PHYS 386 HW3

(1) Read Cousins, “Why isn’t every physicist a Bayesian?” Answer the following questions from the paper with just a sentence or some bullet points each. We will discuss these in class next week.

A. Section IV: What are some of the issues in incorporating a physical boundary into a confidence interval? (Note: this paper was written 4 years before Feldman-Cousins).

> - In the Bayesian framework, the credible intervals depend on the prior distribution. For example, the choice of which quantity should have a uniformly distributed prior for values greater than zero (m, m^2 or $\ln m$, etc.) changes the estimated upper limit.
> - In the classical framework, there is no way to incorporate physical boundaries as priors, so one can get unphysical upper limits (such as the negative mass upper limits in neutrino experiments).

B. Section V: What are the criteria for construction of a classical confidence interval?

> For the construction of a classical central confidence interval (μ_1, μ_2), one must find μ_1 s.t.
>
> $$P(n \geq n_0 | \mu_1) = \frac{1}{2}(1 - \mathrm{C.L.}),$$
>
> and μ_2 s.t.
>
> $$P(n \leq n_0 | \mu_2) = \frac{1}{2}(1 - \mathrm{C.L.}),$$
>
> where n_0 is the number of events observed and $\mathrm{C.L.}$ is the confidence level. One can also construct an approximate 68% $\mathrm{C.L.}$ confidence interval by finding μ_1, μ_2 s.t. the maximum of the log likelihood function decreases by 1/2, i.e.,
>
> $$\ln \mathcal{L}(n_0 | \mu_1) = \ln \mathcal{L}(n_0 | \mu_2) = \ln \mathcal{L}(n_0 | \hat{\mu}) - \frac{1}{2},$$
>
> where $\hat{\mu} = n_0$ is the maximum likelihood estimator of μ_t.

C. Section V: What are the criteria for the construction of a Bayesian confidence (credible) interval?

> For the construction of a Bayesian credible interval, one needs to construct the posterior P.D.F. from the likelihood function and Bayes theorem:
>
> $$P(\mu_t|n_0) = \frac{\mathcal{L}(n_0|\mu_t)P(\mu_t)}{\int_0^\infty \mathcal{L}(n_0|\mu_t)P(\mu_t)\,\mathrm{d}\mu_t}.$$
>
> When there is no prior knowledge on μ_t, one must choose an *uninformative* prior P.D.F. such as a uniform distribution. The problem with this approach, as stated in part A., is that it is not trivial to choose for what function of μ_t one should assume a uniform distribution, since this choice affects the credible intervals. After choosing a prior, the Bayesian credible interval (μ_1, μ_2) can be calculated by the criterion
>
> $$\int_{\mu_1}^{\mu_2} P(\mu_t|n_0)\,\mathrm{d}\mu_t = \mathrm{C.L.},$$
>
> where $\mathrm{C.L.}$ is the confidence level of the credible interval, and an additional criterion which depends on the kind of credible interval one wants. For a central interval, the additional criterion is given by:
>
> $$\int_0^{\mu_1} P(\mu_t|n_0)\,\mathrm{d}\mu_t = \int_{\mu_2}^{\infty} P(\mu_t|n_0)\,\mathrm{d}\mu_t.$$
>
> For the shortest Bayesian interval, the additional criterion is that the posterior $P(\mu_t|n_0)$ for any μ_t outside the credible interval is less than $P(\mu_t|n_0)$ for all μ_t instide the credible interval.

D. Section V: What prior would Jeffreys suggest we use in the absence of any knowledge about the true value of our parameter, mu? What is the reasoning behind this set of priors?

> Harold Jeffreys suggests using the prior $P(\mu_t) = 1/\mu_t$ when nothing is known about μ_t because it is invariant under changes of power of the parameter being estimated, i.e., the priors $P(\mu_t) = 1/\mu_t$ and $P(\mu_t^k) = 1/\mu_t^k$ are consistent for any power k. Note that this is equivalent to a uniform distribution for $\ln \mu_t$. This prior also allows consistency between experiments measuring the same decay process with different standards for the passage of time.

