

Midterm

χ^2 distribution and fitting procedure

due: November 16, 2016

problem 1 PHYS 139/239

χ^2 is a “statistic” defined as the **sum of the squares of n independent t-values**.

$$\chi^2 = \sum_i \left(\frac{x_i - \mu_i}{\sigma_i} \right)^2, \quad x_i \sim N(\mu_i, \sigma_i)$$

Chisquare(ν) is a **distribution** (special case of Gamma), defined as

$$\chi^2 \sim \text{Chisquare}(\nu), \quad \nu > 0$$
$$p(\chi^2)d\chi^2 = \frac{1}{2^{\frac{1}{2}\nu} \Gamma(\frac{1}{2}\nu)} (\chi^2)^{\frac{1}{2}\nu-1} \exp\left(-\frac{1}{2}\chi^2\right) d\chi^2, \quad \chi^2 > 0$$

prove the theorem that $p(\chi^2)$ is distributed with Chisquare(ν) statistics where ν is the number of x_i variables from normal distributions

problem 1 PHYS 139/239

Hints:

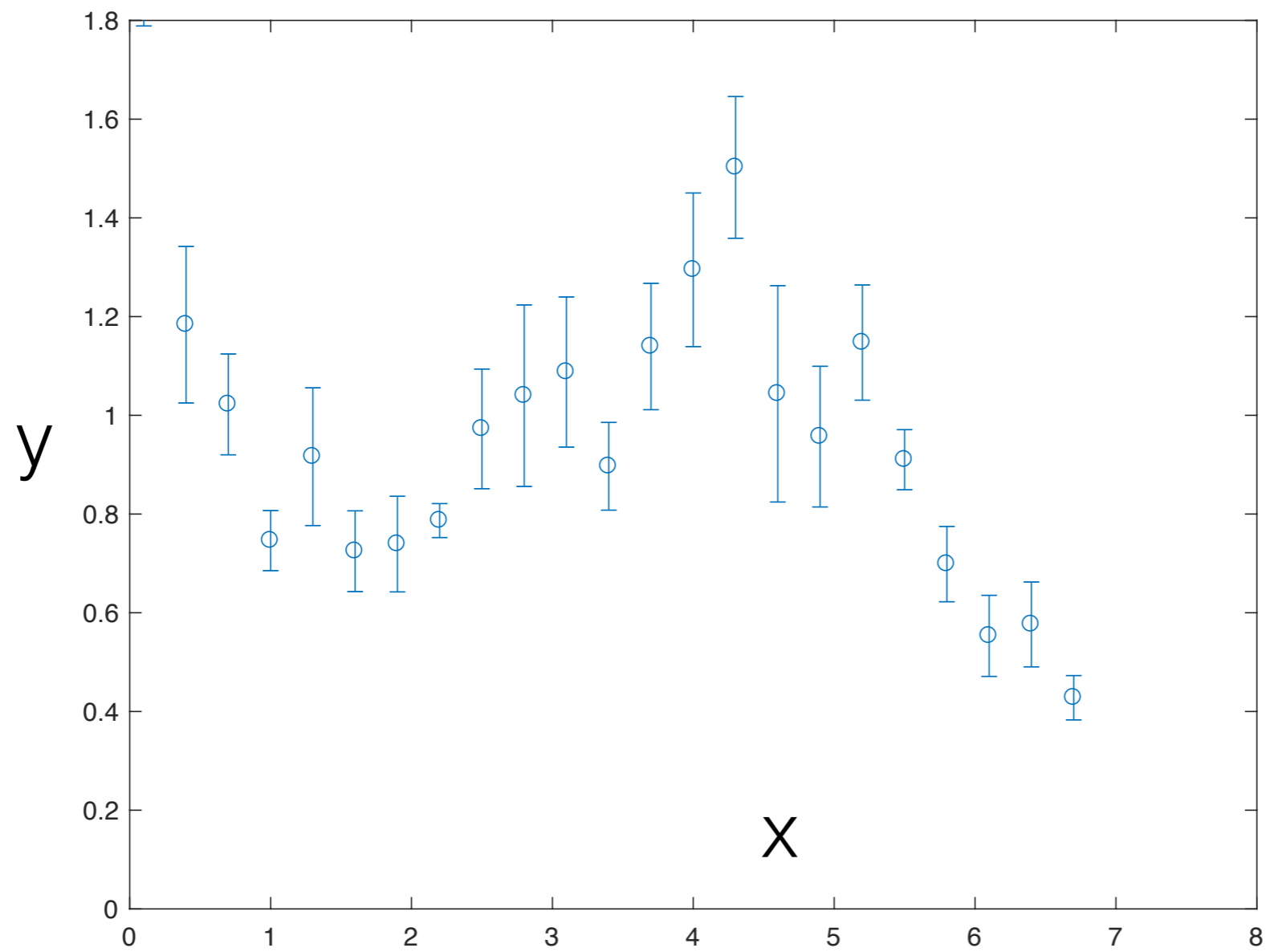
- show first the statement for a single x_i normal variable with zero mean value
- using characteristic functions show that the theorem is true for arbitrary number of x_i variables with zero mean values
- show that after shifting the mean values to arbitrary values the theorem still holds

