

def twoD_Gaussian(xdata_tuple, amplitude, xo, yo, sigma_x, sigma_y, theta, offset): (x, y) = xdata_tuple xo = float(xo)yo = float(yo) a = (np.cos(theta)**2)/(2*sigma x**2) + (np.sin(theta)**2)/(2*sigma y**2) $b = -(np.sin(2*theta))/(4*sigma_x**2) + (np.sin(2*theta))/(4*sigma_y**2)$ $c = (np.sin(theta)**2)/(2*sigma_x**2) + (np.cos(theta)**2)/(2*sigma_y**2)$ g = offset + amplitude*np.exp(- (a*((x-xo)**2) + 2*b*(x-xo)*(y-yo))+ c*((y-yo)**2)))return g.ravel()

share improve this answer answered Nov 11 '14 at 11:52 Ferdinand van Wyk

The reason for this is that automatic tuple unpacking when it is passed to a function as a parameter

has been removed as of Python 3. For more information see here: PEP 3113

add a comment

shape (n*m) not (n,m) respectively. So you use ravel() to flatten your 2D arrays: xdata = np.vstack((xx.ravel(),yy.ravel()))

ydata = data noisy.ravel()

curve_fit() wants to the dimension of xdata to be (2,n*m) and not (2,n,m). ydata should have

1 By the way: I'm not sure if the parametrization with the trigonometric terms is the best one. E.g., taking the one described here might be a bit more robust under numerical aspects and large deviations.

share improve this answer edited Feb 5 '14 at 1:23 answered Feb 5 '14 at 1:09

popt, pcov = opt.curve_fit(twoD_Gaussian, xdata, ydata, p0=initial_guess)

add a comment Your Answer

Post Your Answer



G Sign up using Google f Sign up using Facebook **Email** Required, but never shown Sign up using Email and Password

By clicking "Post Your Answer", you agree to our terms of service, privacy policy and cookie policy

or ask your own question.

COMPANY

Contact Us

Not the answer you're looking for? Browse other questions tagged python numpy scipy

Disable Responsiveness

STACK OVERFLOW

Questions

PRODUCTS

NETWORK

Technology >

Culture / Recreation >

Life / Arts

Science >

Other >

Blog Facebook Twitter LinkedIn