# 3D PET Image reconstruction based on MLEM algorithm

Zbigniew Rudy
Artur Słomski

**Computational science** is the study and implementation of numerical algorithms to solve problems for which a **quantitative theory** exists.

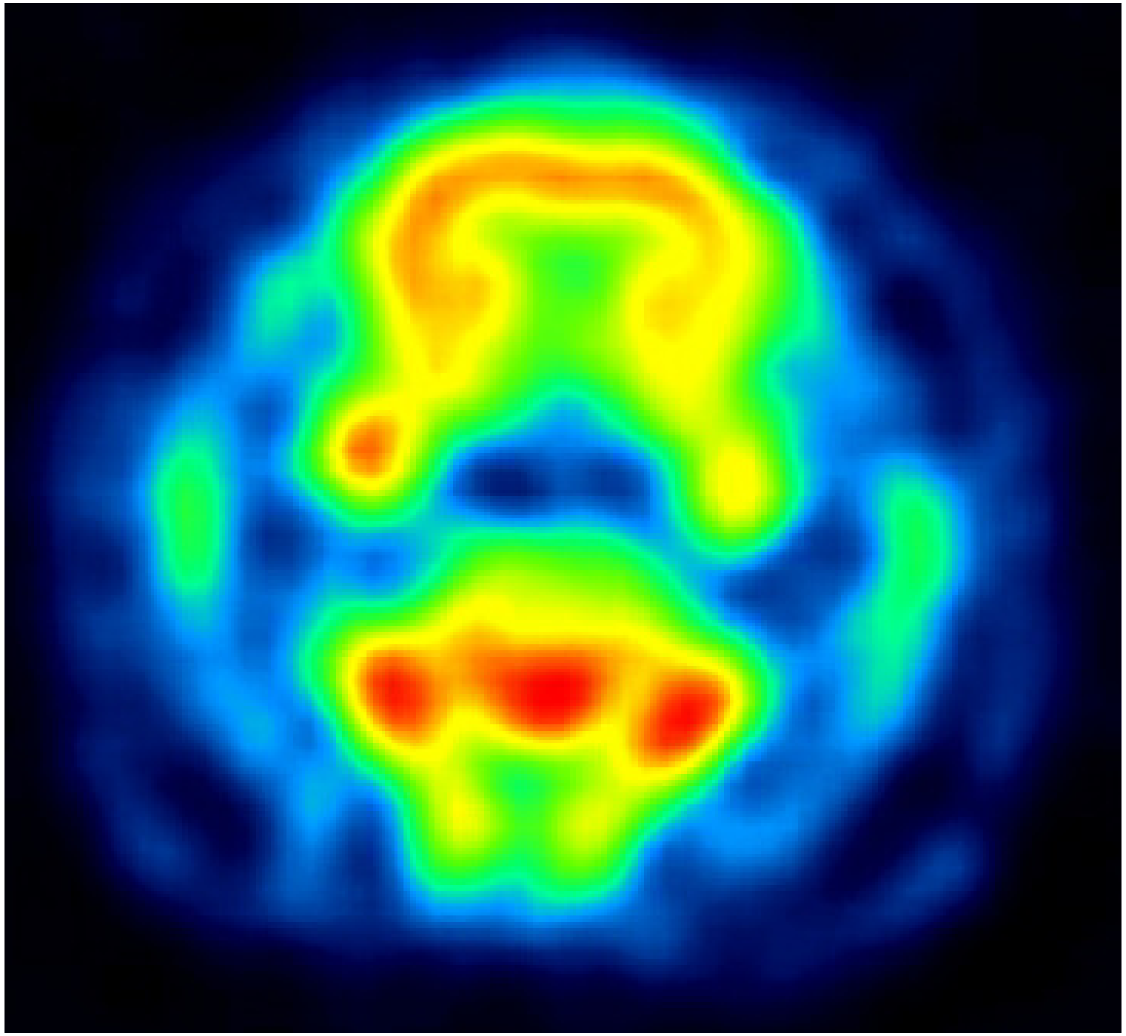
**Broad class of problems;** it is an essential component of modern research in different disciplines: accelerator physics, astrophysics, fluid mechanics,lattice field theory,plasma physics, simulations of physical systems, protein structure prediction….

**Subset of problems: boundary conditions**
(in particular in physics)

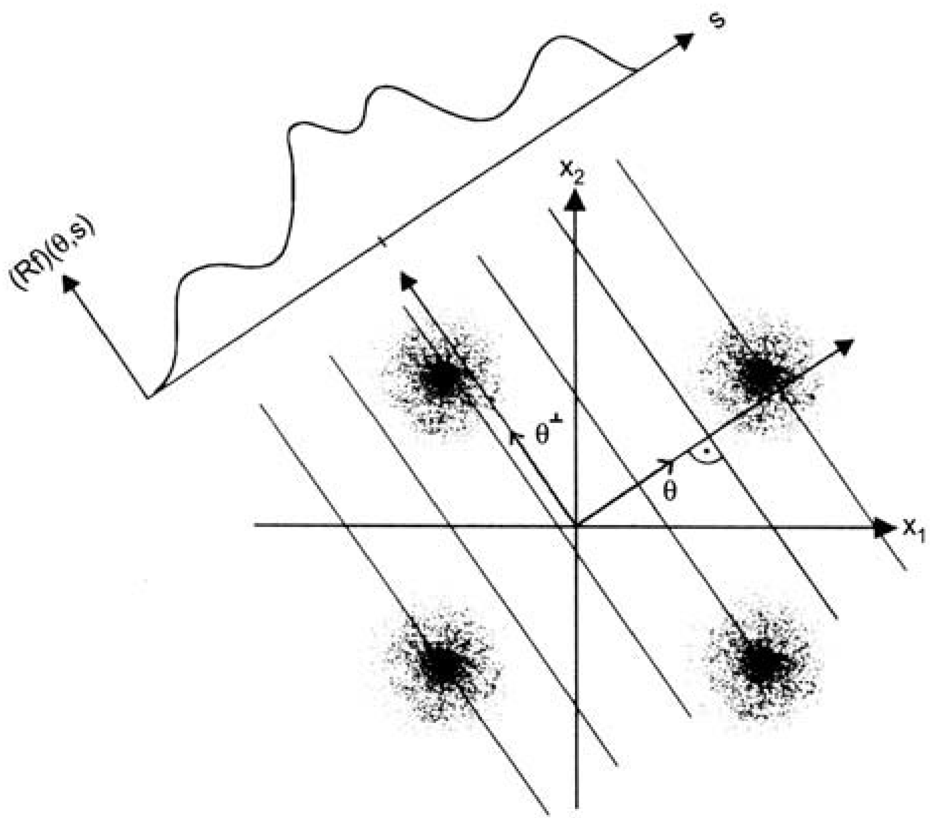
Like Laplace equation

$$\frac{\partial^2}{\partial x^2}u(x,y,z) + \frac{\partial^2}{\partial y^2}u(x,y,z) + \frac{\partial^2}{\partial z^2}u(x,y,z) = 0$$

Boundary condition could be ill-posed !



Task: (defined here for two-dimensional radiation space):
How to determine function f(x1, x2)
( f(x1, x2) it is radiation intensity)
from projection R(θ, s) ?



Austrian mathematician Johann Radon (1917) proved that from projections one can reconstruct radiation intensity (problem is well-posed).

However, the solution does not have a closed-form expression. Numerical methods are required.

