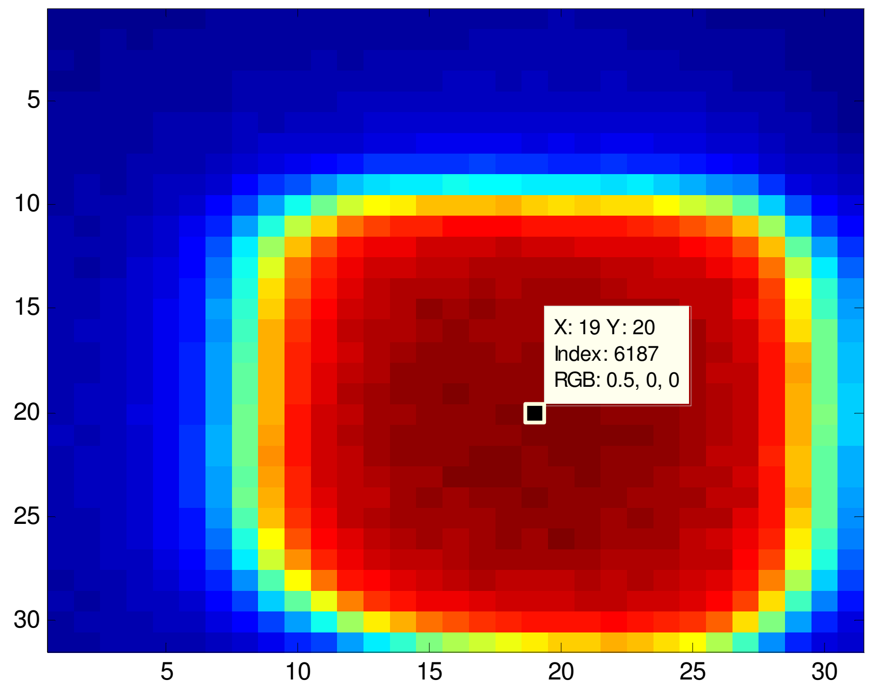
3D Medipix Testbeam Scatter fits for background subtraction

