Quasi Random Generator

Avni Aksoy

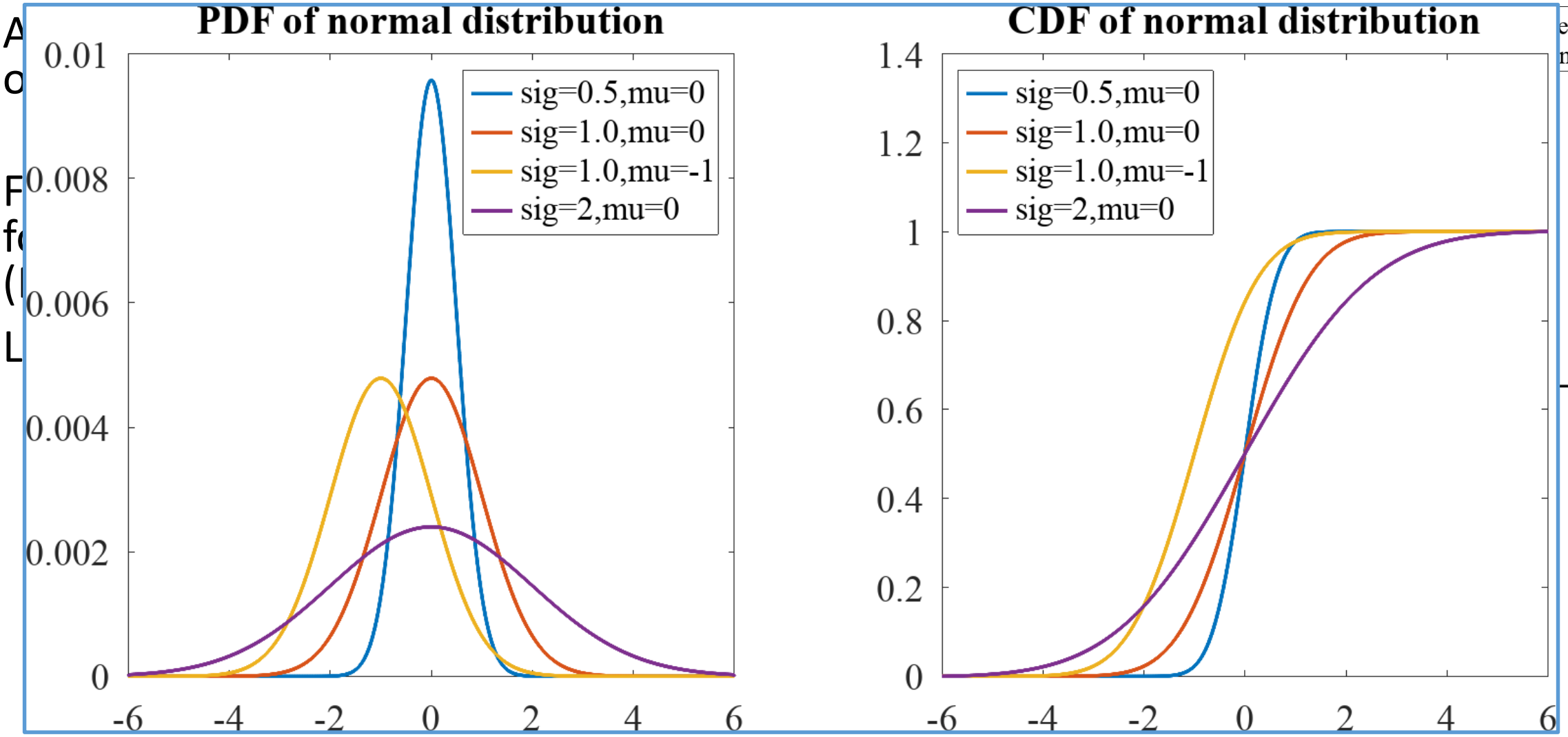
Ankara University & CERN

Introduction

- For the most of the gun/injector study, the emision process determines the beam properties the cathode (and may continue through rest of the machine..)
  - The lowest emittance value one can get at the injection
  - Some 3D-distribution parameters (i.e. uncorrelated energy spread..)
- Usually the (macro) particles are generated with some physical constrains «randomly»
- In order to have good statistical agreement one needs to create «enough» number of macro-particle
  - we have «pseudo-real» numbers not really «real-numbers»..
  - if we do not have enough random number we might be struggling with «noises»
    - Large computing power
- Having «quasi-random» numbers let us to cover the domain of interest quickly over «pure random» numbers.