E. What are some of the drawbacks of the classical vs. Bayesian methods for interval construction?

> - Bayesian intervals depend on the prior, and it is not clear which one to choose.
> - Some Bayesian intervals/limits fail the criterion of the frequentist coverage.
> - In the presence of systematic Gaussian error on the sensitivity of the experiment, classical limits can lead to unacceptable results.

F. Section VI: How might you deal with a systematic Gaussian error on top of your Poisson process?

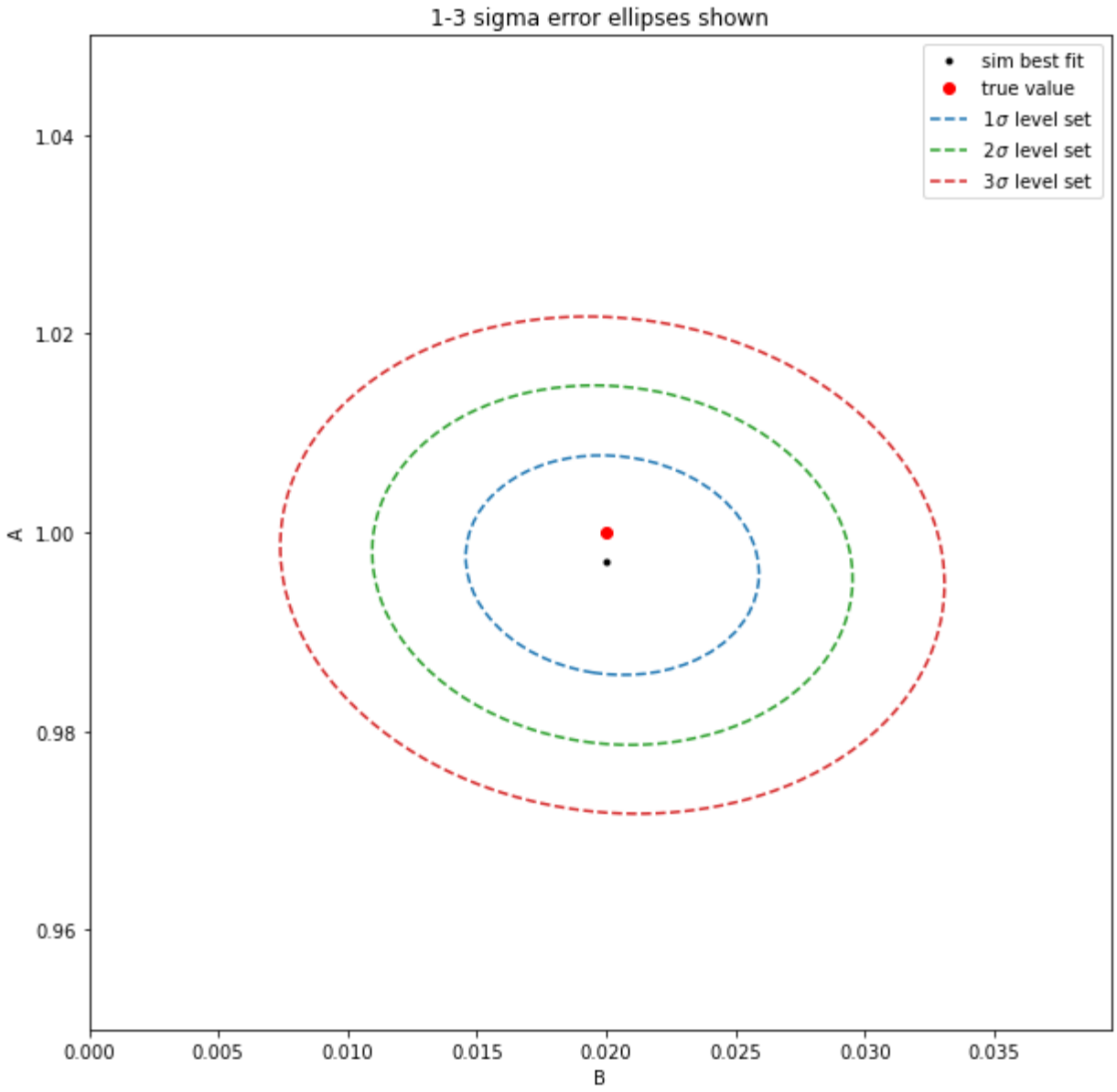
> - One could use a Bayesian approach to calculate a posterior P.D.F. for the sensitivity as a subsidiary measurement.
> - One could take a fully Bayesian approach as long as it satisfies the frequentist coverage criterion.