Aaron Mac Raighne

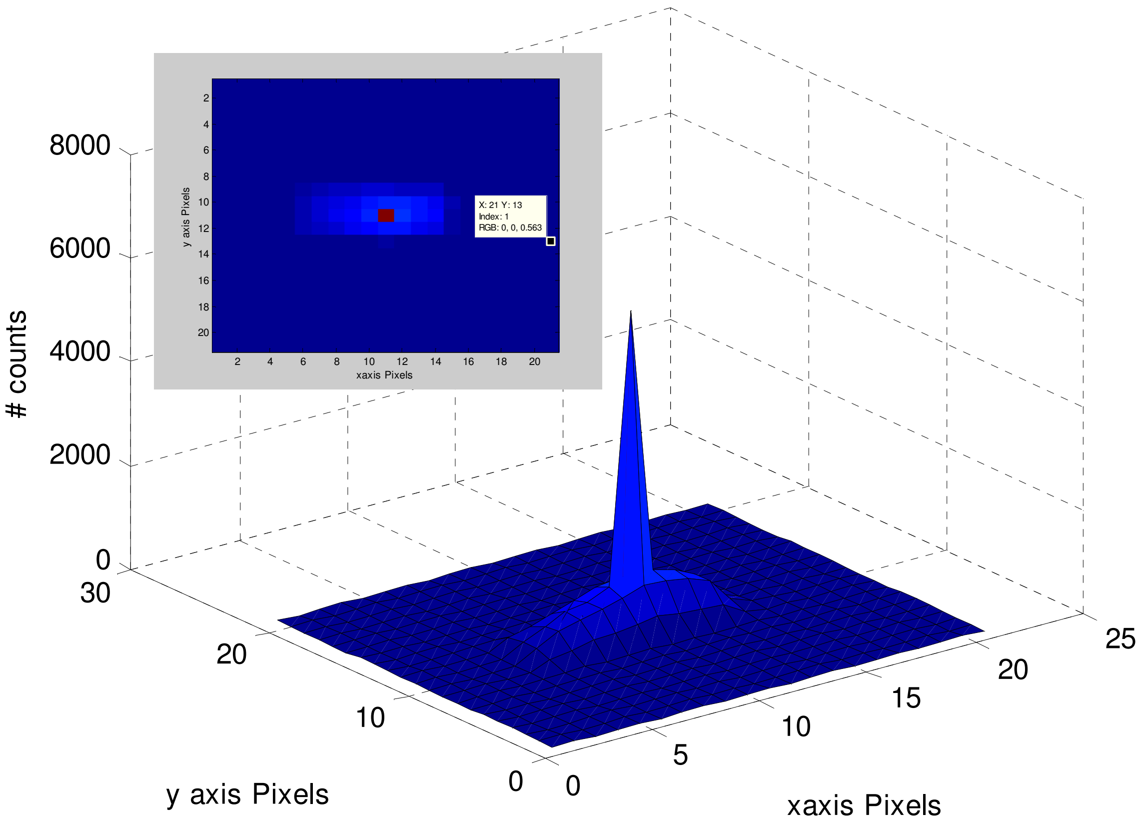
Using planar C4_1

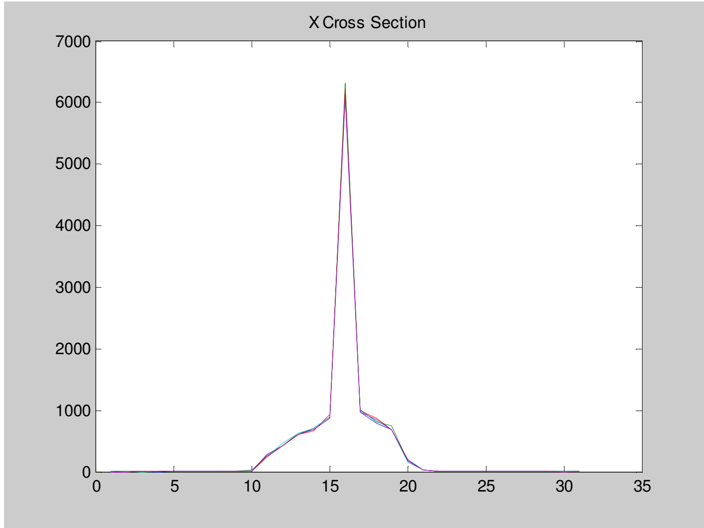
taking the 5 acquisition files from this position and using GuassFit.m found the x and y cross sections for these five files

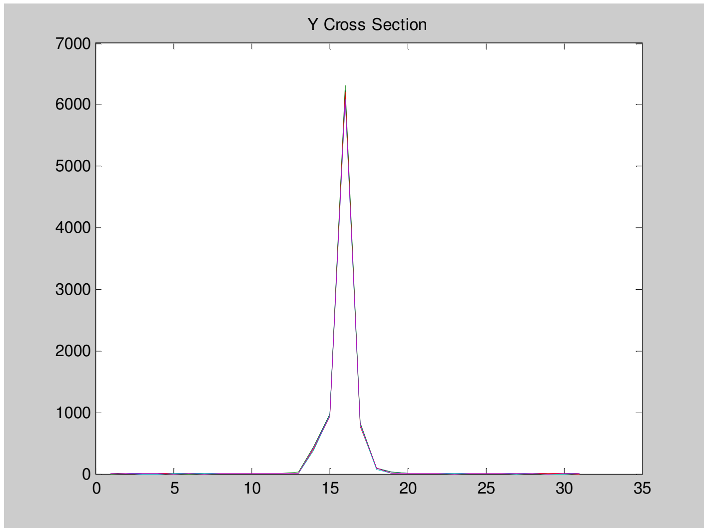


The fits must only be taken from the files which correspond to a beam in the centre of the pixel so that we can be sure that there are no charge sharing effects and the tail effects from the Gaussian of the 'focussed beam' are minimal

Example of the beam from the final aq file opened 1stPixelScan2889.txt
and the five cross sections in x and y