PDF and CDF

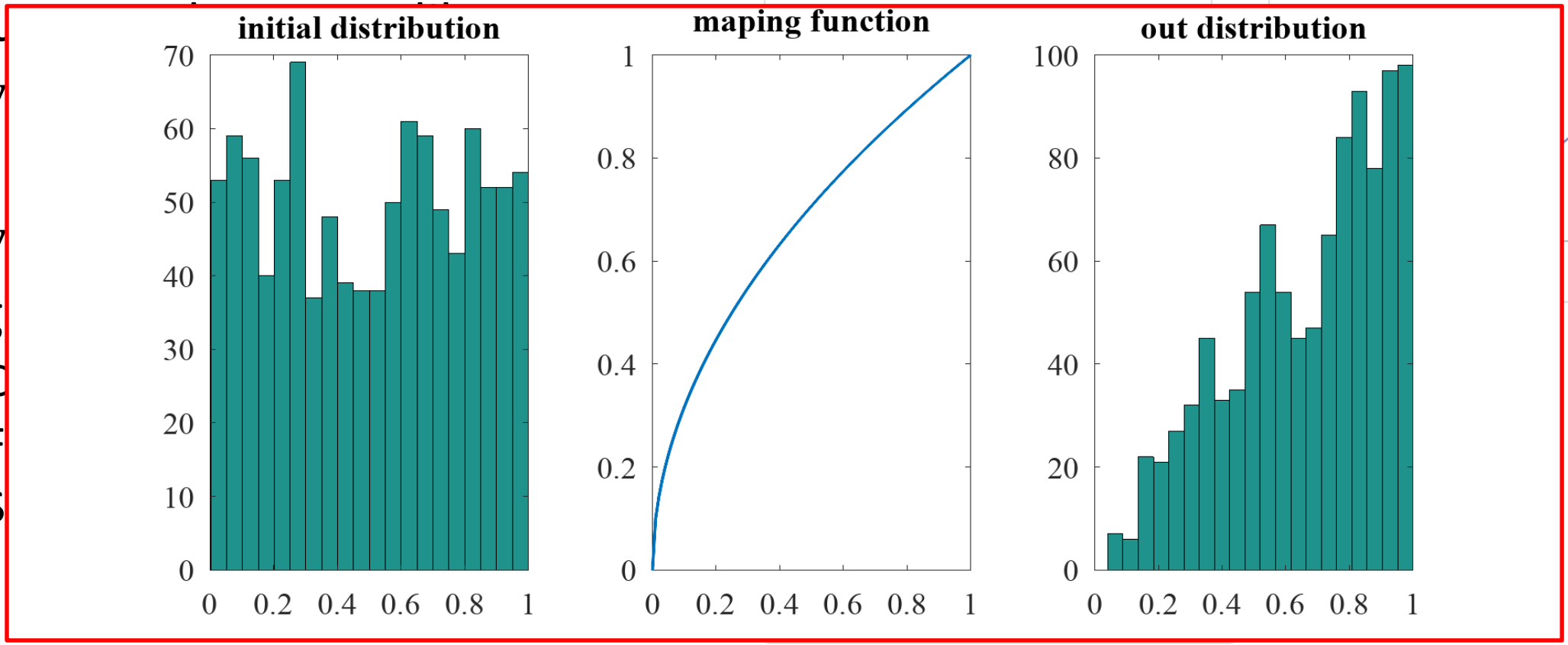
- A
  o
- F
  f
  (
- L



eate
nious

Mapping Uniform Distribution to any PDF

- Assu
  - y
- The
  - y
- To g
  - C
  - F
  - S



- If w ith give

If we have «statistical - good» uniform distribution the distribution we create will be «statistical – good» also..

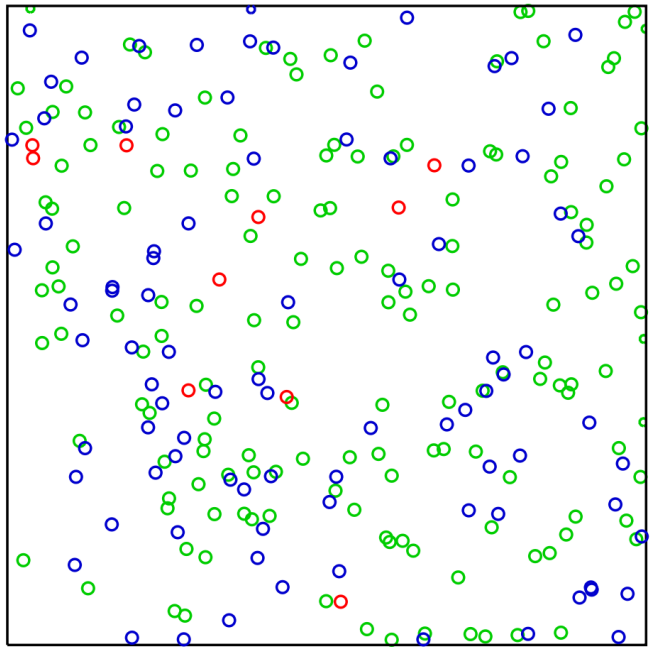
«Statistical good» distribution (Low-discrepancy sequence)

- It is not a random distribution actually it is a sequence..
  - Simplest uniform N number between 0-1 is
    - $X_i = i/N$
  - However we can not use this sequence in order not to have correlation between coordinates
- Se
  ne
  -
- Th
  -
  -
- ior
  -
- Ra
  - X=randPDF(PDF,SEQ)

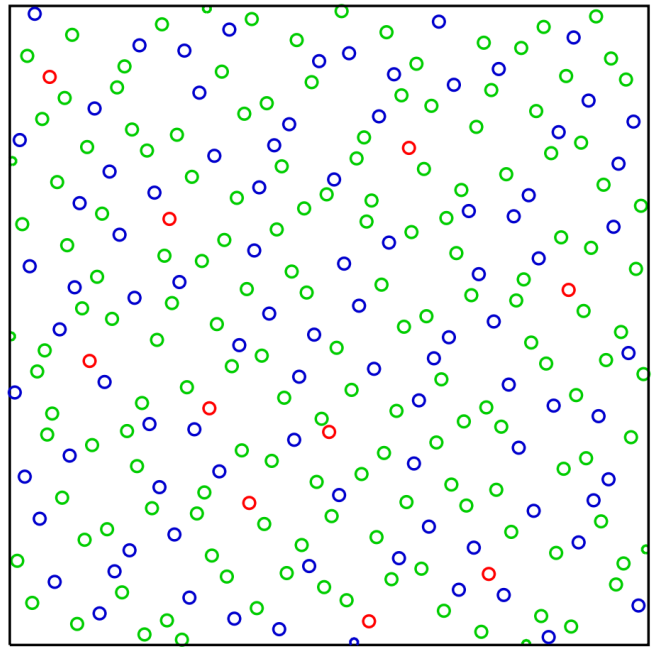
Algorithm

```
algorithm Halton-Sequence is
    inputs: index i
            base b
    output: result r

    f ← 1
    r ← 0

    while i > 0 do
        f ← f/b
        r ← r + f * (i mod b)
        i ← ⌊i/b⌋

    return r
```

Pseudorandom



Halton Sequence



The Generator

Particle Generator

- Most of the particle-tracking tools require, the initial particle distribution externally instead of generating them internally..
- Astra-*generator* is commonly used for creating initial distribution.. However
  - it has limited distribution types..
  - İt has limitations with random numbers with «noise_reduc» options meanning can not create quasi-rand distributions for all types
  - İt has few bugs..
  - And source code is not available..
  - …
- We decided to develope a script (octave/matlab) to do same job which uses same input file and returns same output..