(2) An exercise using the chi-squared notebook. You will be working from the exercise in “now repeat the fit using this full inverse covaraince to compute Chi-squared” pertaining to fitting A and B for our toy BAO feature including a non-trivail covariance matrix.

A. Use Table 1 to set the delta chi-squared levels correctly corresponding to 1, 2, and 3 sigma.. Note these levels and show the resulting contours.

In [2]:

```
# 1, 2 and 3 sigma delta chi-squared levels for 2 parameters:
sigma_1 = 2.30
sigma_2 = 6.17
sigma_3 = 11.8

## define a grid in A and B and compute a grid search.
A_points = np.arange(0.95,1.05,.001)
B_points = np.arange(0.0,.04,.0005)
N_pts_A = np.size(A_points)
N_pts_B = np.size(B_points)
chi_sq_map = np.zeros([N_pts_A,N_pts_B])

# run simulation
Raw_data, Binned_r, Binned_mean, Binned_error, inv_cov = run_sim()

## compute chi_squared for each point
i = 0
while (i < N_pts_A):
    j = 0
    while (j < N_pts_B):
        chi_sq_map[i,j] = chi_squared(Binned_r,Binned_mean, inv_cov,A_points[i],B_points
        j+=1
    i+=1

delta_chi_sq_map = chi_sq_map - np.min(chi_sq_map)

######
### make a plot of the 1,2,3 sigma error contours on the fit,
### show the "truth" from the simulations, and
### show the best fit.

## make a plot of chi_sq
delta_chi_sq_map = chi_sq_map - np.min(chi_sq_map)
plt.contour(B_points, A_points, delta_chi_sq_map,
            levels=np.array([sigma_1, sigma_2, sigma_3]),
            colors=['C0', 'C2', 'C3'],
            linestyles=['dashed'])

## plot the best fit
pos_min = np.where(chi_sq_map == np.min(chi_sq_map))
plt.plot(B_points[pos_min[1]],A_points[pos_min[0]],'k.', label='sim best fit')

## plot the "true" parmaters (eg what we used in the simulation)
plt.plot(np.array(0.02),np.array(1),'ro', label='true value')
plt.xlabel("B")
plt.ylabel("A")
plt.title("1-3 sigma error ellipses shown")
plt.plot([],[], color='C0', linestyle='--', label='$1\, \sigma$ level set')
plt.plot([],[], color='C2', linestyle='--', label='$2\, \sigma$ level set')
plt.plot([],[], color='C3', linestyle='--', label='$3\, \sigma$ level set')
plt.legend()
plt.show()
```