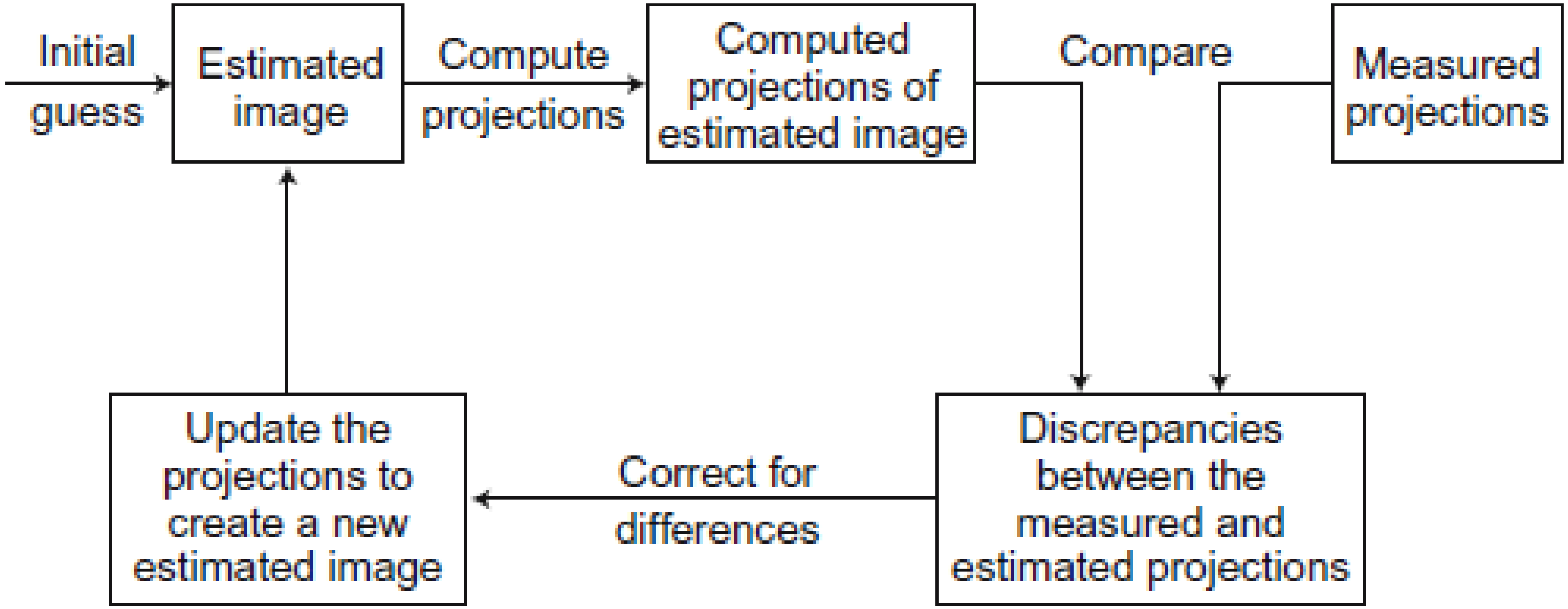
Modern approach: iterative algorithms derived from
**Maximum Likelihood Estimation Method (MLEM).**

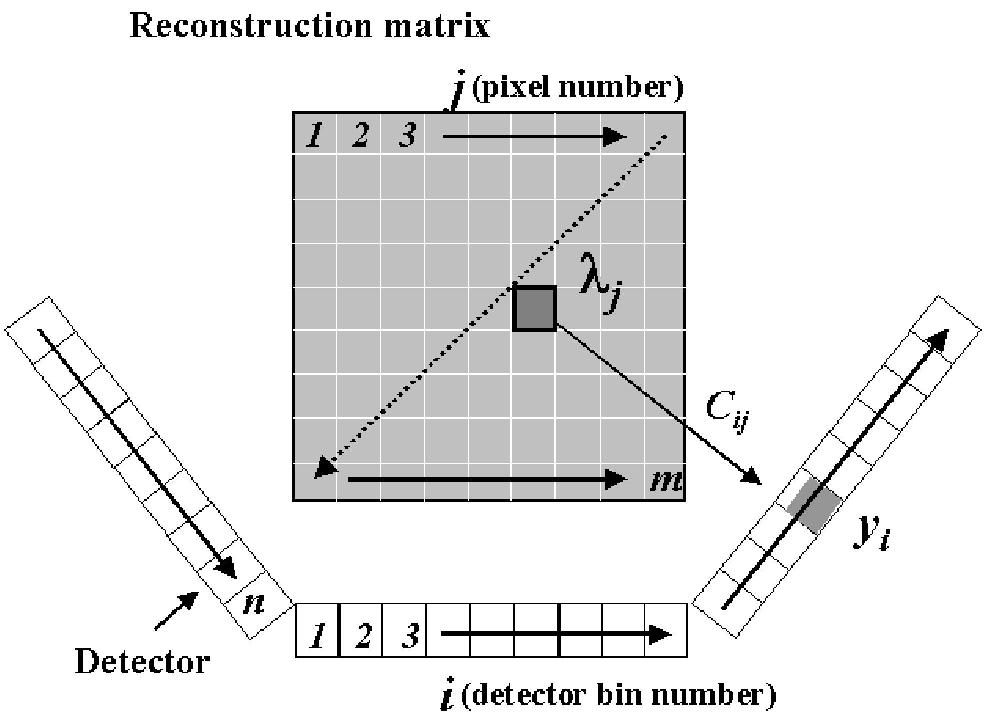
How to store the measured data ?

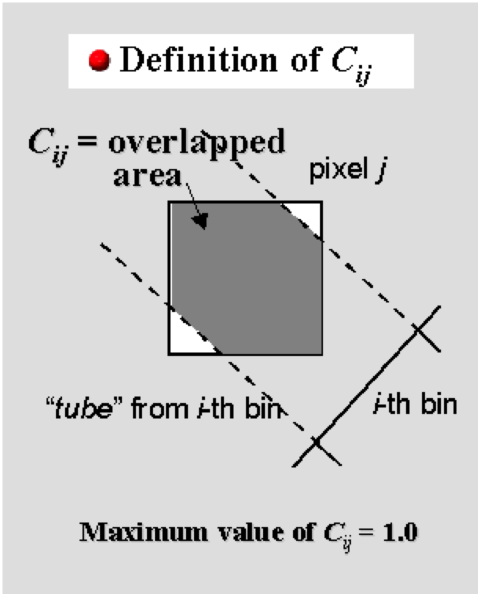
There are two classes of MLEM algorithms:

- Binned data   MLEM  iterative algorithm
- List mode data   MLEM iterative algorithm

Both follow the same strategy (on next page):


Initial
guess
Estimated image
Compute projections
Computed projections of estimated image
Compare
Measured projections
Update the projections to create a new estimated image
Correct for differences
Discrepancies between the measured and estimated projections




**Emission space** (index j) (three-dimensional)

**Projection space** (index i) (four-dimensional even without TOF…)

# MLEM algorithm (for binned data)

$$\lambda_j^{k+1} = \frac{\lambda_j^k}{\sum_{i=1}^{n} C_{ij}} \sum_{i=1}^{n} \frac{C_{ij}\, y_i}{\sum_{j=1}^{m} C_{ij}\, \lambda_j^k}$$

$\lambda_j^k$ – value of pixel j for k iteration
k – iteration number
j – pixel number
i – projection bin number
$C_{ij}$ – probability of detecting an emission from the pixel j in projection bin i

# Binned data MLEM algorithm

1. probability $C_{ij}$

2. initial values $\lambda_j^0$

3. forward projection $p_i = \sum_{j=1}^{m} C_{ij} \lambda_j^k$

# Binned data MLEM algorithm

4. comparison $$y_i^{'} = y_i / p_i$$

5. back projection $$x_j = \sum_{i=1}^{n} C_{ij} y_i^{'}$$

6. normalization $$x_j^{'} = x_j / \sum_{i=1}^{n} C_{ij}$$

# Binned data MLEM algorithm

7. updating $$\lambda_j^{k+1} = \lambda_j^k \, x_j^{'}$$

8. jump to point 3 (loop)

# List mode data MLEM algorithm

(iterative formula)

$$\lambda_l^{t+1} = \lambda_l^t \sum_{j=1}^{N} \frac{p(A_j / l)}{T \sum_{i=1}^{M} p(A_j / i)\, s_i\, \lambda_i^t}$$

$T$ - measurement time

$s_i$ - probability that photons emitted from pixel $i$ will be detected

$M$ - number of pixels in emission space

$N$ - number of events (of measurements)

$P(A_j / l)$ - probability that event generated in pixel $l$ leads to measurement $A_j$

Why do the iterative formulae for ***binned data*** and for ***list mode data*** look so similar ?