	1	2	3	4	5	6	7	8	9	10
Parameter	x	y	z	px	py	pz	clock	macro charge	particle index	status flag
Unit	m	m	m	eV/c	eV/c	eV/c	ns	nC		

Particle Generator

- Each macroparticle is represented by 3D-spacial (x,y,z) and 3D-momentum (p_x,p_y,p_z) coordinates.
- There might be coorelation between them. Like;
  - x-y coordinates can be distributed based on cathode geometry or laser profile
  - x-p_x might requested to provide an emittance value (ε)
  - The energy therefore (p_x,p_y,p_z) coordinates might needed to be equal some value..
  - …
- Before creating any coordinate the relations between them needs to be evaluated like..
  - 3D spacial relation -> 2D spacial relation -> Create 1D space ->

    → 3D momentum relation -> 2D momentum relatio -> Create 1D momentum

Typical input file (with name «generator.in»)

```
&INPUT
    FNAME = 'Example.ini'
    Add=FALSE,      N_add=0,
    IPart=500,      Species='electrons'
    Probe=True,
    Noise_reduc=T.
    Cathode=
    Q_total=
    Ref_zpos
    Dist_z=
    Dist_pz=
    Dist_x=
    Dist_y= 'g' ,      Sig_y=0.75E0, Dist_py= 'g' , Nemit_y=1.0E0, Cor_py=0.0E0
/
```

Name of output file

Number of particles

Particle kind

Wheather will be created on cathode (zi=z0) or in space (ti=t0)

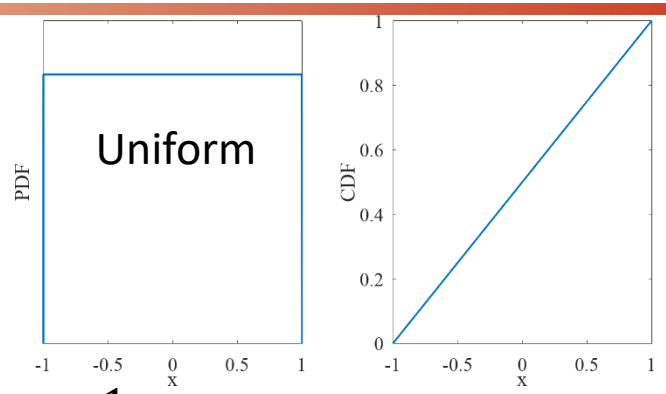
To run the generator in octave/matlab

```
beam=make_beam('generator.in');
```

Gaussian transverse coordinates and transverse momentum..

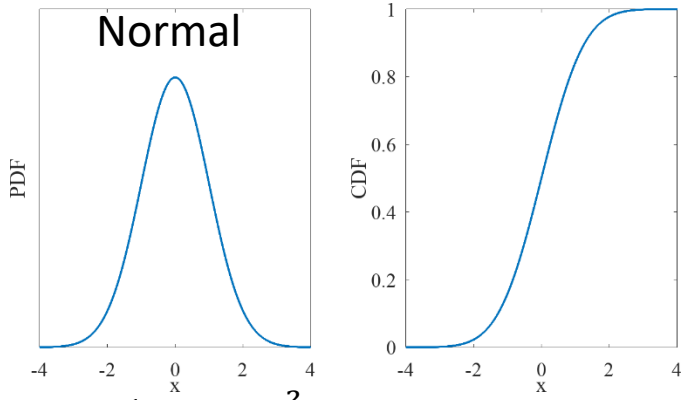
1D distributions

Dimension	Key word	Parameter *FWHM* or σ	unit
temporal, t	Dist_z = '**u**niform'	Lt or sig_clock	ns
longitudinal z	Dist_z = '**u**niform'	Lz or sig_z	mm
longitudinal E_{kin}	Dist_pz = '**u**niform'	LE or sig_Ekin or emit_z	keV or keVmm
transverse x	Dist_x = '**u**niform'	Lx or sig_x	mm
transverse y	Dist_y = '**u**niform'	Ly or sig_y	mm
transverse p_x	Dist_px = '**u**niform'	Lpx or sig_px or Nemit_x	eV/c or mrad mm
transverse p_y	Dist_py = '**u**niform'	Lpy or sig_py or Nemit_y	eV/c or mrad mm



$$f(x) = \frac{1}{FWHM} \quad \text{for } |x| < FWHM, \qquad \sigma = \frac{FWHM}{2\sqrt{3}}$$

Dimension	Key word	Parameter σ	unit
temporal, t	Dist_z = '**g**aussian'	sig_clock	ns
longitudinal z	Dist_z = '**g**aussian'	sig_z	mm
longitudinal E_{kin}	Dist_pz = '**g**aussian'	sig_Ekin or emit_z	keV or keVmm
transverse x	Dist_x = '**g**aussian'	sig_x	mm
transverse y	Dist_y = '**g**aussian'	sig_y	mm
transverse p_x	Dist_px = '**g**aussian'	sig_px or Nemit_x	eV/c or mrad mm
transverse p_y	Dist_py = '**g**aussian'	sig_py or Nemit_y	eV/c or mrad mm



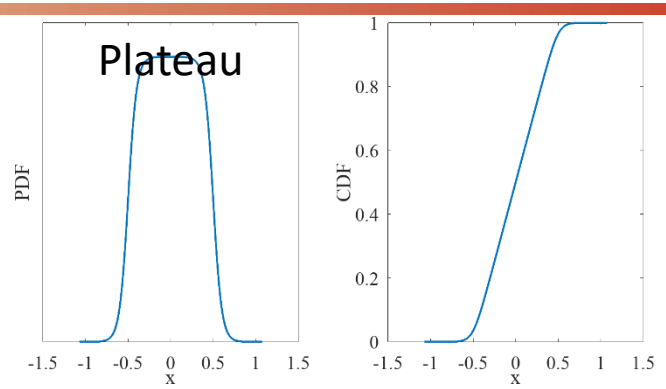
$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad , FWHM = 2\sqrt{2\ln(2)}\sigma$$

1D distributions

Dimension	Key word	Parameter *L and r*	unit
temporal, t	Dist_z = '**p**lateau'	Lt and rt	ns
longitudinal z	Dist_z = '**p**lateau'	Lz and rz	mm
longitudinal E_{kin}	Dist_pz = '**p**lateau'	LE and rE	keV or keVmm
transverse x	Dist_x = '**p**lateau'	Lx and rx	mm
transverse y	Dist_y = '**p**lateau'	Ly and ry	mm
transverse p_x	Dist_px = '**p**lateau'	Lpx and rpx	eV/c or mrad mm
transverse p_y	Dist_py = '**p**lateau'	Lpy and rpy	eV/c or mrad mm