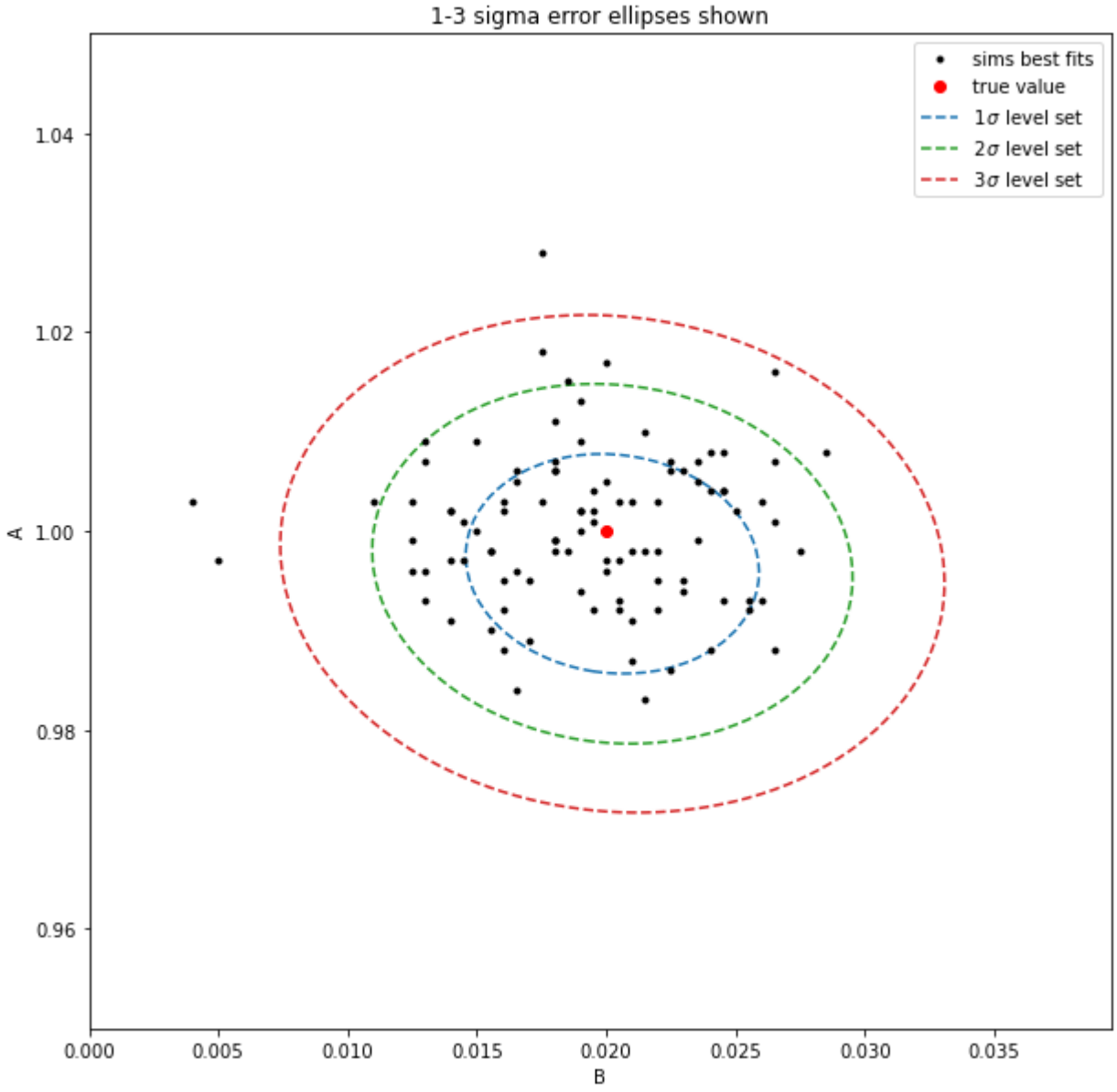


B. Generate a set of 100 simulations and store the best fit chi-squared values in a text file. Plot these as dots on your contour plot. Does this distribution make sense?