problem 2 PHYS 139/239

We measure in an experiment at 23 values of x_i the outcome y_i from normal distributions where the results are listed in the data.txt file:

x	y	y error
0.1000000000000000	1.955692474636036	0.166896282383792
0.4000000000000000	1.183586547503424	0.158551780782385
0.7000000000000000	1.022128862295741	0.102145122199102
1.0000000000000000	0.746134082944572	0.060820536337125
1.3000000000000000	0.916188421395087	0.139506053368529
1.6000000000000000	0.724682156536752	0.081793212333357
1.9000000000000000	0.739127499035786	0.096894347069944
2.2000000000000000	0.786742524422711	0.034353661707974
2.5000000000000000	0.972558512530457	0.121213729440151
2.8000000000000000	1.039776955766267	0.183845107945299
3.1000000000000000	1.087705062846587	0.152064123528747
3.4000000000000000	0.896727858969629	0.088835443972525
3.6999999999999999	1.139381591276074	0.128022842446142
4.0000000000000000	1.294829163615035	0.155588445889791
4.2999999999999999	1.502261299770580	0.143493932937373
4.6000000000000000	1.043529911555928	0.219186627495748
4.8999999999999999	0.956827376670183	0.142469078945670
5.1999999999999999	1.147387265711086	0.116683264235504
5.4999999999999999	0.909994065501967	0.060876724854546
5.7999999999999999	0.698671186235582	0.076323301379691
6.1000000000000000	0.553227945238010	0.082132016628130
6.3999999999999999	0.576371045690540	0.085922021448737
6.6999999999999999	0.427880507687987	0.044877728959367