The distinction between ***binned data*** and ***list mode data*** disappears if the size of each bin is sufficiently small, because then the average number of counts in any bin becomes much less than one. Therefore with high probability every bin contains either 0 or 1 count; ***list mode data*** structure is formed !

# Binned data MLEM 3D

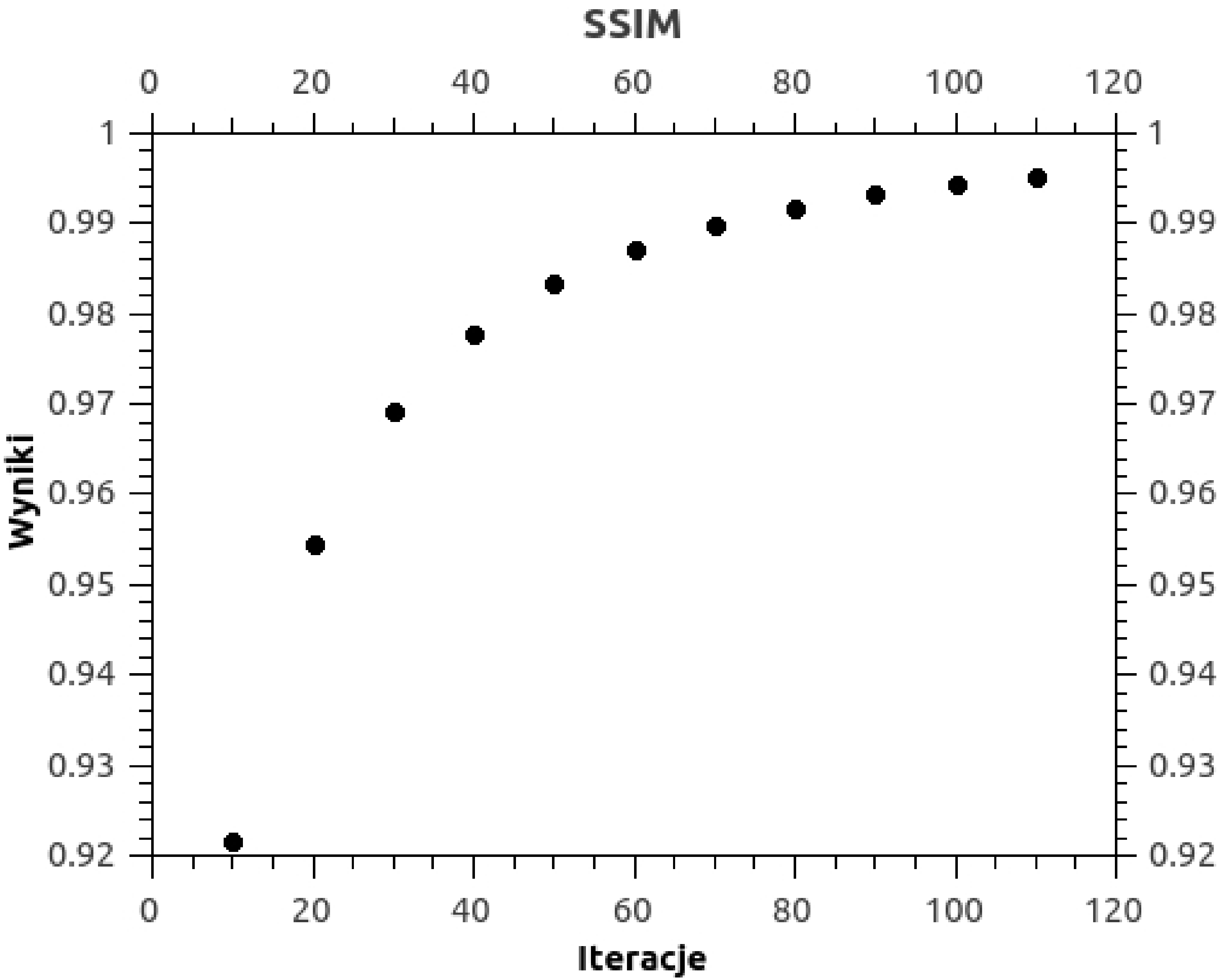
Measured gamma quanta in a Cartesian coordinate system (x, y, z)

↓ Transformation

Gamma quanta trajectory (LOR) stored/binned in a projection space system (coordinates r, θ, φ, sign wekx, sign weky, sign wekz)

r – distance from the origin of the coordinate system to LOR
θ - angle between LOR and positive half of axis OZ,
φ- angle between LOR projection onto XY plane and negative half of axis OX,
sign wekx – sign of a component x of distance vector r
sign weky – sign of a component y of distance vector r
sign wekz – sign of a component z of distance vector r

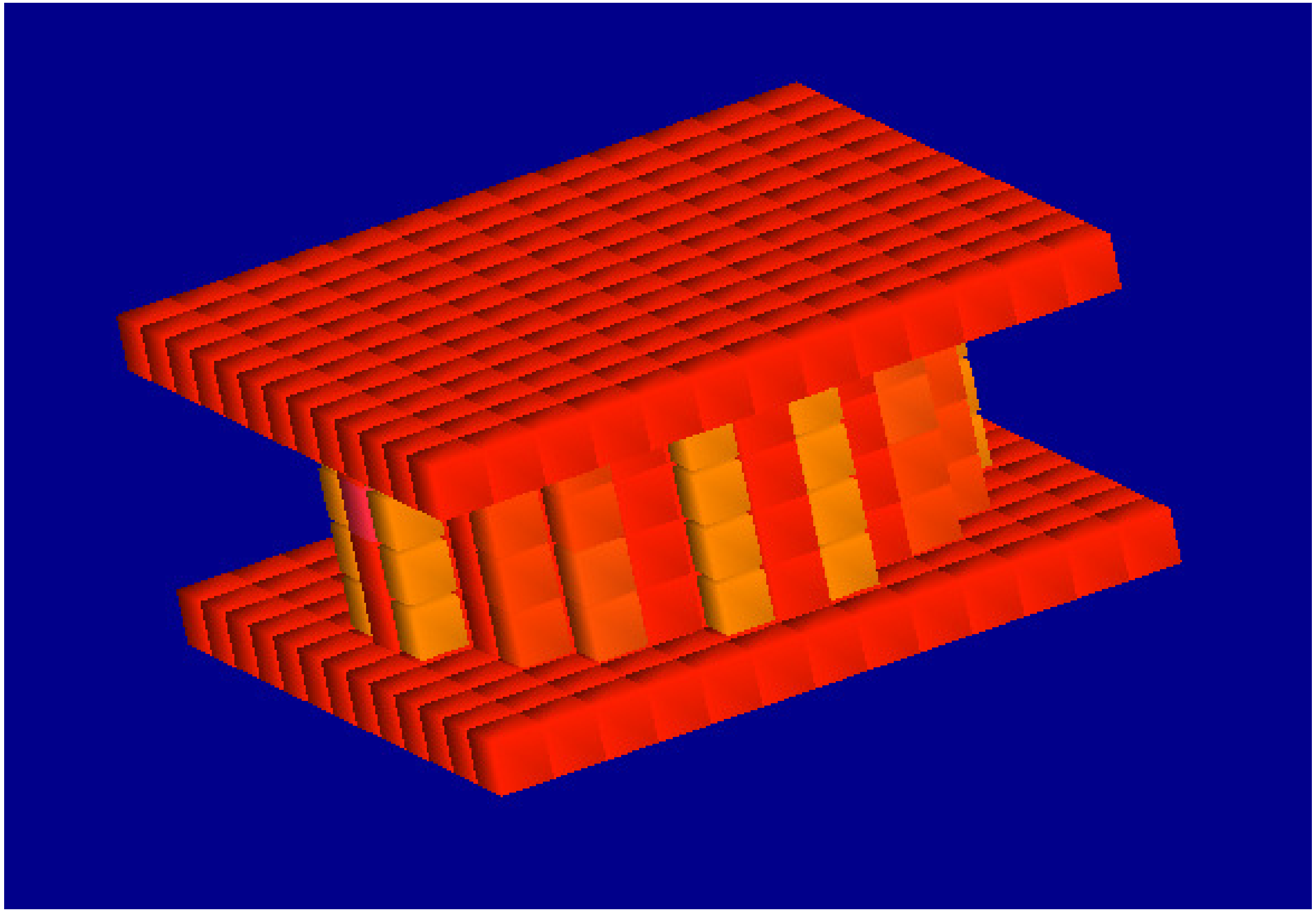
# Algorithm convergence (**S**tructural **SI**milarity **M**etric)