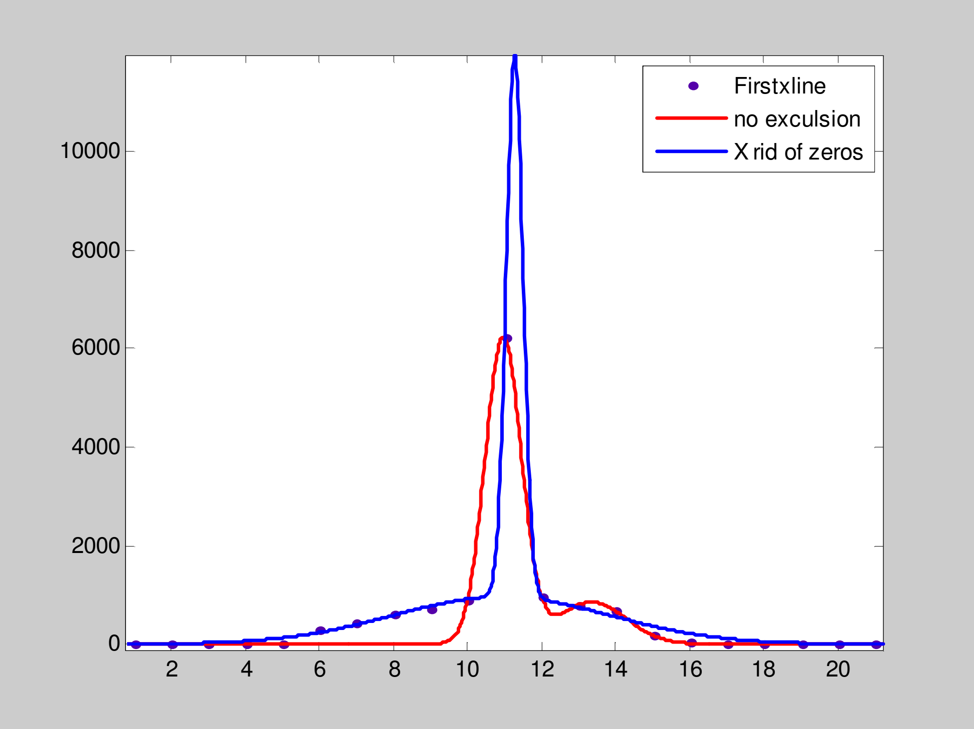






The problem with finding a fit for the centre is the lack of data-points, therefore I think this should be neglected.

Double Gaussian fit to the first of the aq. file cross section

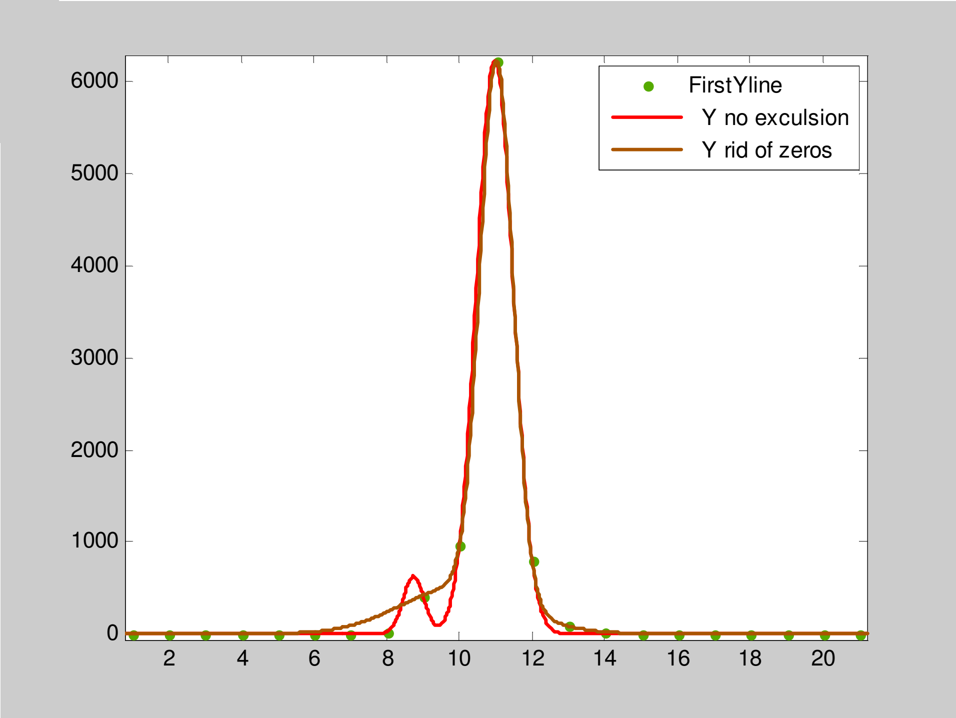


General model Gauss2:
    f(x) = a1*exp(-((x-b1)/c1)^2) + a2*exp(-((x-b2)/c2)^2)
Coefficients:
    a1 = 5823
    b1 = 11.02
    c1 = 0.6421
    a2 = 506.2
    b2 = 9.992
    c2 = 2.263

Goodness of fit:
  SSE: 1.228e-023
  R-square: 1
  Adjusted R-square: NaN
  RMSE: NaN

General model Gauss2:
    f(x) = a1*exp(-((x-b1)/c1)^2) + a2*exp(-((x-b2)/c2)^2)
Coefficients (with 95% confidence bounds):
    a1 = 1.1e+004 (-3.215e+008, 3.215e+008)
    b1 = 11.28 (-3380, 3402)
    c1 = 0.3221 (-2441, 2442)
    a2 = 957.6 (558.2, 1357)
    b2 = 10.93 (10.22, 11.64)
    c2 = 4.161 (2.502, 5.821)