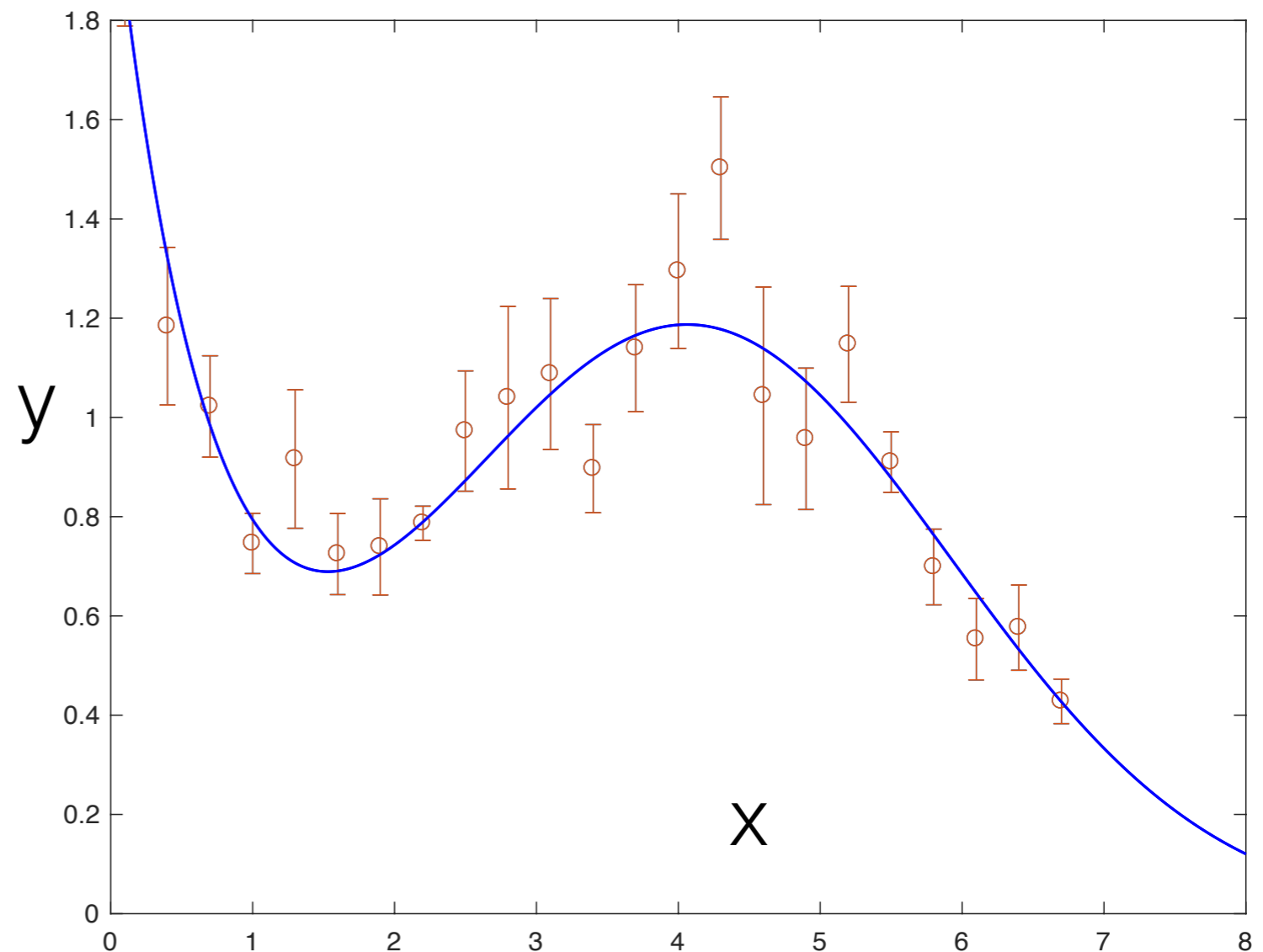
(A) plot the data



(B) assuming that the experiment is described by the theoretical function of five parameters

$$f(x) = b_1 \exp(-b_2 x) + b_3 \exp\left(-\frac{1}{2} \frac{(x - b_4)^2}{b_5^2}\right)$$

fit the data to the $f(x)$ function which represents the measured y_i data and determine from the fit the five parameters using the Maximum Likelihood Method of the χ^2 fitting procedure



- (C) calculate the Hessian and the covariance matrix of the fit
- (D) Calculate the errors (confidence intervals) on the individual b_i parameters when the other four are marginalized. Use the diagonal part of the covariance matrix for this estimate.
- (E) calculate the contour plot of the joint confidence distribution of b_3 and b_5 when the other three parameters are marginalized.
- (F) **239 only:** Same contour plot when the other 3 variables are constrained to some values of your choice. Compare the two results and explain. try to simulate the contour plot.
- (G) Analyze the Goodness-of-Fit for your procedure
- (H) **239 only:** predict with error estimates the x-values where you would measure $y=1$.

Matlab codes which help the solutions:

2B/C: lsqcurvefit

```
FitFunc=@(p,x) wfit.*(p(1)*exp(-p(2)*x) + p(3)*exp(-0.5*(x-p(4)).^2/p(5)^2));
```

```
p0=[5 input data to start];
```

```
[p,resnorm,residual,exitflag,output,lambda,Jacob]= lsqcurvefit(FitFunc,p0,xfit,yfit.*wfit);
```

2B/C: nlinfit

```
FitFunc=@(p,x) (p(1)*exp(-p(2)*x) + p(3)*exp(-0.5*(x-p(4)).^2/p(5)^2));
```

```
p0=[5 input data to start];
```

```
[p,resid,Jacob,CovB,MSE,ErrorModelInfo] = nlinfit(xfit,yfit,FitFunc,p0,'Weights',wfit);
```

Hessian CovB covariance matrix

`hessian(f,v)` finds the Hessian matrix of the scalar function `f` with respect to vector `v` in Cartesian coordinates. If you do not specify `v`, then `hessian(f)` finds the Hessian matrix of the scalar function `f` with respect to a vector constructed from all symbolic variables found in `f`. The order of variables in this vector is defined by `symvar`.

2H: nlpredci

`[Ypred,delta] = nlpredci(modelfun,X,beta,R,'Covar',CovB)` returns predictions, Ypred, and 95% confidence interval half-widths, delta, for the nonlinear regression model modelfun at input values X. Before calling nlpredci, use nlinfit to fit modelfun and get the estimated coefficients, beta, residuals, R, and variance-covariance matrix, CovB.

example

