```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit as opti

def function_plot(x,A,B,C):
    return A*np.exp(x/B)+C
parameter_names = ["A", "B", "C"]

fname = 'astro11.csv'
x_data = (np.loadtxt(fname, delimiter=',', comments ='#', usecols=(0,1)))[:, 0]
y_data = (np.loadtxt(fname, delimiter=',', comments ='#', usecols=(0,1)))[:, 1]
sgm = 1

plt.scatter(x_data, y_data, c='r', s=1)
plt.title("Plot of given x and y coordinates")
plt.xlabel("x coordinates",)
plt.ylabel("y coordinates")
plt.show()
assumption = [1,-1,0]
fit_parameters, fit_cov = opti(function_plot, x_data, y_data, p0=assumption)

def chi_square (fit_parameters, x, y, sigma):
    if sigma is None:
        sigma = 1
    return np.sum((y-function_plot(x, *fit_parameters))**2/sigma**2)

chi2 = chi_square(fit_parameters, x_data, y_data, sgm)
dof = len(x_data) - len(fit_parameters)
chi2_df = chi2/dof
print ("\nGoodness of fit, via Chi squared")
print ("Chi squared per degree of freedom = {:.4}\n".format(chi2_df))

fit_cov_new = fit_cov*(1/chi2_df)
fit_para_error = np.sqrt(np.diag(fit_cov_new))

print("The fitted parameters:")
for i in range(len(fit_parameters)):
    print('{} = {:.4} +/- {:.4}'.format(parameter_names[i], fit_parameters[i],
                                          fit_para_error[i]))

plt.scatter(x_data, y_data, zorder=1, c='r', label="Raw Data", s=1)
plt.plot(x_data, function_plot(x_data, *fit_parameters), zorder=10, c='b',
                                                 label="Best Fit", linewidth=3)
plt.title("Plot of given coordinates with a Best Fit")
plt.xlabel("X Coordinates")
plt.ylabel("Y Coordinates")
plt.legend()
plt.show()
```