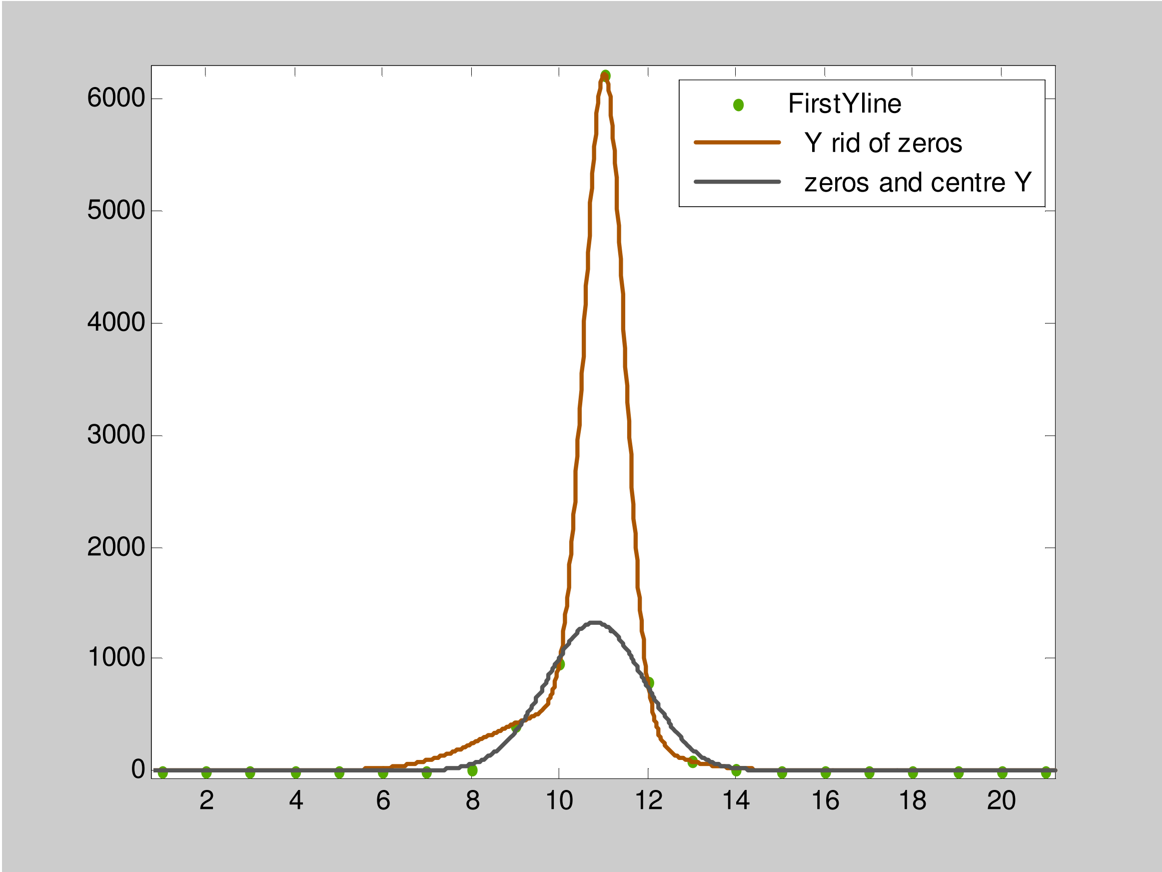
Goodness of fit:
  SSE: 5.691e+004
  R-square: 0.998
  Adjusted R-square: 0.9956
  RMSE: 119.3



This is the same as shown before but excluding the centre pixel and the zeros and fitting a single Gaussian to look at the scattering contribution alone

General model Gauss1:
f(x) = a1*exp(-((x-b1)/c1)^2)
Coefficients (with 95% confidence bounds):
a1 = 1324 (853.7, 1794)
b1 = 10.82 (10.58, 11.07)
c1 = 1.571 (1.119, 2.023)