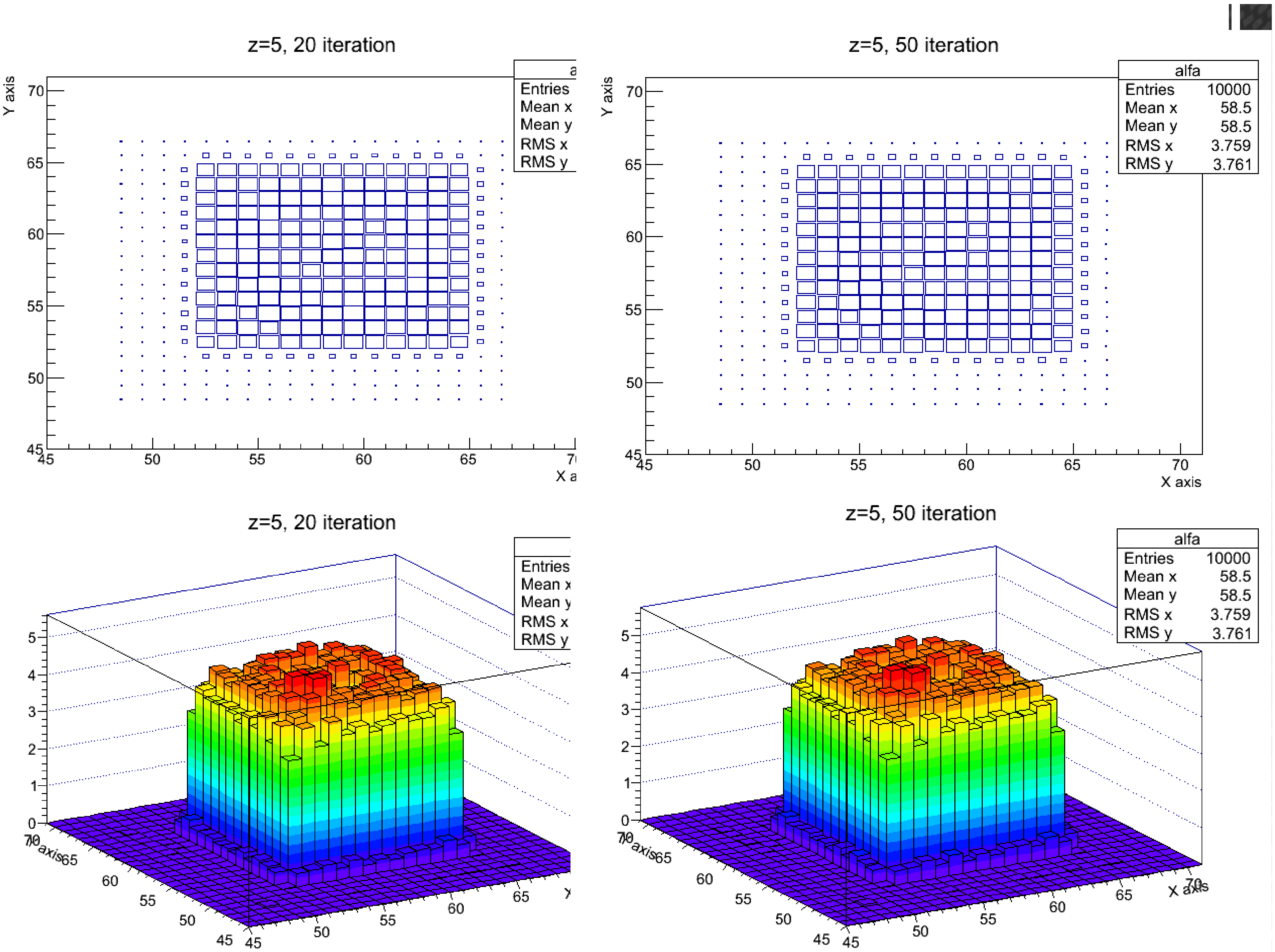




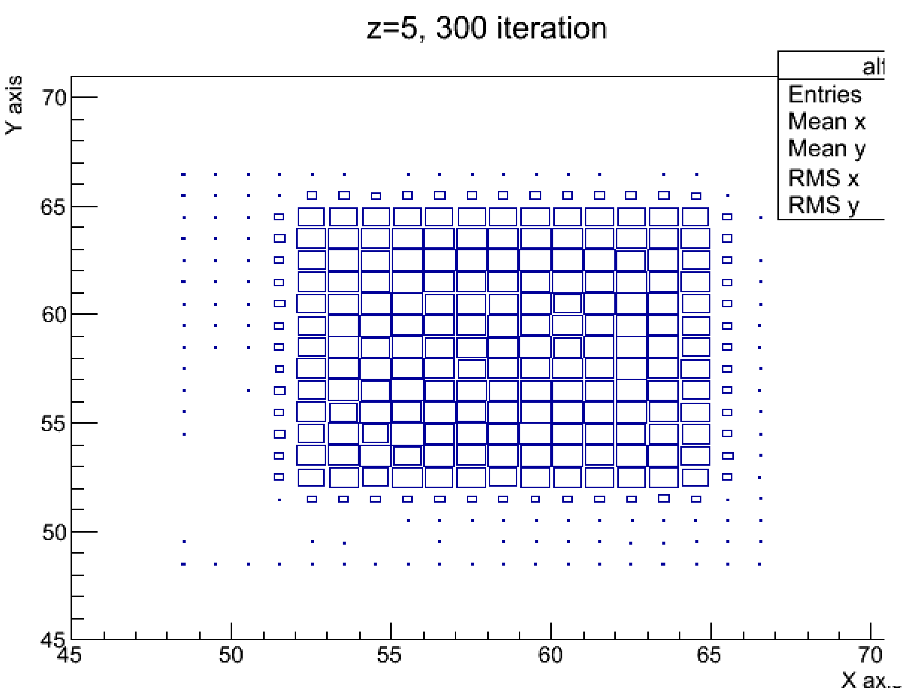

Wiktor Czosnowski

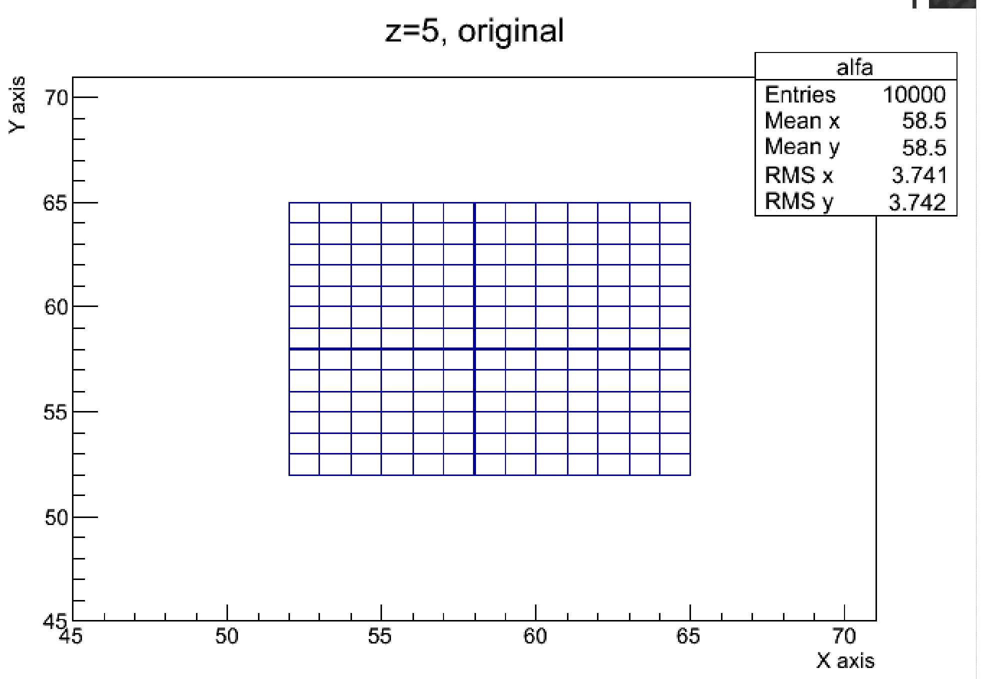
# Examples of algorithm results

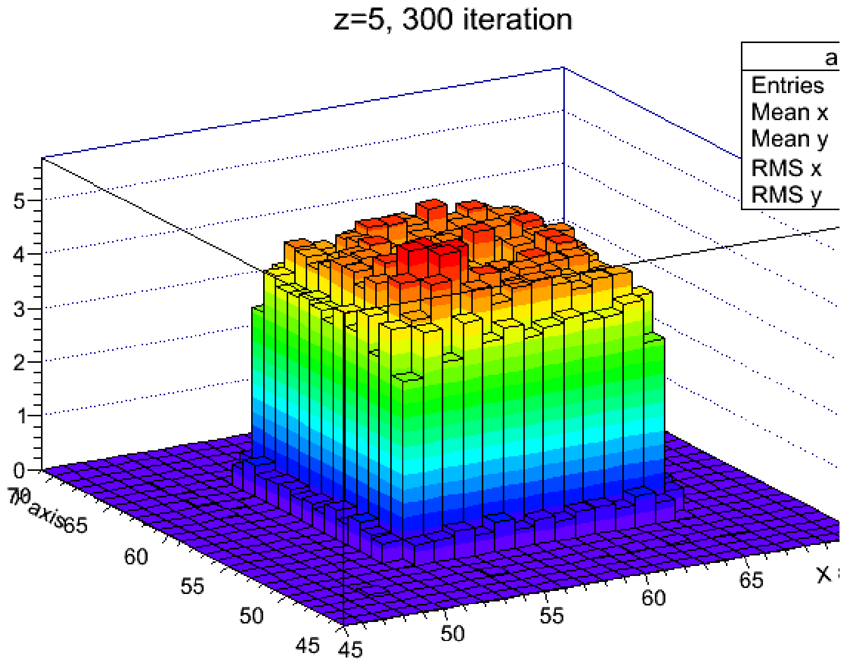


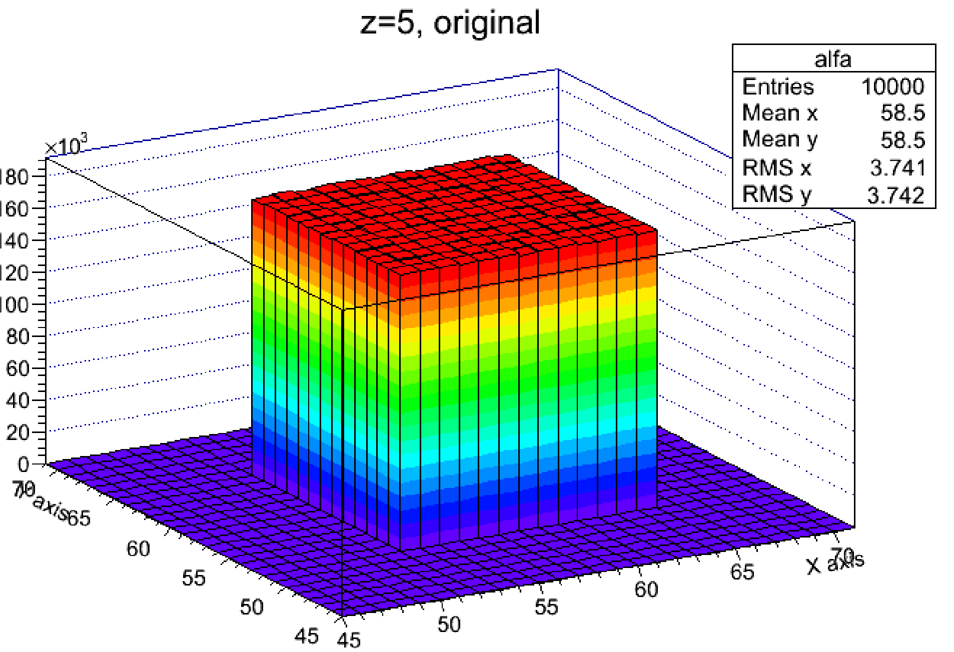


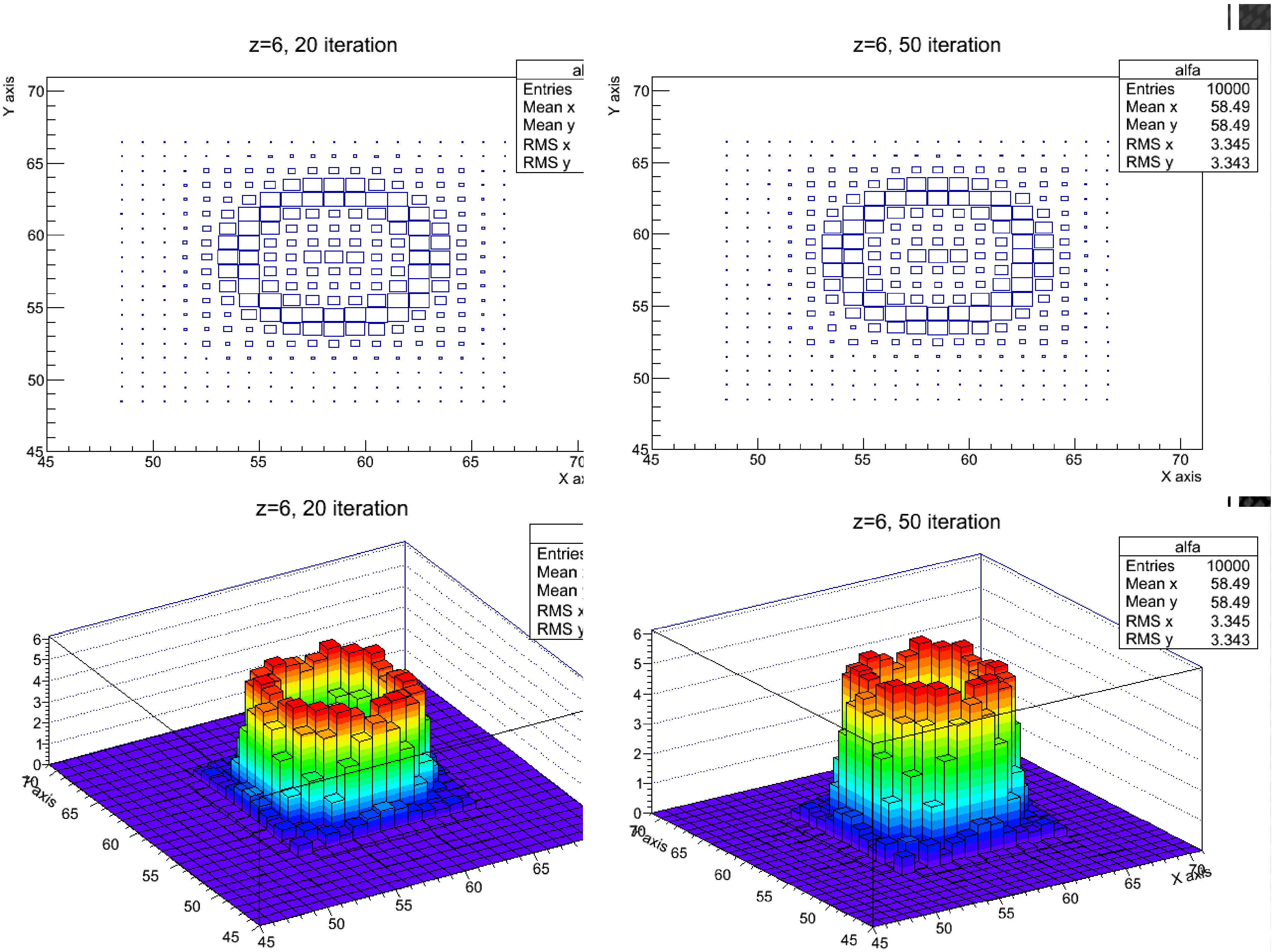
z=5, 20 iteration
Y axis
X a
Entries
Mean x
Mean y
RMS x
