.

Smearing Output

- Before vertex smearing
- After vertex smearing

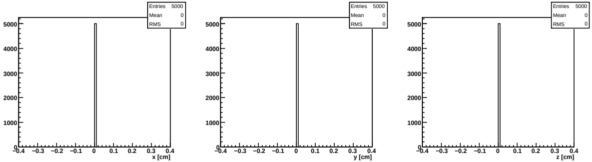


- This should be treated differently according the HESR beam properties [1]

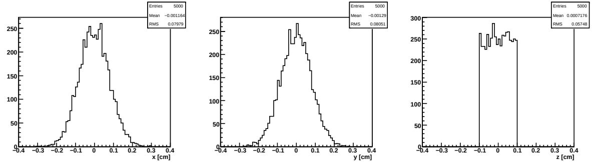
[1] D. Reistad, B. Galnander, K. Rathsman, A. Sidorin, "Calculations on High-Energy Electron Cooling in the HESR", Proceedings of COOL 2007, Bad Kreuznach. and V.Ziemann, NIMA 556 (2006) 641.

Smearing Output

- Before vertex smearing



- After vertex smearing



- This should be treated differently according the HESR beam properties [1]

  + angluar smearing

[1] D. Reistad, B. Galnander, K. Rathsman, A. Sidorin, "Calculations on High-Energy Electron Cooling in the HESR", Proceedings of COOL 2007, Bad Kreuznach. and V.Ziemann, NIMA 556 (2006) 641.

Antiproton Beam Properties

Antiproton Beam Properties

- At the IP, the emittance ε of the $\bar{p}$-beam is 1 $mm\,mrad$.

Antiproton Beam Properties

- At the IP, the emittance ε of the $\bar{p}$-beam is 1 *mm mrad* .
- $\varepsilon = 2\sigma_x \cdot 2\sigma_\theta$
  - σ_x : RMS of the transversal divergance of the beam.
  - σ_θ : RMS of the angular divergance of the beam.

Antiproton Beam Properties

- At the IP, the emittance ε of the $\bar{p}$-beam is 1 $mm\,mrad$.
- $\varepsilon = 2\sigma_x \cdot 2\sigma_\theta$
  - σ_x : RMS of the transversal divergance of the beam.
  - σ_θ : RMS of the angular divergance of the beam.
- For $\sigma_x \leq 0.8mm$, $\mathcal{L}_{eff} \geq 0.8\mathcal{L}_{max}$ because of the beam-target overlap (pellet target).

Vertex Smearing

Event-by-event smearing:

Vertex Smearing

Event-by-event smearing:

- Choose $\sigma_x = \sigma_y = 0.8mm$.

Vertex Smearing

Event-by-event smearing:

- Choose $\sigma_x = \sigma_y = 0.8mm$.
- If $\sigma_x \leq 0.5 \times R$
  $\Longrightarrow$ Homogeneous "square" distribution of $\bar{p}$ in the beam with a side of $(2 \times \sigma_x)$.
  where R is the cross section radius of the pellet flux ($R \sim 2mm$).

Vertex Smearing

Event-by-event smearing:

- Choose $\sigma_x = \sigma_y = 0.8mm$.
- If $\sigma_x \leq 0.5 \times R$
  $\Longrightarrow$ Homogeneous "square" distribution of $\bar{p}$ in the beam with a side of $(2 \times \sigma_x)$.
  where R is the cross section radius of the pellet flux ($R \sim 2mm$).
- Gaussian smearing in the z-direction with $\sigma_z = R/2 = 1mm$.

Angular Smearing

For the 3-momentum vectors $\vec{p}_i$ ($i = \{p, \bar{p}\}$):

Angular Smearing

For the 3-momentum vectors $\vec{p}_i$ ($i = \{p, \bar{p}\}$):

- Rotation of $\delta\theta$ around $\hat{n} = \frac{\vec{p}_i \times \hat{k}}{|\vec{p}_i| \cdot |\hat{k}|}$ ($\hat{k}(0, 0, 1)$.)

  $\delta\theta$ is a Gaussian distribution with $\sigma_\theta = 0.3 mrad$.
  ($\delta\theta$ must be positive numbers.)

Angular Smearing

For the 3-momentum vectors $\vec{p}_i$ $(i = \{p, \bar{p}\})$:

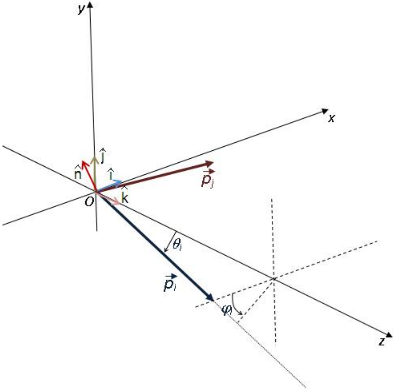
- Rotation of $\delta\theta$ around $\hat{n} = \frac{\vec{p}_i \times \hat{k}}{|\vec{p}_i| \cdot |\hat{k}|}$ $(\hat{k}(0,0,1).)$
  $\delta\theta$ is a Gaussian distribution with $\sigma_\theta = 0.3 mrad$.
  ($\delta\theta$ must be positive numbers.)
- Rotation of $\delta\varphi$ around $\vec{p}_i$.
  $\delta\varphi$ is an uniform distribution $[-\pi, +\pi]$.