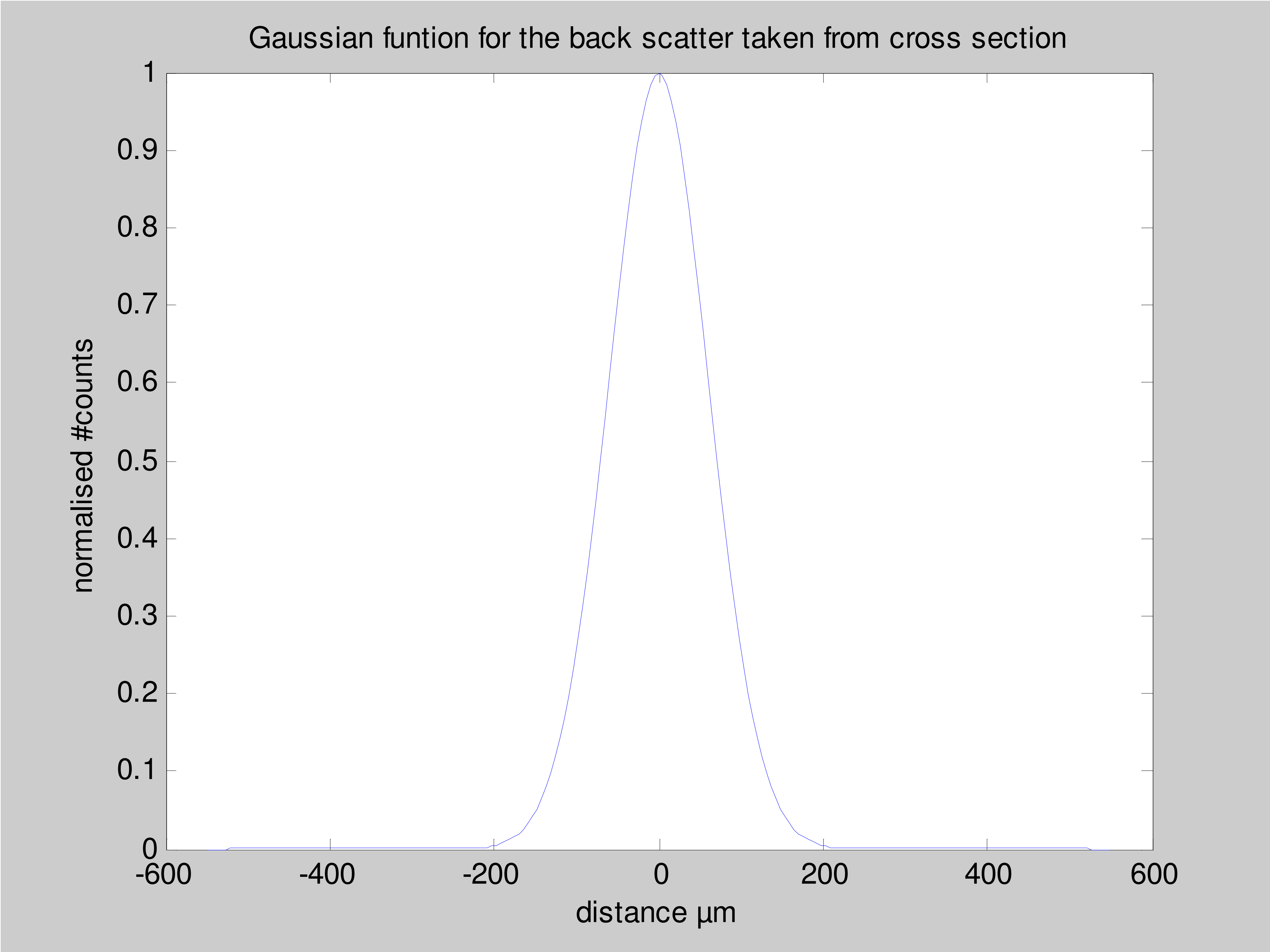
Goodness of fit:
SSE: 2.267e+004
R-square: 0.9739
Adjusted R-square: 0.9565
RMSE: 86.94