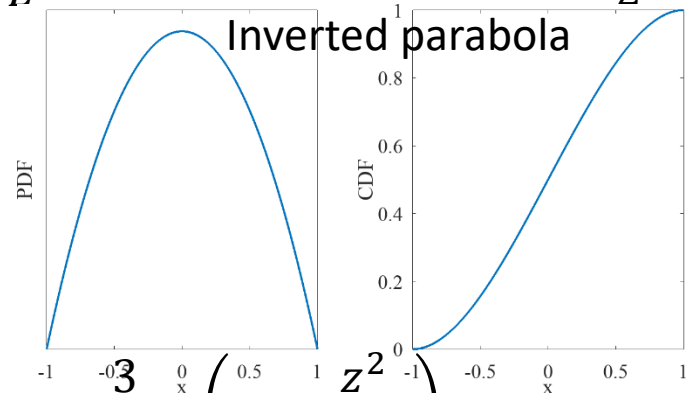
Dimension	Key word	Parameter σ	unit
temporal, t	Dist_z = '**inverted**'	sig_clock	ns
longitudinal z	Dist_z = '**inverted**'	sig_z	mm
longitudinal E_{kin}	Dist_pz = '**inverted**'	sig_Ekin or emit_z	keV or keVmm
transverse x	Dist_x = '**inverted**'	sig_x	mm
transverse y	Dist_y = '**inverted**'	sig_y	mm
transverse p_x	Dist_px = '**inverted**'	sig_px or Nemit_x	eV/c or mrad mm
transverse p_y	Dist_py = '**inverted**'	sig_py or Nemit_y	eV/c or mrad mm

Plateau



$$f(x) = \frac{1}{L}\left(1 + e^{\frac{2}{rt}(2|x|-L)}\right)^{-1} \quad \text{for } rt < \frac{L}{2}\,,\ FWHM = L$$

Inverted parabola



$$f(x) = \frac{3}{4\,z_{max}}\left(1 - \frac{z^2}{z_{max}^2}\right) \quad \text{for } |z| < z_{max}\,,$$

$$FWHM = \sqrt{2}z_{max} \quad , \qquad \sigma = \frac{z_{max}}{\sqrt{5}}$$

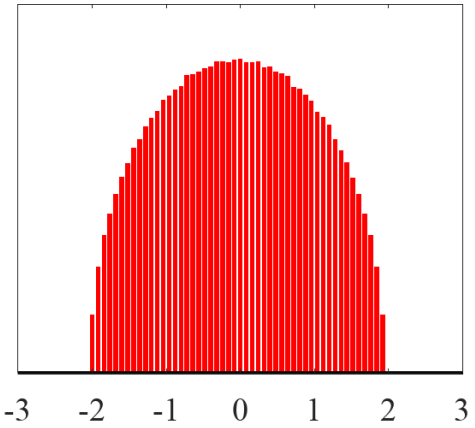
2D Distributions – Radial

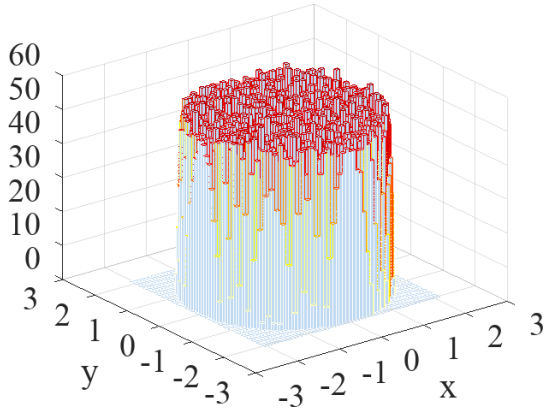
- Valid only for transverse space (x-y) and transverse momentum (px-py) pairs
- Command: ‘r’, ‘ru’, ‘radial’, ‘radial-uniform’

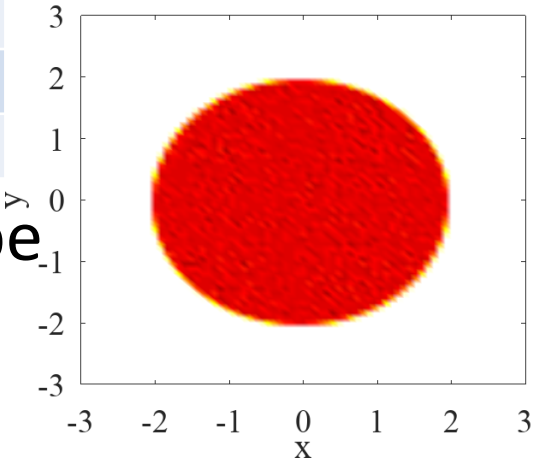
$$f(x) = \frac{1}{\pi r^2} \quad \text{for} \quad x^2 + y^2 \leq r^2\,, \quad FWHM = \sqrt{3}r, \quad \sigma = \frac{r}{2}$$

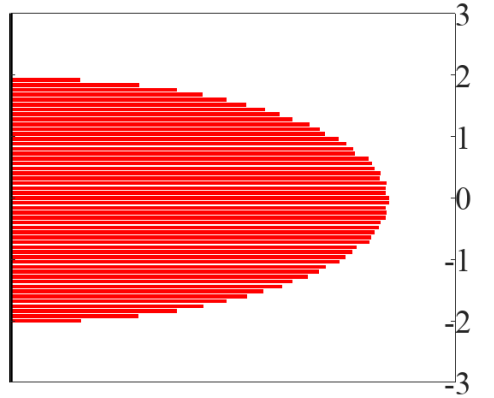
Dimension	Key word	Parameter *FWHM* or σ	unit
transverse x	Dist_x = ‘ru’	Lx or sig_x	mm
transverse y	Dist_y = ‘**ru**'	Ly or sig_y	mm
transverse p_x	Dist_px = ‘ru’	Lpx or sig_px or Nemit_x	eV/c or mrad mm
transverse p_y	Dist_py = ‘**ru**'	Lpy or sig_py or Nemit_y	eV/c or mrad mm

- At least one componenet (x or y) has to be defined
  - if x coordinate is defined, y is ignored
  - İf Nemit_u is defined sig_u is ignored