RMS y
z=5, 50 iteration
alfa
Entries 10000
Mean x 58.5
Mean y 58.5
RMS x 3.759
RMS y 3.761
Y axis
X axis
z=5, 20 iteration
Entries
Mean x
Mean y
RMS x
RMS y
z=5, 50 iteration
alfa
Entries 10000
Mean x 58.5
Mean y 58.5
RMS x 3.759
RMS y 3.761
X axis


z=5, 300 iteration
alf
Entries
Mean x
Mean y
RMS x
RMS y
Y axis
X ax.
45
50
55
60
65
70

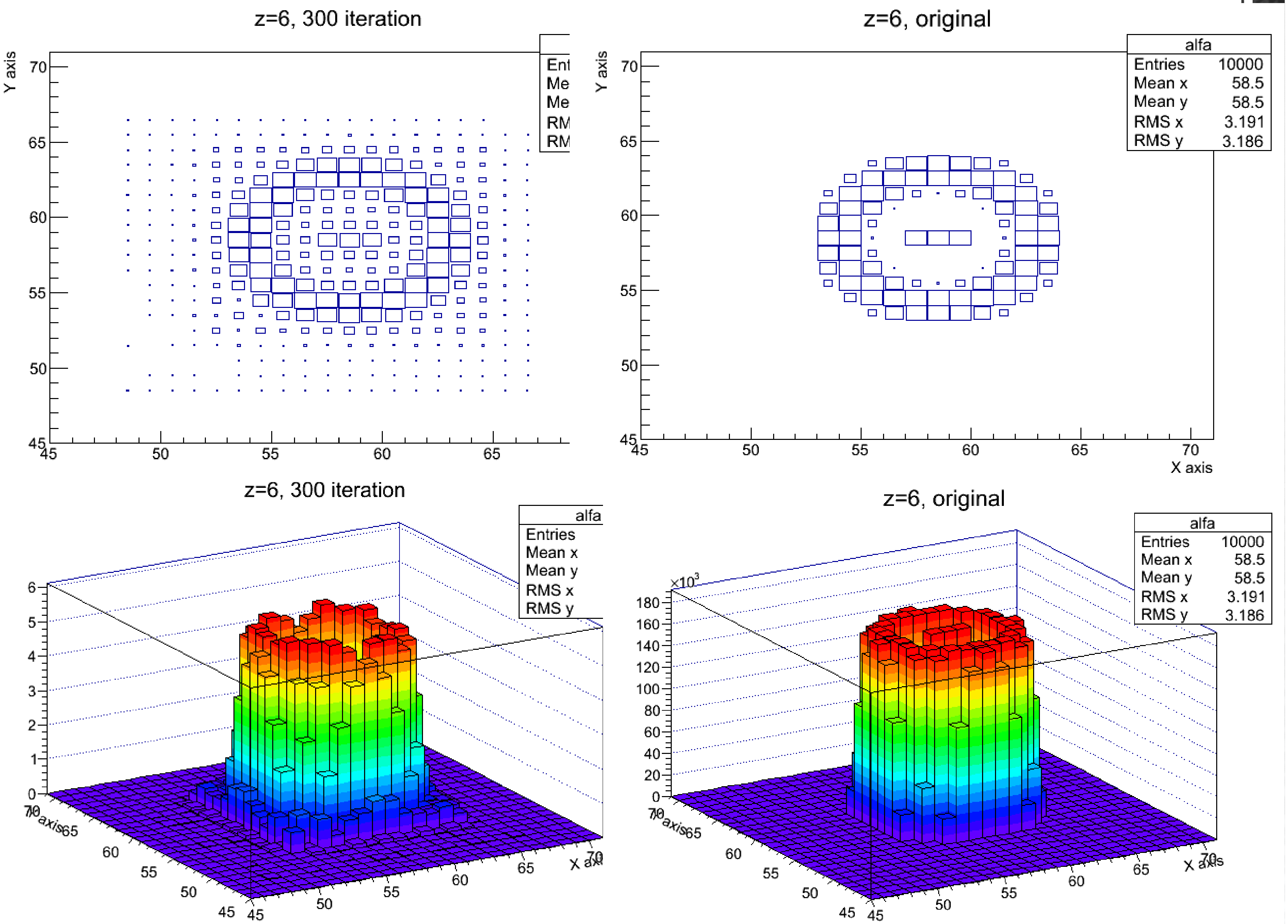

z=5, original
alfa
Entries 10000
Mean x 58.5
Mean y 58.5
RMS x 3.741
RMS y 3.742
Y axis
X axis
45
50
55
60
65
70


z=5, 300 iteration
a
Entries
Mean x
Mean y
RMS x
RMS y
0
1
2
3
4
5
45
50
55
60
65
70


z=5, original
alfa
Entries 10000
Mean x 58.5
Mean y 58.5
RMS x 3.741
RMS y 3.742
×10³
0
20
40
60
80
100
120
140
160
180
X axis
45
50
55
60
65
70