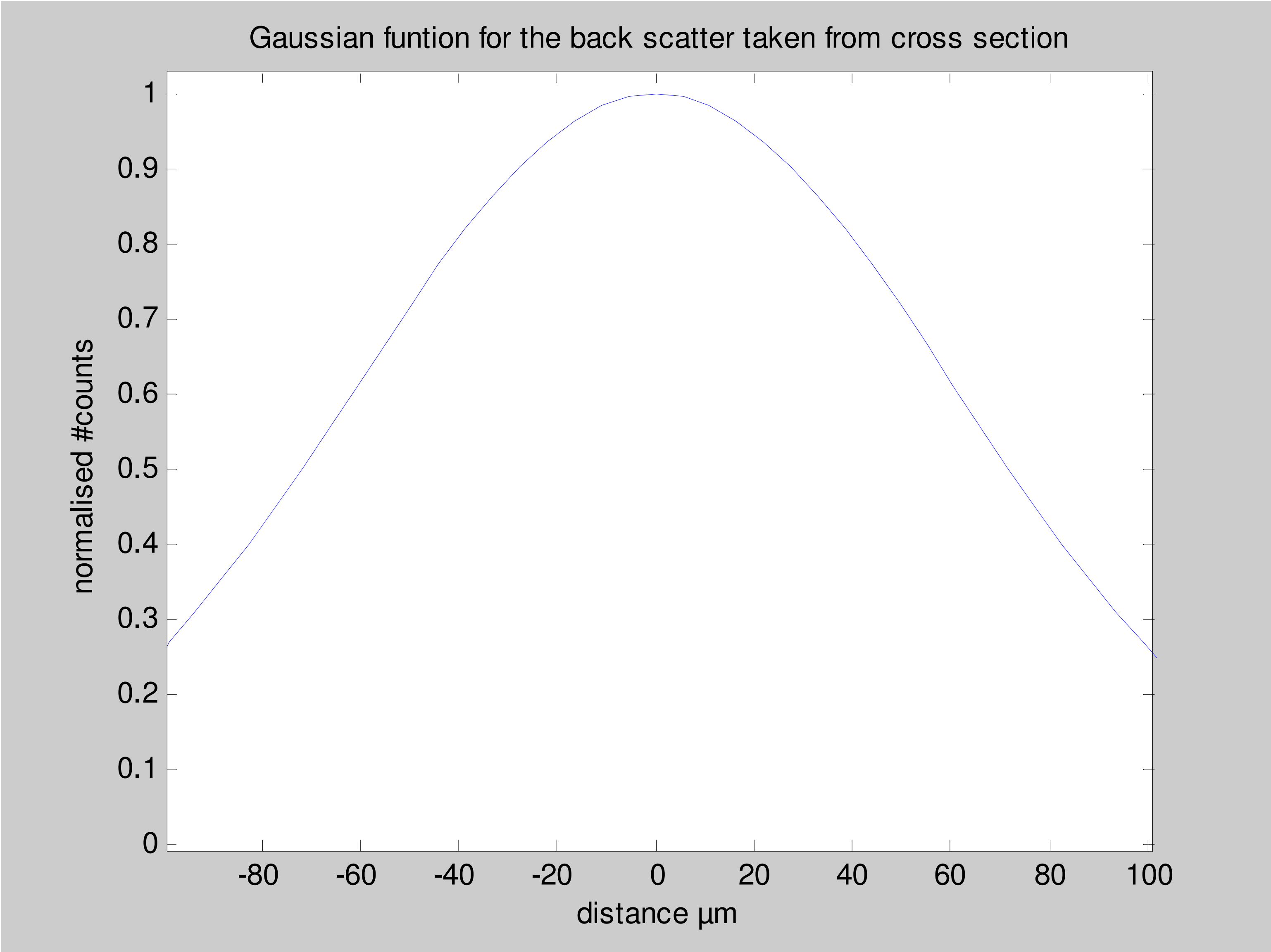


Compared to the double Gaussian

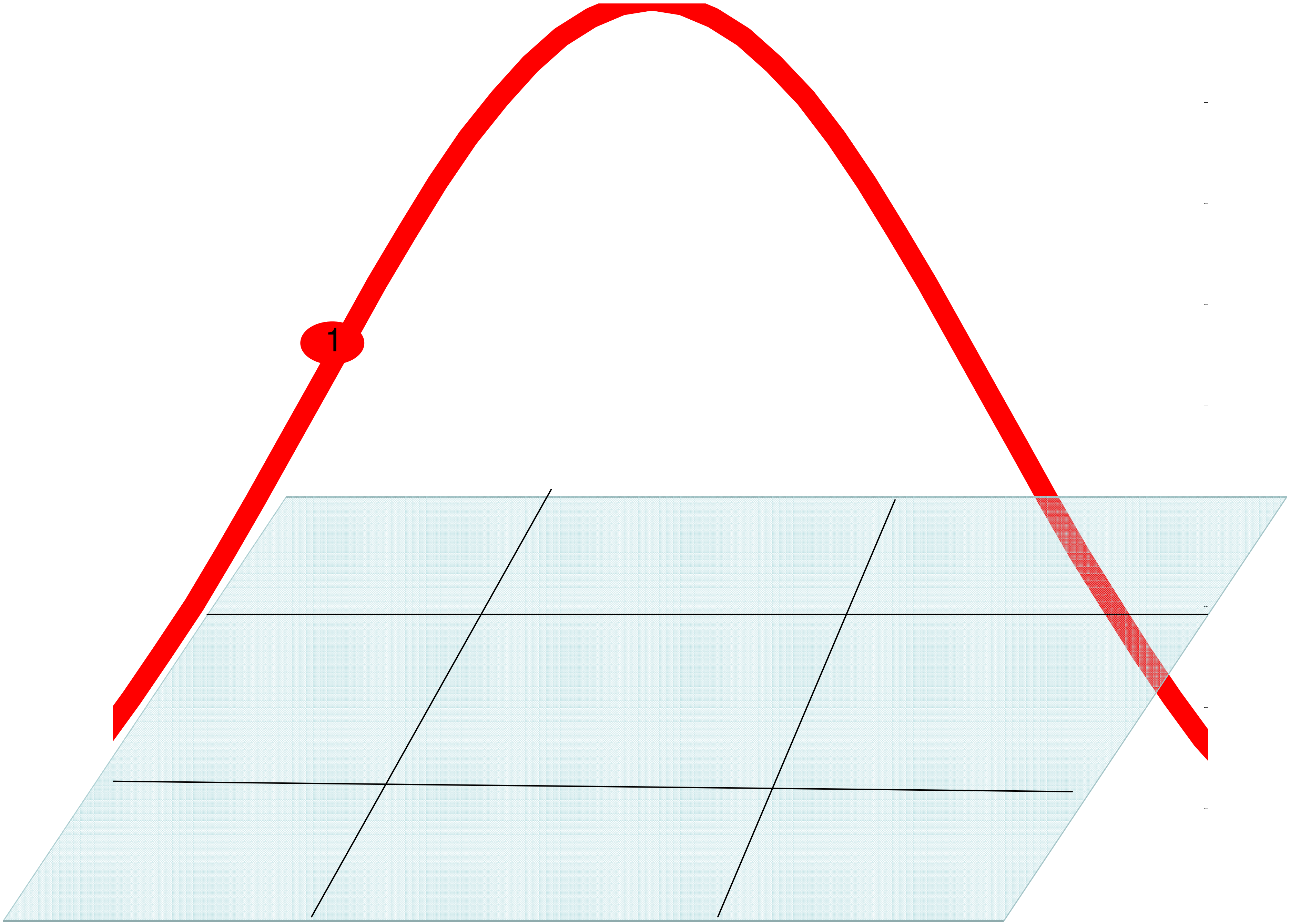
General model Gauss2:
f(x) = a1*exp(-((x-b1)/c1)^2) + a2*exp(-((x-b2)/c2)^2)
Coefficients:
a1 = 5823
b1 = 11.02
c1 = 0.6421
a2 = 506.2
b2 = 9.992
c2 = 2.263