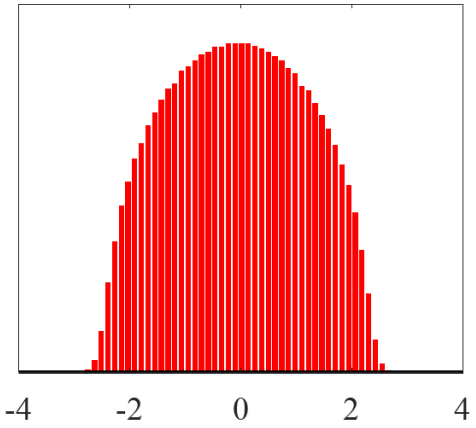


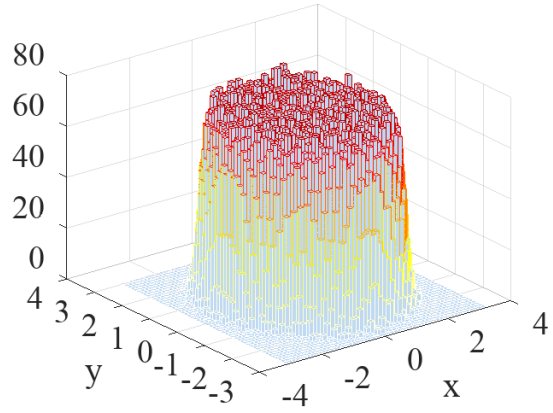


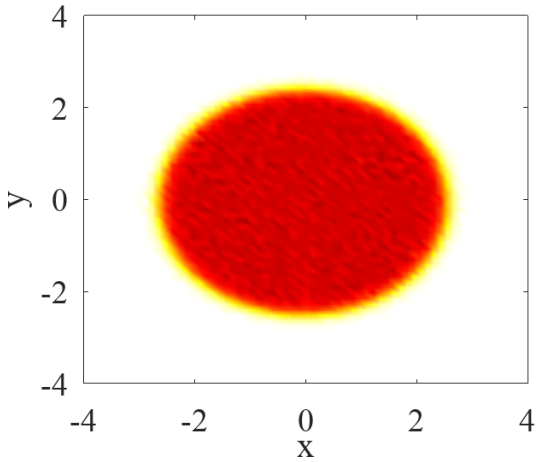
2D Distributions – radial-plateau

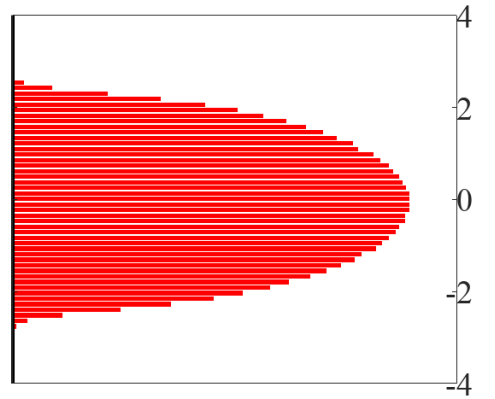
- Valid only for transverse space (x-y) and transverse momentum (px-py) pairs
- Command: ‘rp’, ‘radial-plateau’
- At least one componenet (x or y) has to be defined
  - if x coordinate is defined, y is ignored

Dimension	Key word	Parameter *L and r*	unit
transverse x	Dist_x = ‘rp’	Lx and rx	mm
transverse y	Dist_y = ‘**rp**'	Ly and ry	mm
transverse p_x	Dist_px = ‘rp’	Lpx and rpx	eV/c
transverse p_y	Dist_py = ‘**rp**'	Lpy and rpy	eV/c





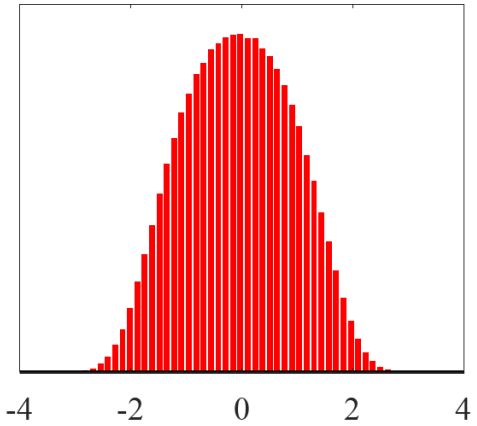


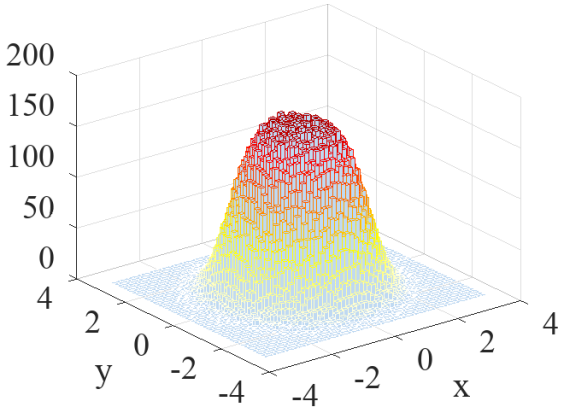


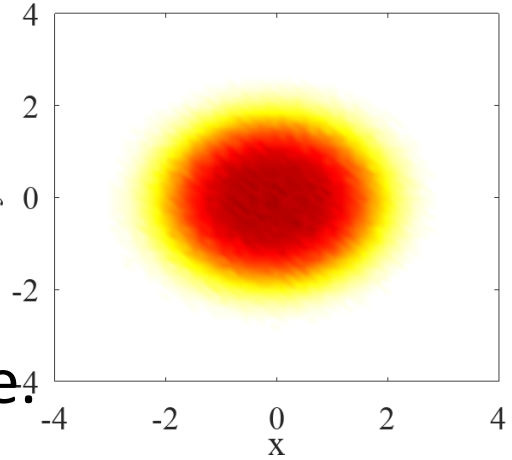
2D Distributions – radial-gaussian

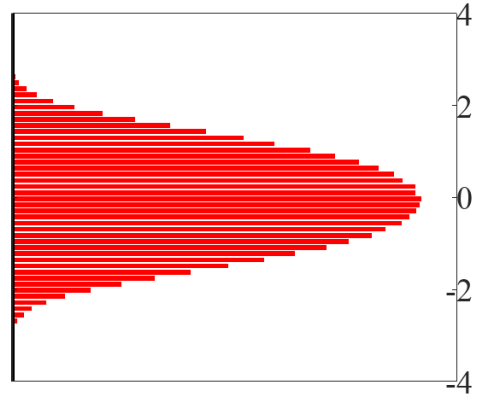
- Valid only for transverse space (x-y) and transverse momentum (px-py) pairs
- Command: ‘rg’, ‘radial-gaussian’
- At least one componenet (x or y) has to be defined
  - if x coordinate is defined, y is ignored
  - İf Nemit_u is defined sig_u is ignored

Dimension	Key word	Parameter *sigma*	unit
transverse x	Dist_x = ‘**rg**’	sig_x	mm
transverse y	Dist_y = ‘**rg**'	sig_y	mm
transverse p_x	Dist_px = ‘rg’	sig_px or nemit_x	eV/c or mrad mm
transverse p_y	Dist_py = ‘**rg**'	sig_py or nemit_y	eV/c or mrad mm