Angular Smearing

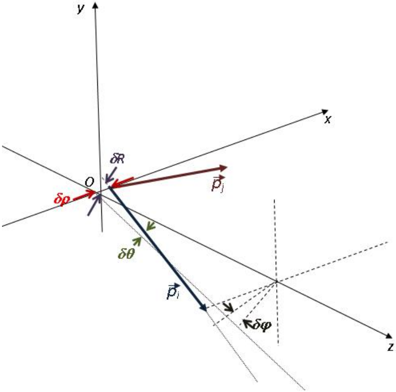
For the 3-momentum vectors $\vec{p}_i$ ($i = \{p, \bar{p}\}$):

- Rotation of $\delta\theta$ around $\hat{n} = \frac{\vec{p}_i \times \hat{k}}{|\vec{p}_i| \cdot |\hat{k}|}$ ($\hat{k}(0,0,1)$.)
  $\delta\theta$ is a Gaussian distribution with $\sigma_\theta = 0.3 mrad$.
  ($\delta\theta$ must be positive numbers.)
- Rotation of $\delta\varphi$ around $\vec{p}_i$.
  $\delta\varphi$ is an uniform distribution $[-\pi, +\pi]$.
- No correlation between $\delta\theta$ and $\delta\varphi$.

Before smearing



After smearing



Tentative Implementation

Tentative Implementation

- In FairPrimaryGenerator class:

Tentative Implementation

- In FairPrimaryGenerator class:
  - change the flat distribution on Z to a gaussian.

Tentative Implementation

- In FairPrimaryGenerator class:
  - change the flat distribution on Z to a gaussian.
  - change the gaussian distributions on X and Y to an uniform.

Tentative Implementation

- In FairPrimaryGenerator class:
  - change the flat distribution on Z to a gaussian.
  - change the gaussian distributions on X and Y to an uniform.
  - introduce the angular smearing

Tentative Implementation

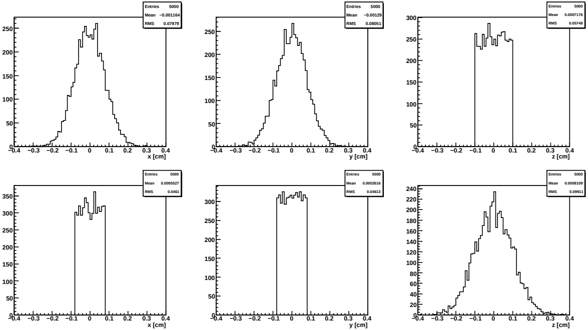
- In FairPrimaryGenerator class:
  - change the flat distribution on Z to a gaussian.
  - change the gaussian distributions on X and Y to an uniform.
  - introduce the angular smearing
- In simulation macro, add:
  *primGen->SetAngularDivergence(0.3*1E-3);*
  primGen->SmearAngle(kTRUE);

Tentative Implementation

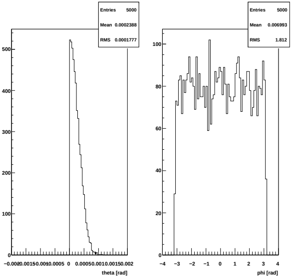
- In FairPrimaryGenerator class:
  - change the flat distribution on Z to a gaussian.
  - change the gaussian distributions on X and Y to an uniform.
  - introduce the angular smearing
- In simulation macro, add:
  *primGen->SetAngularDivergence(0.3*1E-3);*
  primGen->SmearAngle(kTRUE);
- Use FairBoxGenerator to generate event (5000 event, 1 particle/event)

Results

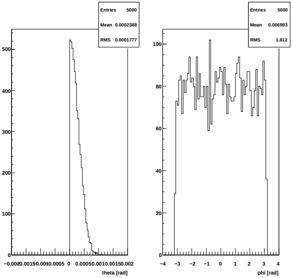
- Effect of the modification on the vertex (bottom plots) in comparison with the existing one (top plot):



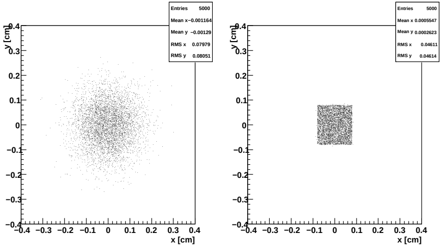
- Angular divergence distribution:



- Angular divergence distribution:



- Beam profile in xy plane at 1m from the IP:



Conclusion

Conclusion

- Pandaroot provides a beam smearing tool.

Conclusion

- Pandaroot provides a beam smearing tool.
- Some change need to be considered : vertex and angular smearing distribution,

Conclusion

- Pandaroot provides a beam smearing tool.
- Some change need to be considered : vertex and angular smearing distribution,
- If $\varepsilon = 1$ *mm mrad*, the resolution of one subdetector is affected ($\delta\theta \geq$ 0.3*mrad* for the LM).