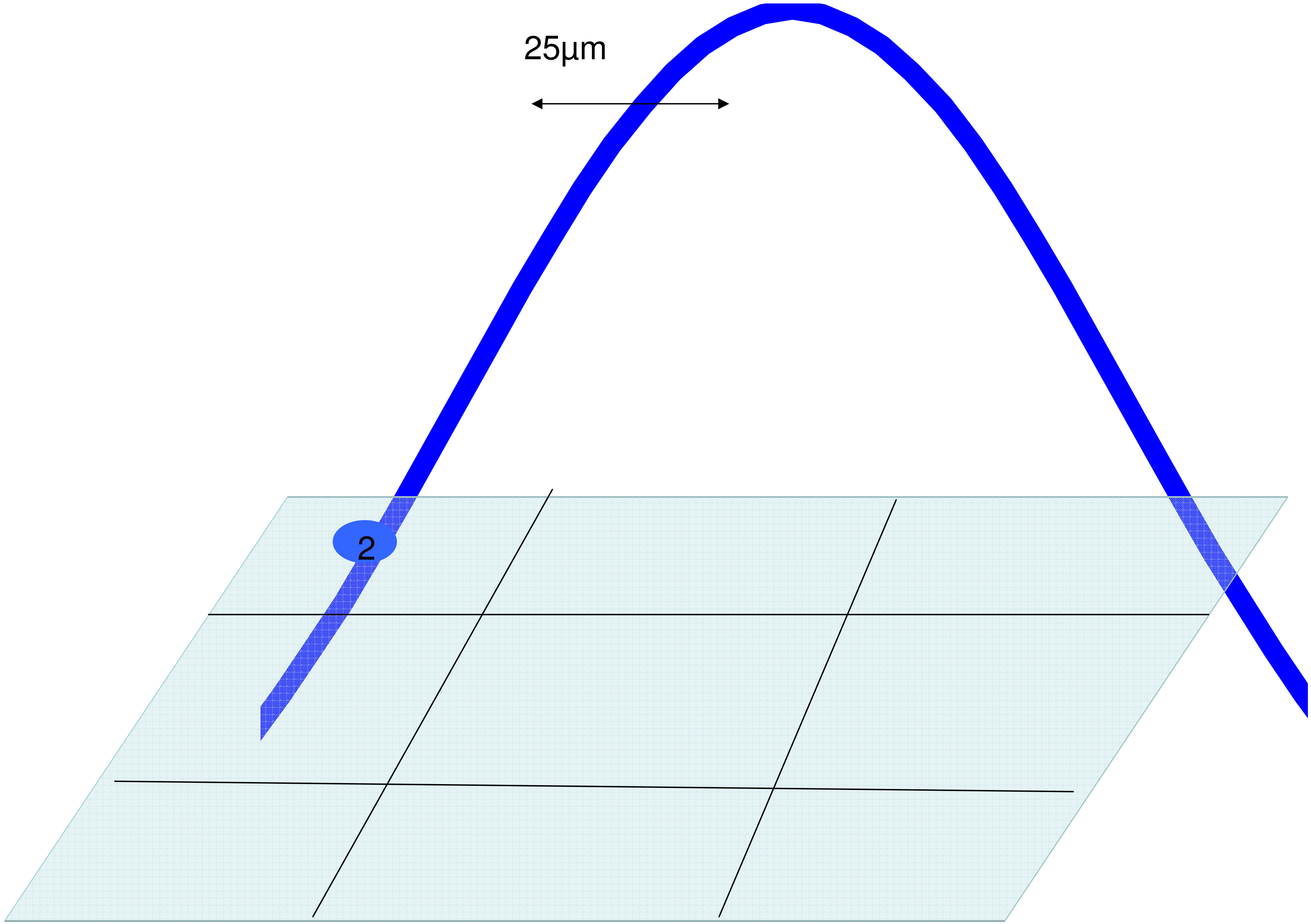
Goodness of fit:
SSE: 1.228e-023
R-square: 1
Adjusted R-square: NaN
RMSE: NaN