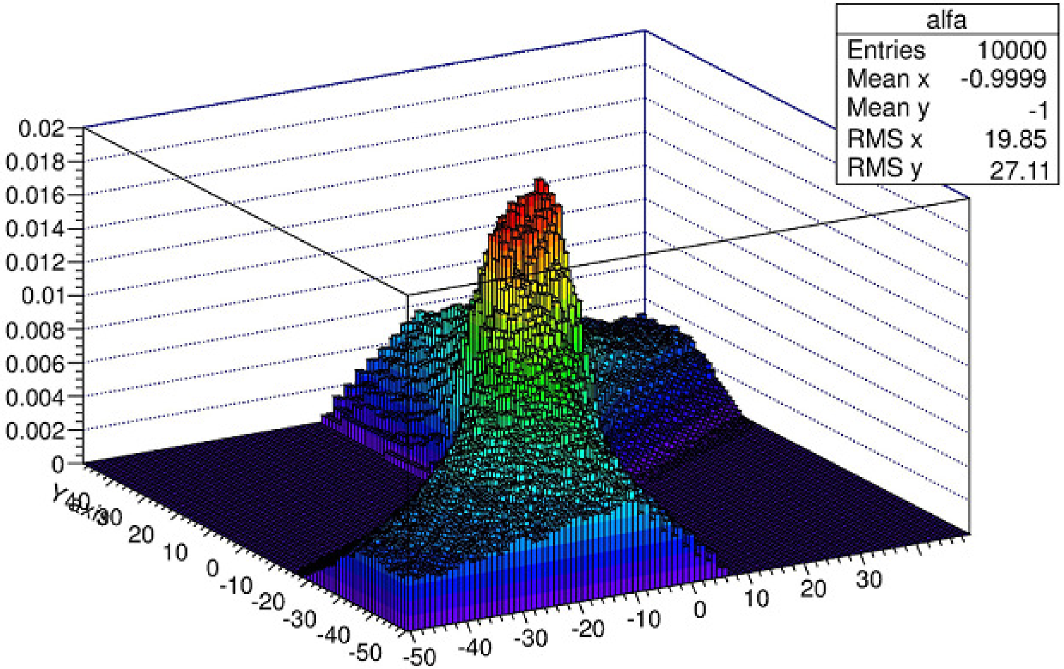


z=6, 20 iteration
Y axis
X axis
Entries
Mean x
Mean y
RMS x
RMS y
z=6, 50 iteration
alfa
Entries 10000
Mean x 58.49
Mean y 58.49
RMS x 3.345
RMS y 3.343
Y axis
X axis
z=6, 20 iteration
Entries
Mean
Mean
RMS x
RMS y
z=6, 50 iteration
alfa
Entries 10000
Mean x 58.49
Mean y 58.49
RMS x 3.345
RMS y 3.343
X axis

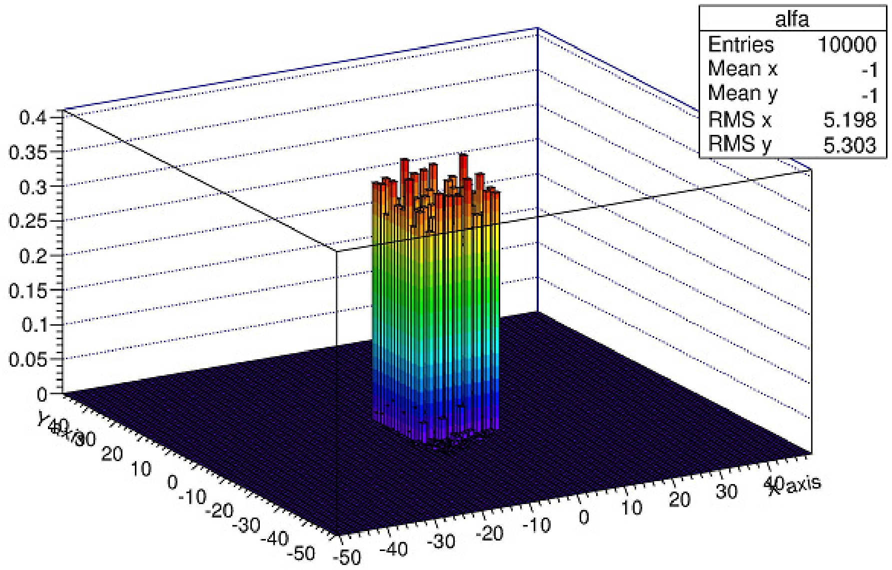


z=6, 300 iteration
z=6, original
alfa
Entries 10000
Mean x 58.5
Mean y 58.5
RMS x 3.191
RMS y 3.186
Y axis
X axis
z=6, 300 iteration
alfa
Entries
Mean x
Mean y
RMS x
RMS y
z=6, original
alfa
Entries 10000
Mean x 58.5
Mean y 58.5
RMS x 3.191
RMS y 3.186