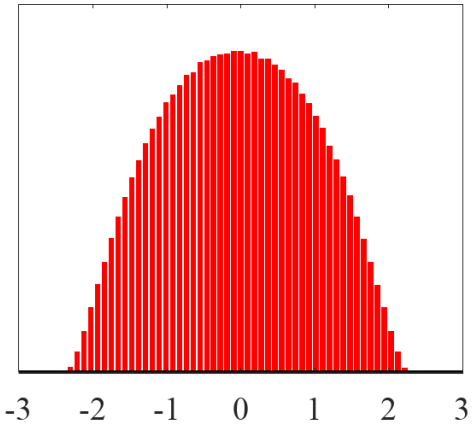


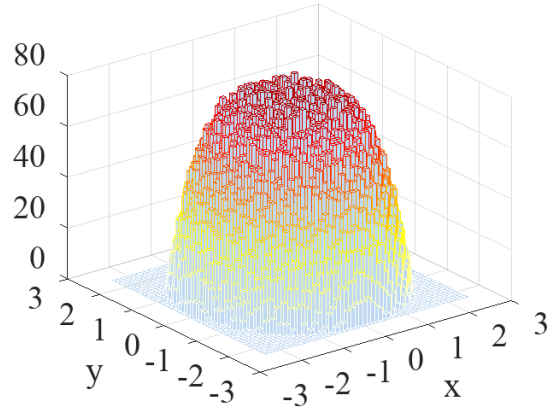
- İf C_sig_? İs defined, gaussian is **truncated**, i.e. x_{max}=C_sig_x ×sig_x

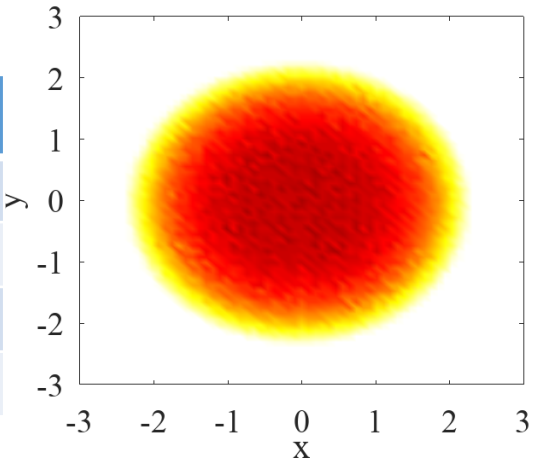
2D Distributions – radial-parabola

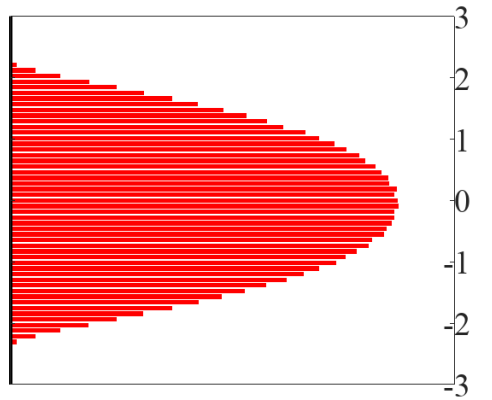
- Valid only for transverse space (x-y) and transverse momentum (px-py) pairs
- Command: ‘ri’, ‘radial-parabola’
- At least one componenet (x or y) has to be defined
  - if x coordinate is defined, y is ignored
  - İf Nemit_u is defined sig_u is ignored

Dimension	Key word	Parameter *sigma*	Unit
transverse x	Dist_x = ‘ri’	sig_x	mm
transverse y	Dist_y = ‘**ri**'	sig_y	mm
transverse p_x	Dist_px = ‘ri’	sig_px or nemit_x	eV/c
transverse p_y	Dist_py = ‘**ri**'	sig_py or nemit_y	eV/c









3D distributions - ellipsoid

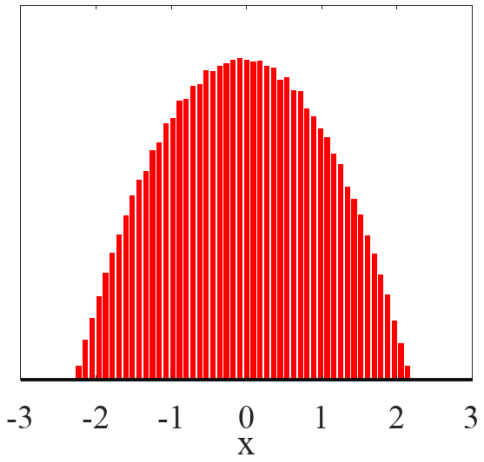
- Valid only for spacial coordinate (x-y-z)
- Command: ‘ue’, ‘uniform-ellipsoid’

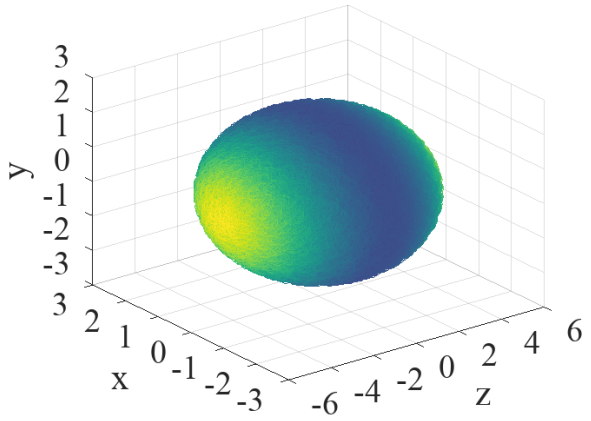
$$f(x) = \frac{1}{4\pi\, Lx\, Ly\, Lz} \quad \text{for} \quad \frac{x^2}{Lx^2} + \frac{y^2}{Ly^2} + \frac{z^2}{Lz^2} \le 1\,,$$

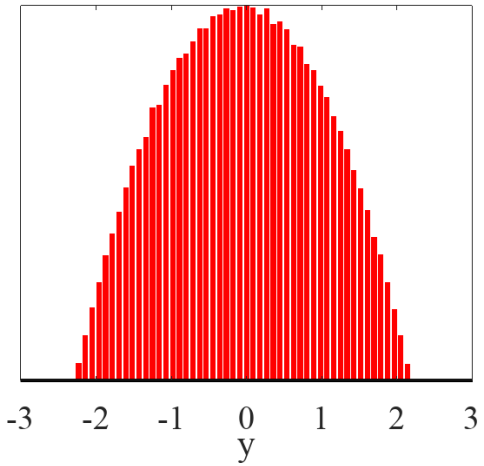
$$FWHM_u = \sqrt{2}\,Lu, \qquad \sigma_u = \frac{Lu}{\sqrt{5}}$$