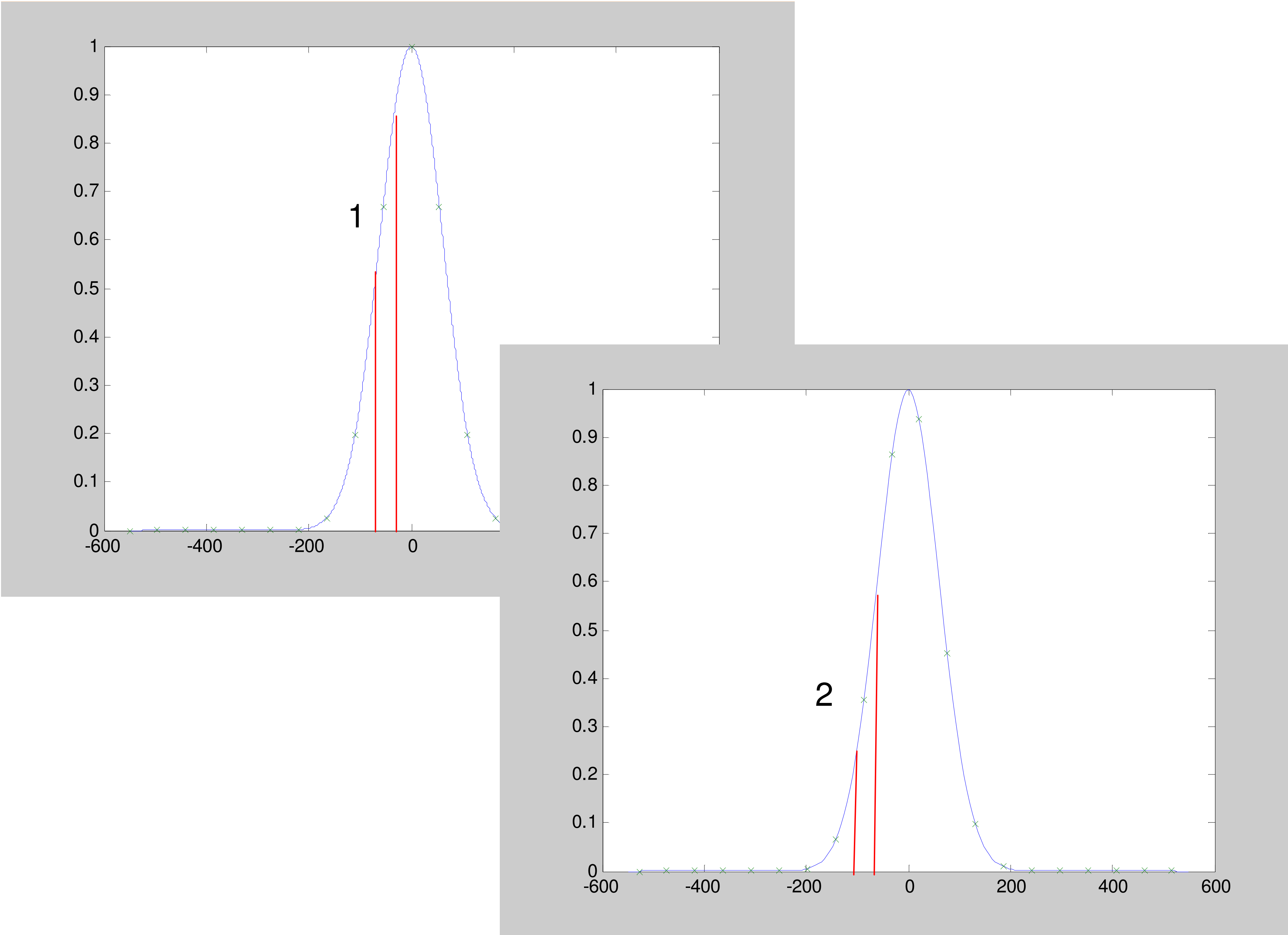

Gaussian funtion for the back scatter taken from cross section
normalised #counts
distance µm
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-600
-400
-200
0
200
400
600



Zoomed in to show view across ~3 pixels


1


25µm
2


1
2
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-600
-400
-200
0
200
400
600

Issues

- Needs to be done in 2D, can take a cross section in the y-dir also and generate a 2D function
- Because this depends on the distance from the centre of the pixel a source of error will be found in determining the centre of the pixel
- Time consuming, need to write script to go through all the acquisition files and rebuild the pixel maps