In [3]:

```
best_fit_chi2 = np.zeros((100, 2))
s_chi_sq_map = np.zeros([N_pts_A,N_pts_B])

for k in range(100):
    # run simulation
    s_Raw_data, s_Binned_r, s_Binned_mean, s_Binned_error, s_inv_cov = run_sim()

    ## compute chi_squared for each point
    for i in range(N_pts_A):
        for j in range(N_pts_B):
            s_chi_sq_map[i,j] = chi_squared(s_Binned_r,
                                            s_Binned_mean,
                                            s_inv_cov,
                                            A_points[i],
                                            B_points[j])

    # best fit
    s_pos_min = np.where(s_chi_sq_map == np.min(s_chi_sq_map))
    best_fit_chi2[k, 0] = A_points[s_pos_min[0]]
    best_fit_chi2[k, 1] = B_points[s_pos_min[1]]
```

In [4]:
```
plt.contour(B_points, A_points, delta_chi_sq_map,
            levels=np.array([sigma_1, sigma_2, sigma_3]),
            colors=['C0', 'C2', 'C3'],
            linestyles=['dashed'])
plt.plot(best_fit_chi2[:, 1],best_fit_chi2[:, 0],'k.', label='sims best fits')
plt.plot(np.array(0.02),np.array(1),'ro', label='true value')
plt.xlabel("B")
plt.ylabel("A")
plt.title("1-3 sigma error ellipses shown")
plt.plot([],[], color='C0', linestyle='--', label='$1\, \sigma$ level set')
plt.plot([],[], color='C2', linestyle='--', label='$2\, \sigma$ level set')
plt.plot([],[], color='C3', linestyle='--', label='$3\, \sigma$ level set')
plt.legend()
plt.show()
```