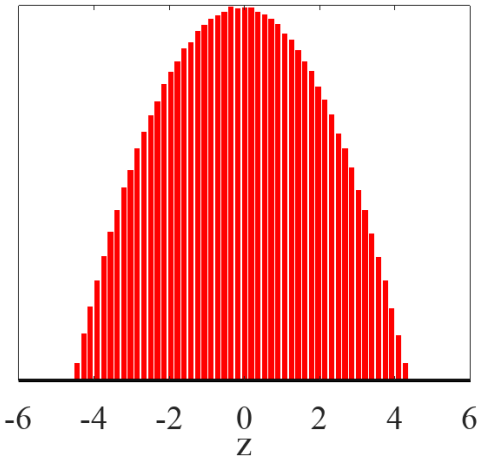
Dimension	Key word	Parameter *FWHM* or σ	unit
temporal, t	Dist_z = '**ue**'	Lt or sig_clock	ns
longitudinal z	Dist_z = '**ue**’	Lz or sig_z	mm
transverse x	Dist_x = ‘ue’	Lx or sig_x	mm
transverse y	Dist_y = ‘**ue**'	Ly or sig_y	mm

- the projection onto the z-axis (eqiv. x- and y-axis) is an parabola
- İf one component is defined others are ignored







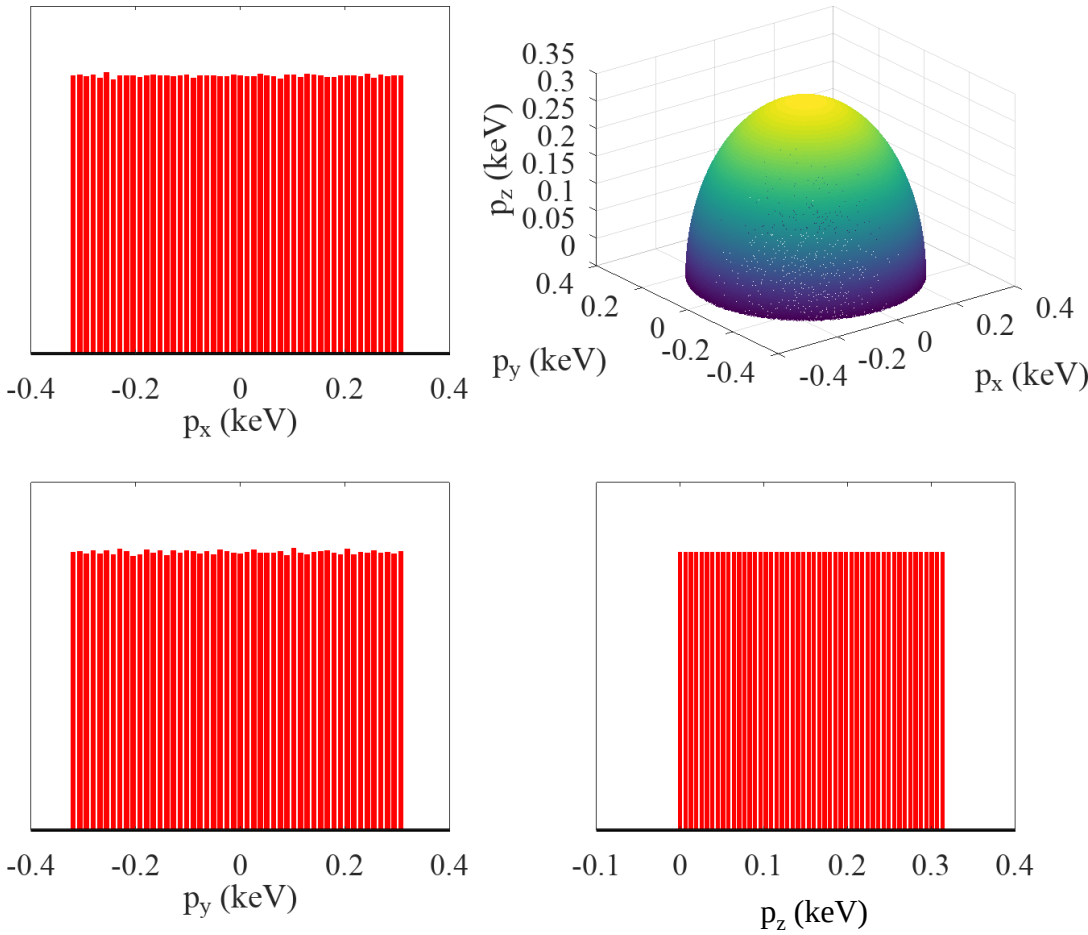


3D distributions - isotropic

- Valid only for momentum space (px-py-pz).
- Command: ‘ue’, ‘uniform-ellipsoid’
- Momentum of particles fills surface of half sphere.
  - $p_x^2 + p_y^2 + p_x^2 = p^2 = E_{kin}^2 + 2E_{kin}$
- RMS momentums: $\sigma_{p_x} = \sigma_{p_y} = 2\sigma_{p_z} = \frac{p}{\sqrt{3}}$
- Mean longitudinal momentum $\langle p_z \rangle = \frac{p}{2}$
- Transverse norm emittance: $\varepsilon_x = \frac{\sigma_x}{\sqrt{3}} \sqrt{\frac{2\,E_{kin}}{m_0\,c^2}}$

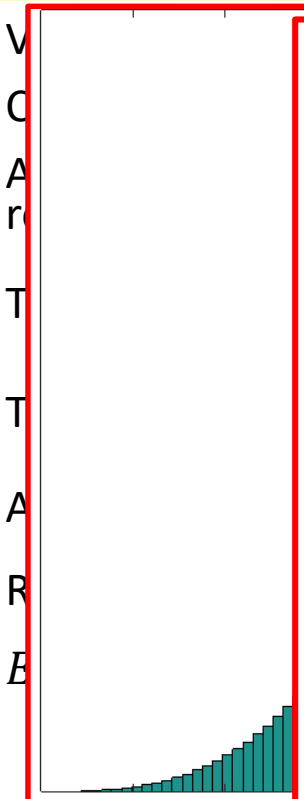
Dimension	Key word	Parameter *FWHM*	unit
p_x, p_y, p_z	Dist_pz = ‘i'	LE	keV

