```
function F=beamWaistCalculator(xData,IData);
%Takes data taken with photodiode and knife method and fits it to an erf function.
%returns a vector a.
%a(2)*sqrt(2) would be the waist in
%this direction, a(1)*2 is the total power, a(4)-a(1) is the offset)

%function to fit it to
fitF=inline('a(1).*erf((x.-a(3))./a(2))+a(4)','a','x');

%R^2 error
R2=inline('sum((IData-fitF(a,xData)).^2)','xData','IData','a');

%perform fit
aa=fminsearch(@(a)R2(xData,IData,a),[-max(IData),(max(xData)-
min(xData))/2,mean(xData),mean(IData)], optimset('TolX', 1e-15));

%R2(xData,IData,aa);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PLOT%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%the plot is not essential to the function, but it lets the user sanity-check
%the results

%prepare values
xx=min(xData):(max(xData)-min(xData))/1000:max(xData);
yy=fitF(aa,xx);

hold off
plot(xData*.0254,IData,'r+') %convert to mm assuming data in mils (!)
hold on
plot(xx*.0254,yy,'k-'); %convert to mm assuming data in mils (!)
h=legend('Data points','Fit data')
set(h,'fontsize',18)
legend(h,'location','northeast')
xlabel('knife position perpendicular to beam direction (mm)')
ylabel('power (arbitrary units)')
set(findall(gcf,'type','text'),'FontSize',18)
print -depsc waistFitExample.eps

pause();
F=aa;
```