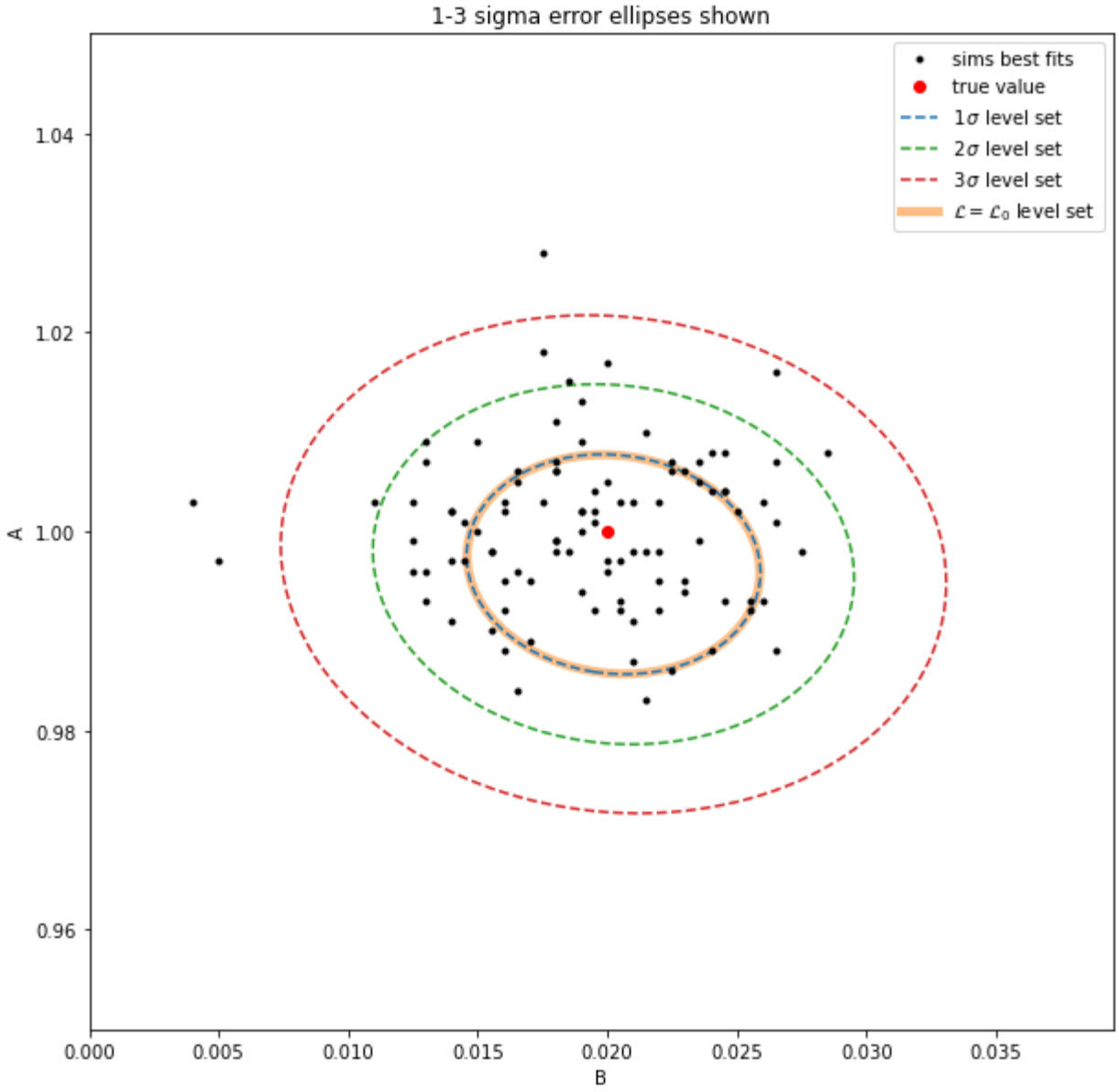


> The proportions of simulations inside the sigma ellipses seem right since they are close to 68%, 95% and 99%. It would have been better if the ellipses were centered at the 'true value', but this is supposedly not known a priori. One better way to do it is taking the average of the simulations and constructing the χ^2 from that Monte Carlo.

C. Use 2log(L) = -chi_sq to convert your chi-squared to a probability distribution. Normalize this probability distribution to make its integral 1. Now choose a level L0 such that the integral of L with (L>L0) = 68%. Draw this contour on your plot.

In [5]:

```
L = np.exp(-delta_chi_sq_map/2)
L = L/np.sum( L * (A_points[1]-A_points[0]) * (B_points[1]-B_points[0]))

contours = np.linspace(0, np.amax(L), 1000)
i = 0
integral = 1.0
while integral > 0.68:
    L_int = np.where(L>contours[i], L, 0)
    integral = np.sum( L_int * (A_points[1]-A_points[0]) * (B_points[1]-B_points[0]) )
    i += 1

plt.contour(B_points, A_points, L,
            levels=np.array([contours[i]]),
            colors=['C1'],
            linestyles=['solid'],
            linewidths = 5,
            alpha = 0.5
            )

plt.contour(B_points, A_points, delta_chi_sq_map,
            levels=np.array([sigma_1, sigma_2, sigma_3]),
            colors=['C0', 'C2', 'C3'],
            linestyles=['dashed'])

plt.plot(best_fit_chi2[:, 1],best_fit_chi2[:, 0],'k.', label='sims best fits')
plt.plot(np.array(0.02),np.array(1),'ro', label='true value')
plt.xlabel("B")
plt.ylabel("A")
plt.title("1-3 sigma error ellipses shown")
plt.plot([],[], color='C0', linestyle='--', label='$1\, \sigma$ level set')
plt.plot([],[], color='C2', linestyle='--', label='$2\, \sigma$ level set')
plt.plot([],[], color='C3', linestyle='--', label='$3\, \sigma$ level set')
plt.plot([],[], color='C1', linewidth=5, alpha=0.5, label='$\mathcal{L}=\mathcal{L}_0$ l
plt.legend()
plt.show()
```



> The level set corresponding to the $\mathcal{L}_0$ s.t. the integrated probability for $\mathcal{L} > \mathcal{L}_0$ is 68% coincides with the $1\,\sigma$ level set of the χ^2 distribution. This makes sense since $1\,\sigma$ corresponds to 68% probability.