K. Floettmann, TESLA-FEL Report 1997-01

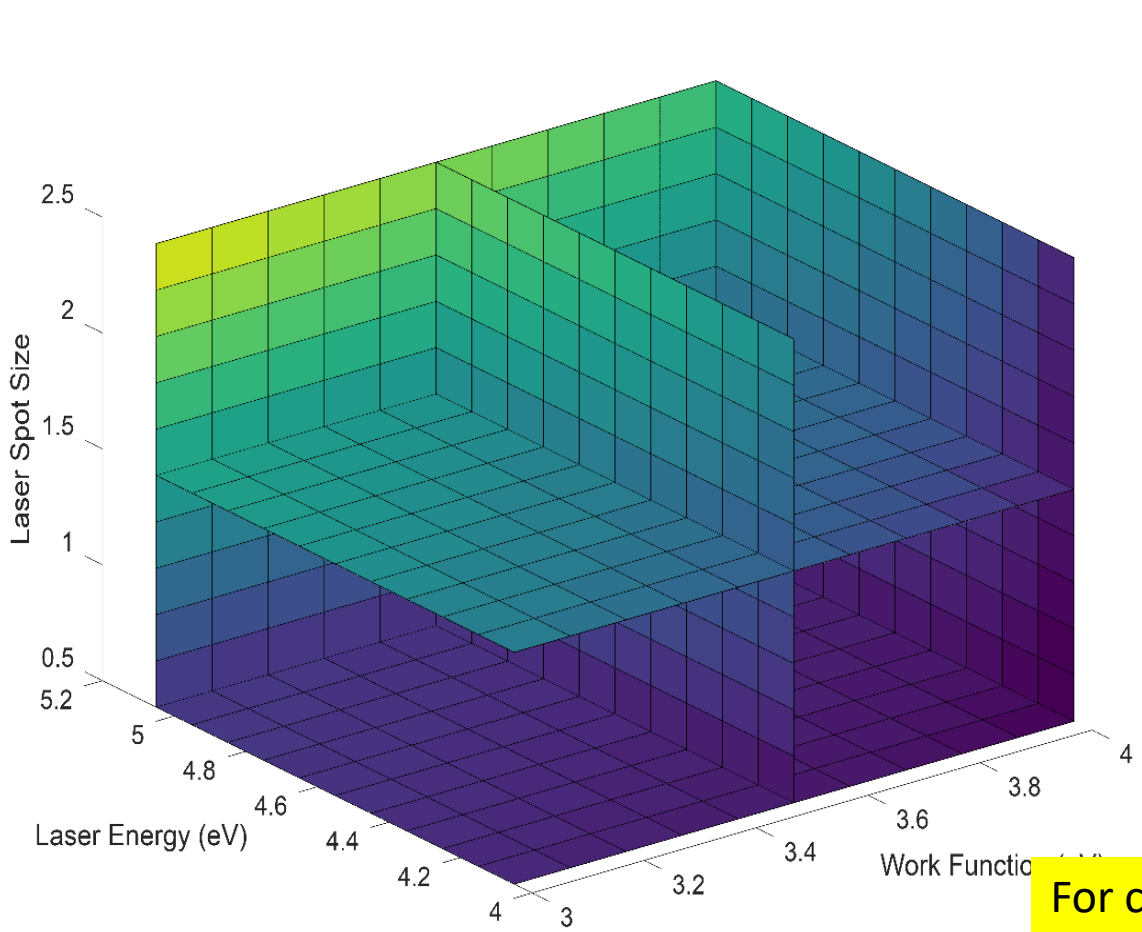


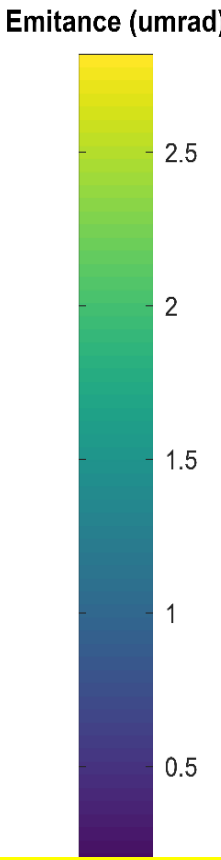
3D distributions – Fermi-Dirac

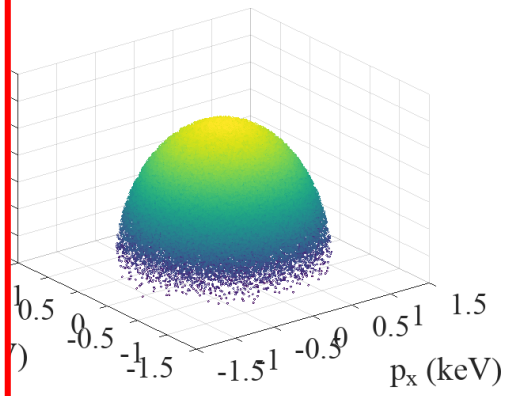
- V
- C
- A
  r
- T
- T
- A
- R
- *E*

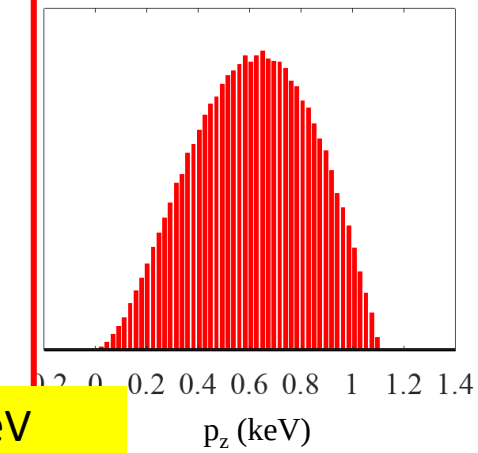


Dime	
p_x, p_y, p_z	Dist_









For cupper ϕ_0=4.25 eV
For CsTe ϕ_0=3.5 eV

074201 (2009)

Conclusion

- The generator is capable to generate all distributions which ASTRA generador does and more..
  - Any distribution can be generated quasi randomly
- There are some statistical math behind the correlated distributions like
  - distributions on disc, filling sphere surface and filling inside sphere for a given statistical distribution
- One can choose sequences as
  - `rand_genertor='sobol', (halton, reversehalton,..)`
  - if `noise_reduc=F` random numbers are created with linear division..
- There are few items needs to be improved/developed
  - Some issues with parsing input file like avoiding commanded out notes..
  - We plan to implement user defined PDF as input which might be useful for temporal distribution.