# 81 points, 2 units distance using 60 degree area

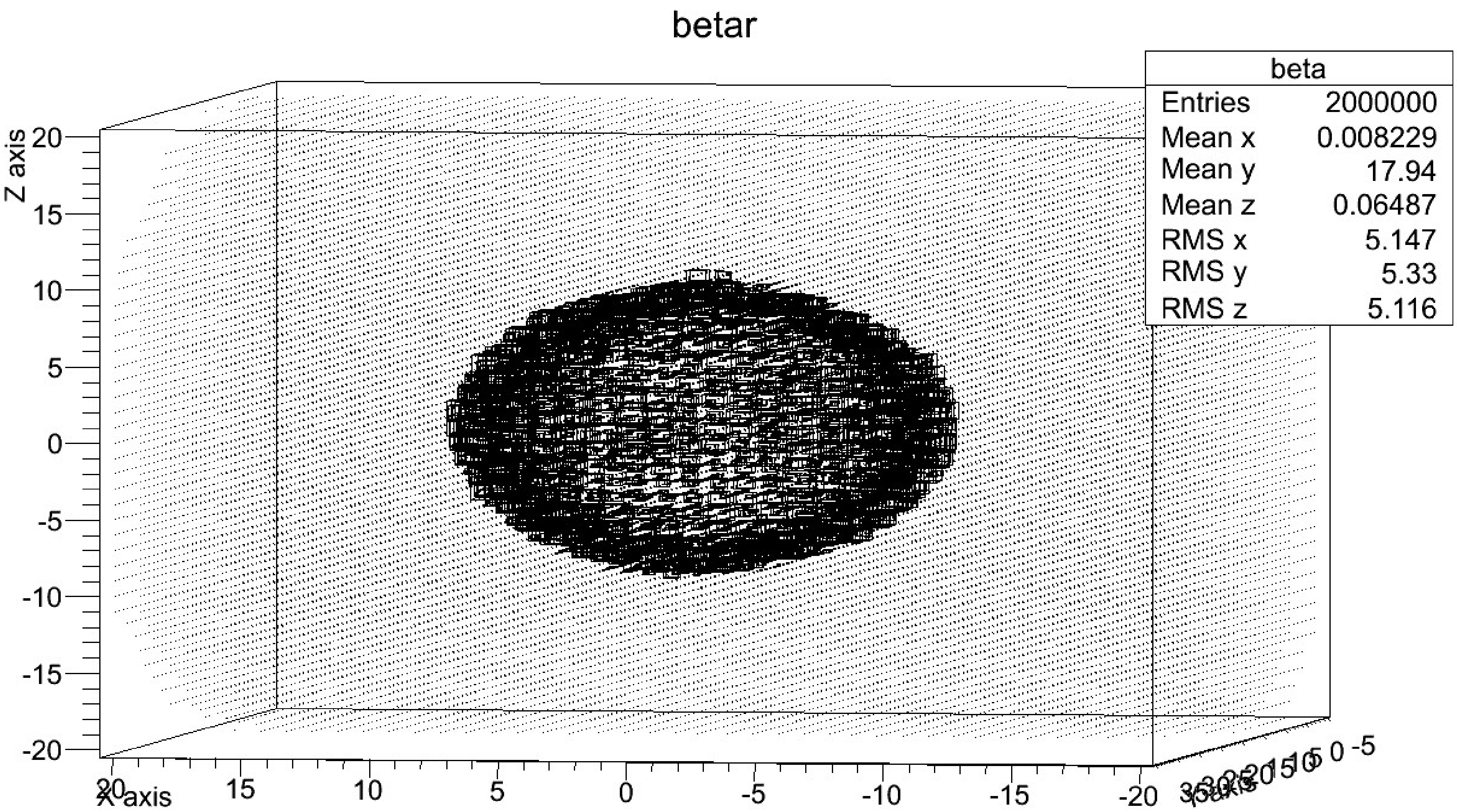


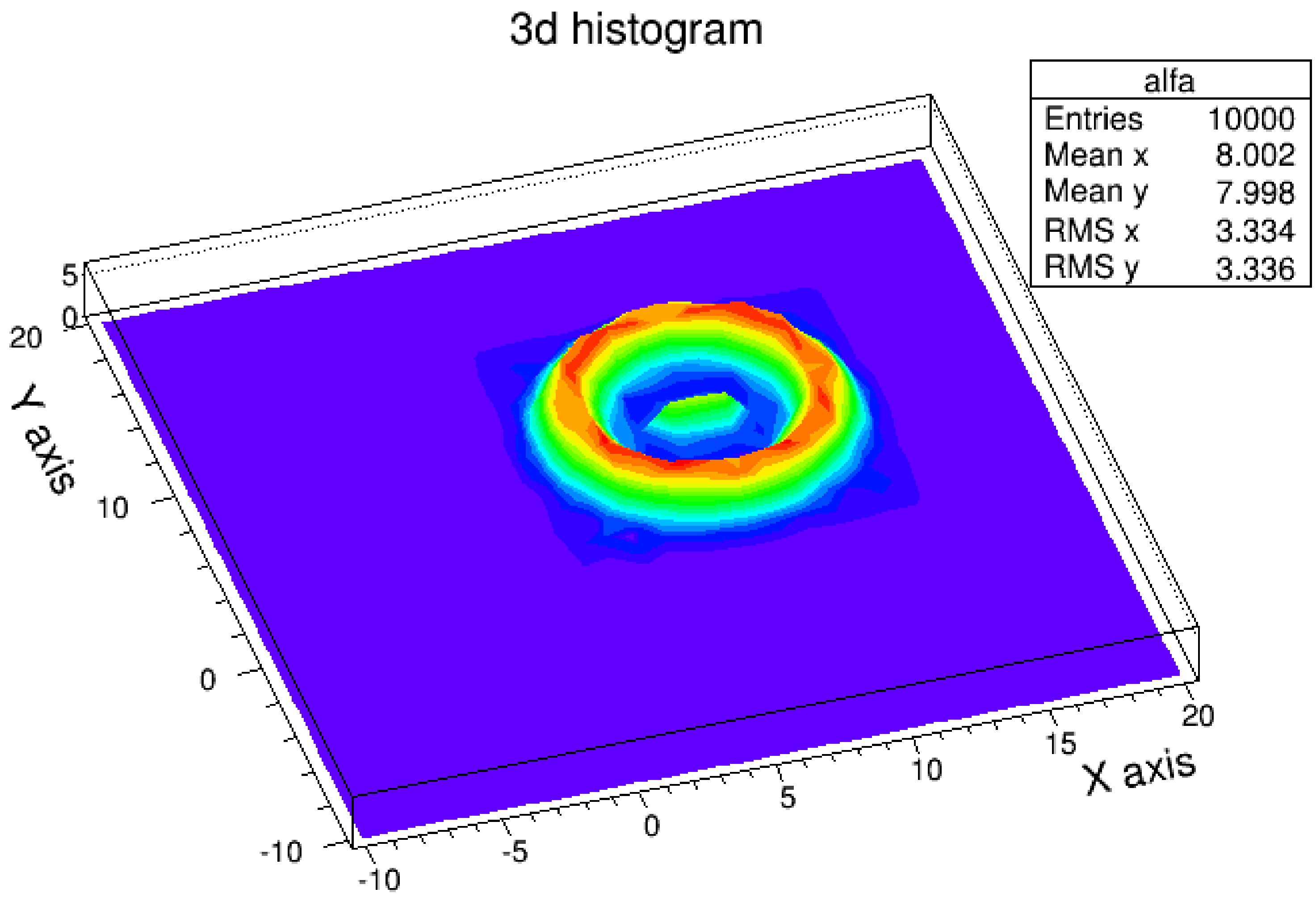

After 1 iteration




After 100 iterations

# Sphere after 60 iterations – 3D




3d histogram
alfa
Entries 10000
Mean x 8.002
Mean y 7.998
RMS x 3.334
RMS y 3.336
Y axis
X axis
5
0
20
10
0
-10
-10
-5
0
5
10
15
20

Thank You !

Q & A

The **structural similarity** (SSIM) metric is a method for measuring the similarity between two images.

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

## MLEM

**Cij calculation**
Probability → $C_{ij}$

**Initialization**
Initial values → $\lambda_j^0$ (>0)

**Forward projection** (Iteration start)
$$p_i=\sum_{j=1}^{m} C_{ij}\lambda_j^k$$

**Comparison**
$$y'_i = y_i/p_i$$

**Back projection**
$$x_j=\sum_{i=1}^{n} C_{ij}\,y'_i$$

**Normalization**
$$x'_j = x_j/\sum_{i=1}^{n} C_{ij}$$

**Update**
$$\lambda_j^{k+1} \leftarrow \lambda_j^k\, x'_j$$

Iteration

## ASIRT

**Cij calculation**
Probability → $C_{ij}$

**Initialization**
Initial values → $\lambda_j^0$ (>0)

**Forward projection**
$$p_i=\sum_{j=1}^{m} C_{ij}\lambda_j^k$$

**Comparison**
$$y'_i = (y_i - p_i)/\sum_{j=1}^{m} C_{ij}$$

**Back projection**
$$x_j=\sum_{i=1}^{n} C_{ij}\,y'_i$$

**Normalization**
$$x'_j = x_j/\sum_{i=1}^{n} C_{ij}$$

**Update**
$$\lambda_j^{k+1} \leftarrow \lambda_j^k + x'_j$$

Iteration

## MSIRT

**Cij calculation**
Probability → $C_{ij}$

**Initialization**
Initial values → $\lambda_j^0$ (>0)

**Back projection**
$$x_{1j}=\sum_{i=1}^{n} \left(C_{ij}\,y_i/\sum_{j=1}^{m} C_{ij}\right)$$

**Forward projection**
$$p_i=\sum_{j=1}^{m} C_{ij}\lambda_j^k$$

**Back projection**
$$x_{2j}=\sum_{i=1}^{n} C_{ij}\,y'_i$$

**Comparison**
$$x'_j = x_{1j}/x_{2j}$$

**Update**
$$\lambda_j^{k+1} \leftarrow \lambda_j^k\, x'_j$$

Iteration

Three types of iterative algorithms:

MLEM - maximum likelihood-expectation maximization

ASIRT - additive simultaneous iterative reconstruction technique

MSIRT - multiplicative simultaneous iterative reconstruction technique

Nice feature of presented iterative (update) equations is that positivity constraint is automatically satisfied (pixels in radiation space should not have intensity smaller than 0…)

Computational complexity of algorithm: for list mode data equals

(number of pixels) times (number of measured events).

For binned data: (number of pixels) times (number of bins).

Often the detection system provides multitude of parameters of measurements, like position coordinates, time measurement, angles, energy and so on. From these measurements usually a small set of parameters is used; one reduces the number of coordinates, because the number of bins increases exponentially ! …or…instead of binning the data one uses list mode data.