```
clear all
close all

% just to be sure
opts = fitoptions('method','nonlinearleastsquares','algorithm','gauss-
newton');

%%

% generate Poisson with lambda = 5
k = 0:20;
lambda = 5;
poiss = exp(-lambda)*(lambda.^k)./factorial(k);
poiss = poiss + 1e-2*randn(size(poiss));

% LS fit to 'gauss1'
norm_f = fit(k(:),poiss(:),'gauss1',opts);
norm_c = coeffvalues(norm_f);
norm_c(3) = norm_c(3)/sqrt(2);

% call optimize fit with appropriate inputs
norm_opt_c = optimize_fit(k,poiss,norm_c,'normal');

% for the plot
norm_LS = norm_c(1)*exp(-((k - norm_c(2)).^2)/(2*norm_c(3)^2));
norm_opt = norm_opt_c(1)*exp(-((k - norm_opt_c(2)).^2)/(2*norm_opt_c(3)^2));


%% Poisson with lambda = 100
k1 = 50:150;
lambda1 = 100;
poiss1 = exp(-lambda1)*(lambda1.^k1)./factorial(k1);
poiss1 = poiss1 + 1e-3*randn(size(poiss1));

norm_f1 = fit(k1(:),poiss1(:),'gauss1',opts);
norm_c1 = coeffvalues(norm_f1);
norm_c1(3) = norm_c1(3)/sqrt(2);

norm_opt_c1 = optimize_fit(k1,poiss1,norm_c1,'normal');

norm_LS1 = norm_c1(1)*exp(-((k1 - norm_c1(2)).^2)/(2*norm_c1(3)^2));
norm_opt1 = norm_opt_c1(1)*exp(-((k1 -
norm_opt_c1(2)).^2)/(2*norm_opt_c1(3)^2));

%%

% generate Lorentzian
x = 400:700;
x0 = 550;
gamma = 50;
lorentz = (gamma^2)./((x - x0).^2 + gamma^2);
lorentz = lorentz + 1e-1*randn(size(lorentz));

% construct Lorentzian fittype model
```

```
lorentz_fun = 'A*(gamma^2)./((x - x0).^2 + gamma^2)';
lorentz_coefs = {'x0','gamma','A'};
lorentz_model =
fittype(lorentz_fun,'coefficients',lorentz_coefs,'independent','x');

% LS fit
lorentz_f = fit(x(:),lorentz(:),lorentz_model);
lorentz_c = coeffvalues(lorentz_f);

% optimize
lorentz_opt_c = optimize_fit(x,lorentz,lorentz_c,'lorentz');

% curves
lorentz_LS = (lorentz_c(2)^2)./((x - lorentz_c(1)).^2 + lorentz_c(2)^2);
lorentz_opt = (lorentz_opt_c(2)^2)./((x - lorentz_opt_c(1)).^2 +
lorentz_opt_c(2)^2);

%%

tau = 1:1e5; % units here are seconds
k_corr = 1e-4; % Hz
corr = exp(-k_corr*tau);

corr_fun = '1./((1 + tau/taud).*sqrt(1 + 0.01*(tau/taud)))';
corr_coefs = {'taud'};
corr_model = fittype(corr_fun,'coefficients',corr_coefs,'independent','tau');

corr_f = fit(tau(:),corr(:),corr_model);
taud = corr_f.taud;

taud_opt = optimize_fit(tau,corr,taud,'fcs');

corr_fit = 1./((1 + tau/taud).*sqrt(1 + 0.01*(tau/taud)));
corr_opt = 1./((1 + tau/taud_opt).*sqrt(1 + 0.01*(tau/taud_opt)));


%% plot
figure('papersize',[11 8.5]); orient tall

subplot 221
plot(k,poiss,k,norm_LS,k,norm_opt); legend('data','fit','opt')
text(0.75*max(k),0.7*max(poiss),['\mu_{fit} = ' num2str(norm_c(2))])
text(0.75*max(k),0.55*max(poiss),['\sigma_{fit}^2 = ' num2str(norm_c(3)^2)])
text(0.75*max(k),0.4*max(poiss),['\mu_{opt} = ' num2str(norm_opt_c(2))])
text(0.75*max(k),0.25*max(poiss),['\sigma_{opt}^2 = '
num2str(norm_opt_c(3)^2)])

subplot 222
plot(k1,poiss1,k1,norm_LS1,k1,norm_opt1); legend('data','fit','opt')
text(0.75*max(k1),0.7*max(poiss1),['\mu_{LS} = ' num2str(norm_c1(2))])
text(0.75*max(k1),0.55*max(poiss1),['\sigma_{LS}^2 = '
num2str(norm_c1(3)^2)])
text(0.75*max(k1),0.4*max(poiss1),['\mu_{opt} = ' num2str(norm_opt_c1(2))])
```

```
text(0.75*max(k1),0.25*max(poiss1),['\sigma_{opt}^2 = '
num2str(norm_opt_c1(3)^2)])

subplot 223
plot(x,lorentz,x,lorentz_LS,x,lorentz_opt); legend('data','fit','opt')
text(425,0.95*max(lorentz),['x_0^{LS} = ' num2str(lorentz_c(1))])
text(425,0.8*max(lorentz),['\gamma^{LS} = ' num2str(lorentz_c(2))])
text(425,0.65*max(lorentz),['x_0^{opt} = ' num2str(lorentz_opt_c(1))])
text(425,0.5*max(lorentz),['\gamma^{opt} = ' num2str(lorentz_opt_c(2))])

subplot 224
plot(tau,corr,tau,corr_fit,tau,corr_opt); legend('data','fit','opt')
text(0.7*max(tau),0.7*max(corr),['\tau_D = ' num2str(taud)])
text(0.7*max(tau),0.55*max(corr),['\tau_D^{opt} = ' num2str(taud)])

cd 'C:\Users\Nick\Dropbox\matlab course\completed HW'
print 'PS3_solutions.pdf' -dpdf




function coefs_out = optimize_fit(x,data,coefs,type)


opts = optimset('display','off',...
                'funvalcheck','off',...
                'maxfunevals',1e10,...
                'maxiter',1e8,...
                'tolx',1e-10,...
                'tolfun',1e-10);

if strcmp(type,'normal')
    coefs_out = fminsearch(@norm_sse,coefs,opts);
end

if strcmp(type,'lorentz')
    coefs_out = fminsearch(@lorentz_sse,coefs,opts);
end

if strcmp(type,'fcs')
    coefs_out = fminsearch(@fcs_sse,coefs,opts);
end


function sse = norm_sse(coefs)
    a = coefs(1);
    mu = coefs(2);
    sigma = coefs(3);
    N_fit = a*exp(-((x - mu).^2)/(2*sigma^2));
    residuals = N_fit - data;
    sse = sum(residuals.^2);
end
```

```
function sse = lorentz_sse(coefs)
    x0 = coefs(1);
    gamma = coefs(2);
    A = coefs(3);
    L_fit = A*(gamma^2)./((x-x0(1)).^2 + gamma^2);
    residuals = L_fit - data;
    sse = sum(residuals.^2);
end

function sse = fcs_sse(coefs)
    tau_d = coefs;
    F_fit = 1./((1 + x/tau_d).*sqrt(1 + 0.0001*(x/tau_d)));
    residuals = F_fit - data;
    sse = sum(residuals.^2);
end

end
```

fuzzy but it'll work: