Total Least Squares

Notes for next time: Proof of SVD minimum rank approximation ARMAX example Proof of bias in least squares case from matrix perspective Zero noise column case Simulation of least squares bias case

- Always, ..., Never forget to check your references
- Some linear algebra
- Problem statement
- Derivation of the result
- Simulated experimental results (and mistakes)
- Nonlinear TLS
- My application of NLTLS

Some Credit is Due

Sabine Van Huffel

Has two really good books from SIAM about TLS. The first is an outgrowth of her Ph.D. thesis in 1987 and the second is a collection of papers by many authors and covers more recent advances until 1997. All of the TLS stuff from this talk comes from the first book.

S. Van Huffel, J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis.* SIAM, 1991.

Edited by S. Van Huffel, *Recent Advances in Total Least Squares and Errors-in-Variables Modeling* SIAM, 1997.

Applications

- Parametric system identification with noisy measurements
- System-Id with bounded data uncertainties
- Errors in the measurements with constraints on the parameters.
- Estimating Direction of Arrival for SONAR, RADAR etc
- System identification of kidney using radioactive markers
- Estimation of longitudinal tire properties with low resolution sensors
- etc.

Ease Right In

We as engineers often approximate the world with linear models of the form

$$y_1 = a_1 x_1 + a_2 x_2 + \ldots + a_n x_n$$

where the a_i terms are variables and the x_i terms are the model parameters.

Which looks good since if we take m samples we get

$$y = Ax$$

where,

$$x \in \Re^{n \times 1}, A \in \Re^{m \times n}$$

We remember from linear algebra that depending on m, n and A there are a few different kind of solutions for linear systems of equations that look like this.

Kinds of Solutions

If

$m < n$ (A is fat), or

$m > n$ (A is skinny) and A has a null space

Then there are infinitely many choices of x that satisfy the equation.

If

$$m = n, \text{ and } \text{Null}(A) = \emptyset$$

Then we have a unique solution for x.

Finally, if

$$m > n, \text{ and } \text{Null}(A) = \emptyset$$

Then there may be one, or no solution. Depending on weather or not the equations are over determined.

Finding Solutions

If we can measure the outputs y exactly and our model is correct, then we are done.

$$x = A^{-1}y$$

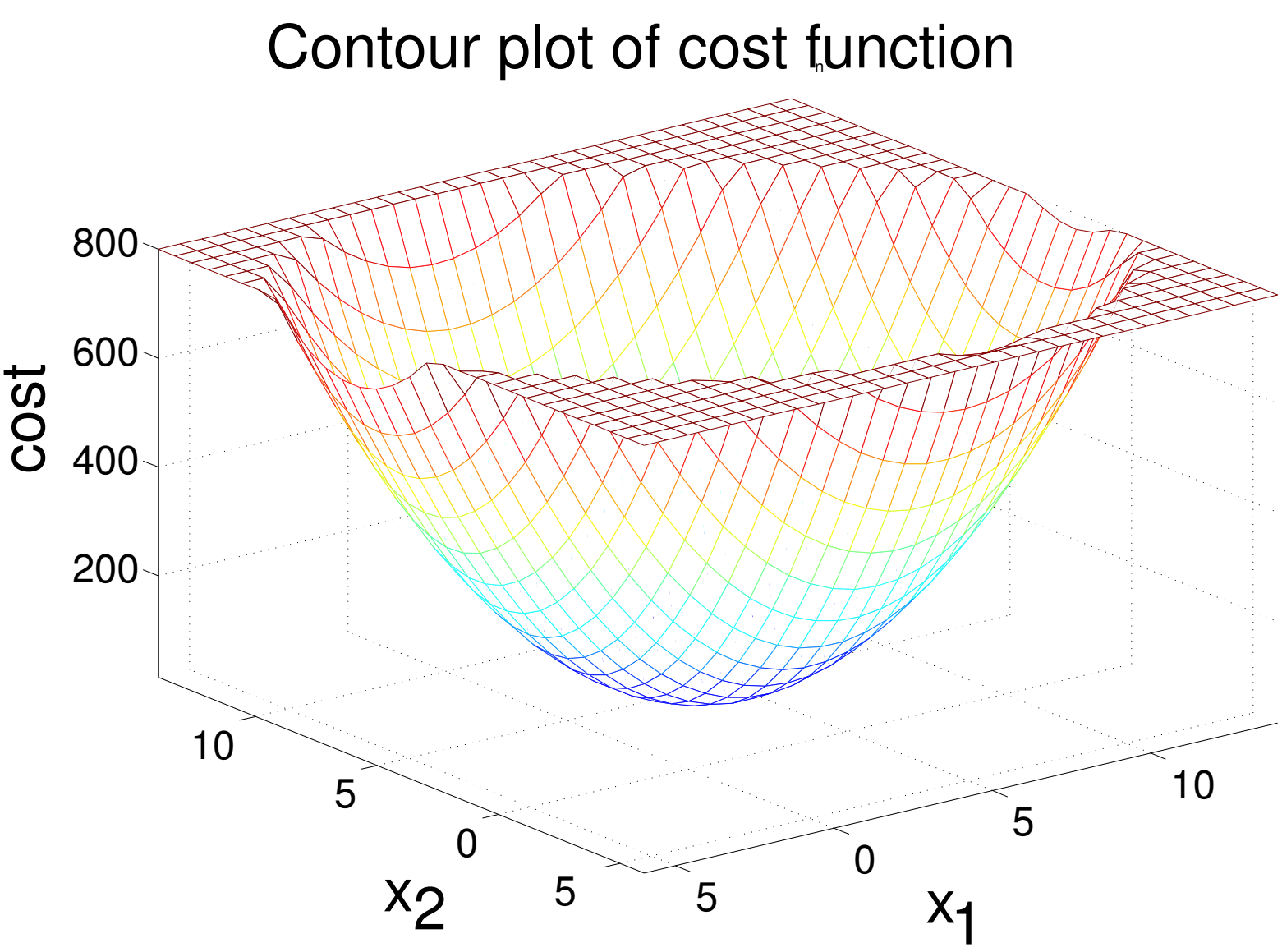
Usually this is not the case, so we assume the output has errors.

$$y = Ax + \epsilon$$

Gauss discovered that if you minimize the sum of the squared measurement errors, that you could get a pretty good analytic guess for x.

$$\begin{aligned}
\epsilon^2 &= \epsilon^T \epsilon \\
&= (y - Ax)^2 \\
&= x^T A^T A x - 2x^T A^T y + y^T y
\end{aligned}$$

A Bowl


Contour plot of cost function
cost
800
600
400
200
10
5
0
5
x_2
5
0
5
10
x_1

Minimizing a quadratic

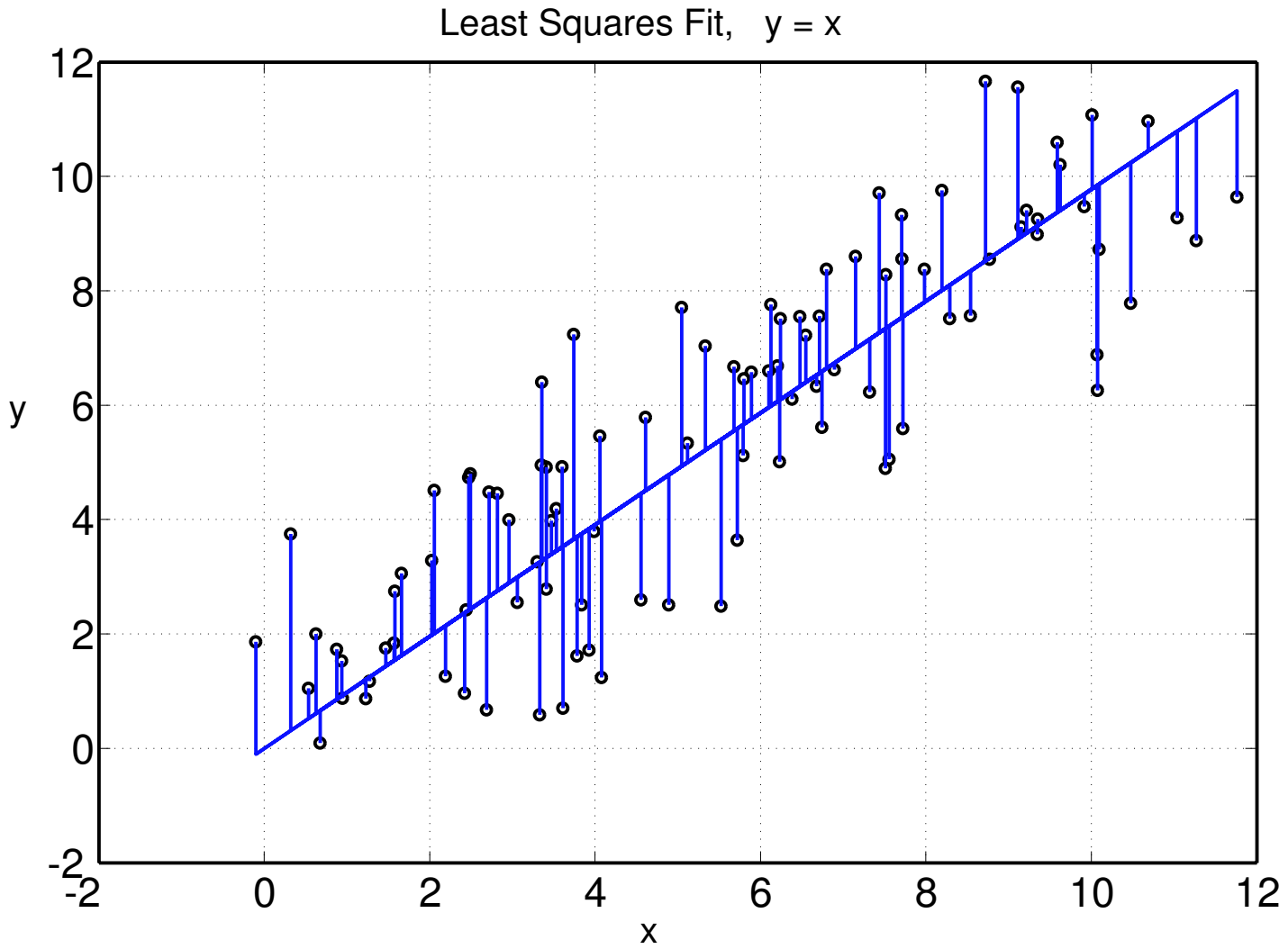
$$\begin{aligned}
\frac{d}{dx}(\epsilon^2) &= \left(x^T A^T A x - 2x^T A^T y + y^T y\right) \\
&= 2A^T A x - 2A^T y \\
&= 0 \\
&\Rightarrow \\
\hat{x} &= (A^T A)^{-1} A^T y \\
&= A \backslash y
\end{aligned}$$

$(A^T A)$ is invertible when the problem is well behaved

That is pretty cool. Convex optimization in 1800. Actually the best you can do in some cases.

Least Squares

Least squares minimizes ϵ for the model: $y = Ax + \epsilon$



What was the measurement again?

Often, we rely on more than one measurement for our models.

$$y_1 - \epsilon_1 = (a_1 + \Delta a_1)x_1 + (a_2 + \Delta a_2)x_2 + \ldots + (a_n + \Delta a_n)x_n$$

Which can be written:

$$y = (A + \Delta A)x + \epsilon$$

Where the $\Delta a_i, \Delta A$ represent measurement errors in addition to ϵ.

The least squares estimate is asymptotically biased for this case.

Think of it as: the ΔA terms multiply the x terms, so the estimator tends to make the $\hat{x}$ small to reduce the size of the equation error.

Reinterpret the Goal

So, Gauss' least squares method gives a bias for this case. But the goal of the estimation scheme is to minimize the measurement errors. We can try that again:

$$\begin{aligned} \text{Minimize:} \quad & \|\Delta A\|, \|\epsilon\| \\ \text{Subject to:} \quad & y + \epsilon = (A + \Delta A)x \end{aligned}$$

Define two new variables:

$$\begin{aligned} \hat{A} &= A + \Delta A \\ \hat{y} &= y + \epsilon \end{aligned}$$

Reinterpret the Goal Some More

Then we can further write this as:

$$\begin{array}{rl} \text{Minimize:} & \\ \Delta A, \epsilon, x & \|[\hat{A}, \hat{y}] - [A, y]\| \\ \text{Subject to:} & y = Ax \end{array}$$

And finally we get to a form that makes some sense:

$$\begin{array}{rl} \text{Minimize:} & \\ \Delta A, \epsilon, x & \|[\hat{A}, \hat{y}] - [A, y]\| \\ \text{Subject to:} & [A, y] \begin{bmatrix} x \\ -1 \end{bmatrix} = 0 \end{array}$$

We can interpret this problem as looking for a matrix $[A, y]$ which is close to $[\hat{A}, \hat{y}]$ and has a null space. This sounds like a job for:

This looks like a job for

SVD form of a Matrix

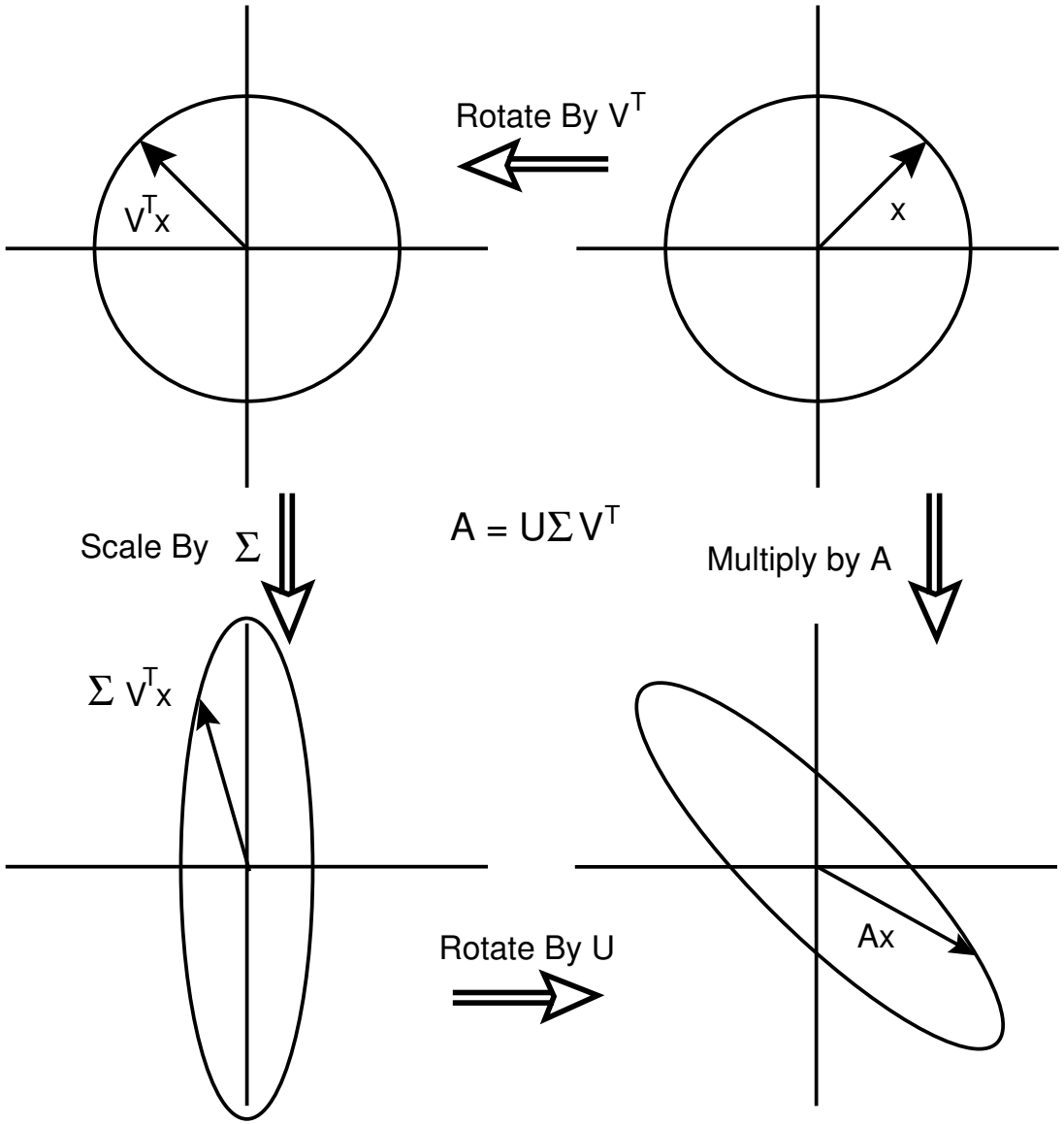
The SVD of a matrix A is defined as

$$A = U\Sigma V^T$$

where,

$$\begin{aligned}
A &\in \Re^{m\times n},\ \text{Rank } A = r \\
U &\in \Re^{m\times r},\ U^TU = I \\
V &\in \Re^{n\times r},\ V^TV = I \\
\Sigma &= \text{diag}(\sigma_1,\ \sigma_2,\ ...,\ \sigma_r\ ),\ \sigma_i \geq \sigma_{i+1} \geq 0
\end{aligned}$$

Graphical SVD



The final interpretation

$$A = U\Sigma V^T$$

$$= \begin{bmatrix} u_1 u_2 \cdots u_r \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_r^T \end{bmatrix}$$

This suggests that A may be written in *diadic* form:

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

Since the singular values are ordered, the input-output directions v_1, u_1 have the largest gain, and directions u_r, v_r have the smallest.

Where are we going with this again?

The whole point is that we want to minimize the errors in our measurements some how.

$$\begin{array}{rl} \text{Minimize:} \\ \Delta A, \epsilon, x & \|[\hat{A}, \hat{y}] - [A, y]\| \\ \text{Subject to:} & [A, y] \begin{bmatrix} x \\ -1 \end{bmatrix} = 0 \end{array}$$

Rank One Approximate of a Matrix

Since, $\sigma_i \geq \sigma_{i+1} \geq 0$, we can optimally approximate $[A, y]$ as the rank one deficient matrix (when the world is well behaved):

$$[A, y] = \sum_{i=1}^{r-1} \sigma_i u_i v_i^T$$

Singular Vector v_n is now a direction of zero gain. Ie,

$$[A, y]\, v_n = 0$$

From before, we are looking for:

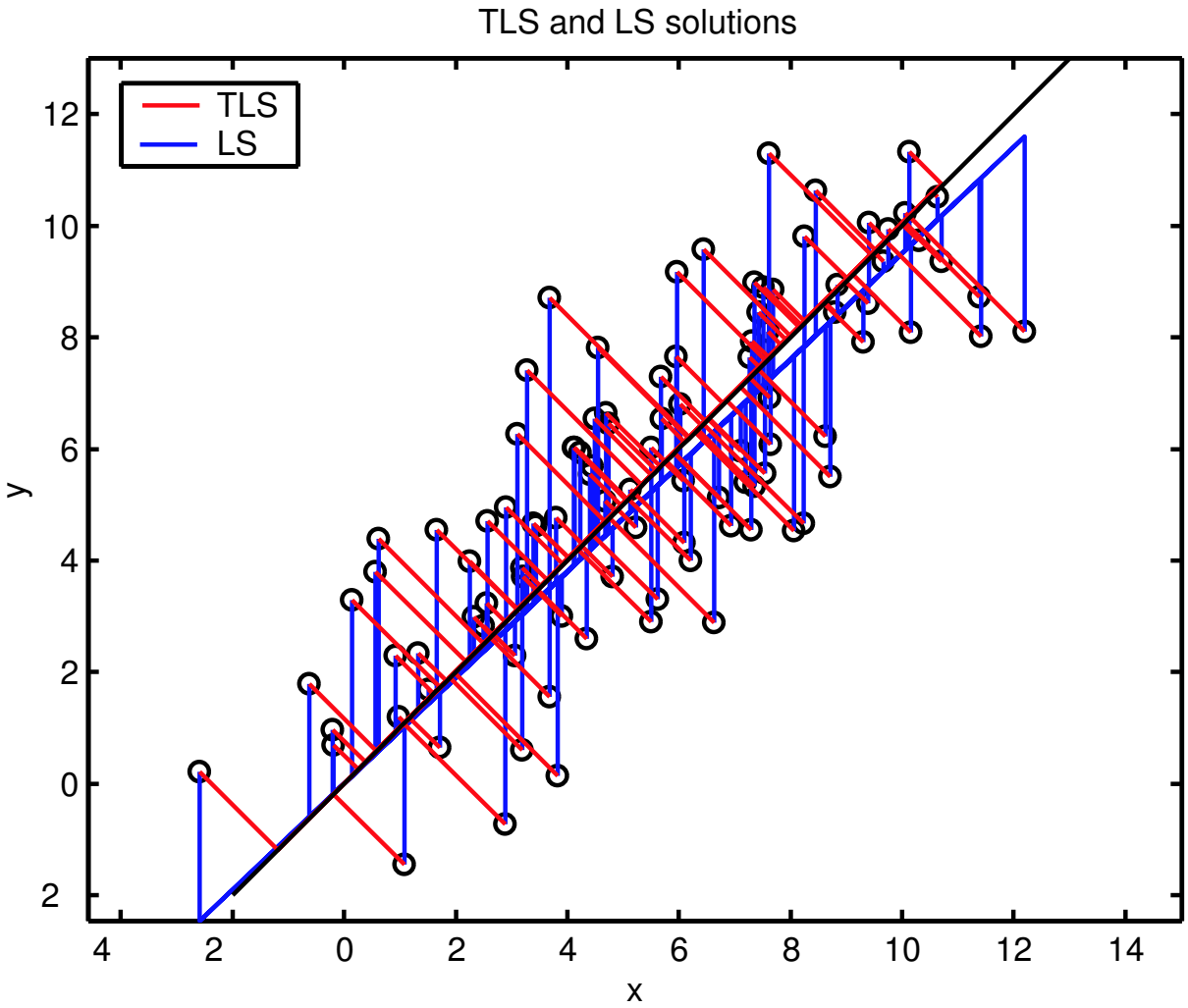
$$[A, y] \begin{bmatrix} x \\ -1 \end{bmatrix} = 0$$

$$\Rightarrow x = \frac{v_n(1 : r-1)}{-v_n(r)}$$

TLS Algorithm

1. Compose the matrix $[\hat{A}, \hat{y}]$
2. Take the SVD, strip the last singular vector v_n
3. your estimate for $x = \frac{v_n(1{:}r-1)}{-v_n(r)}$

Would you show a plot already?



Least squares minimizes the error in y, TLS minimizes the orthogonal distance to a line.

Almost There

Consider the model

$$y_i = ax_i + b$$

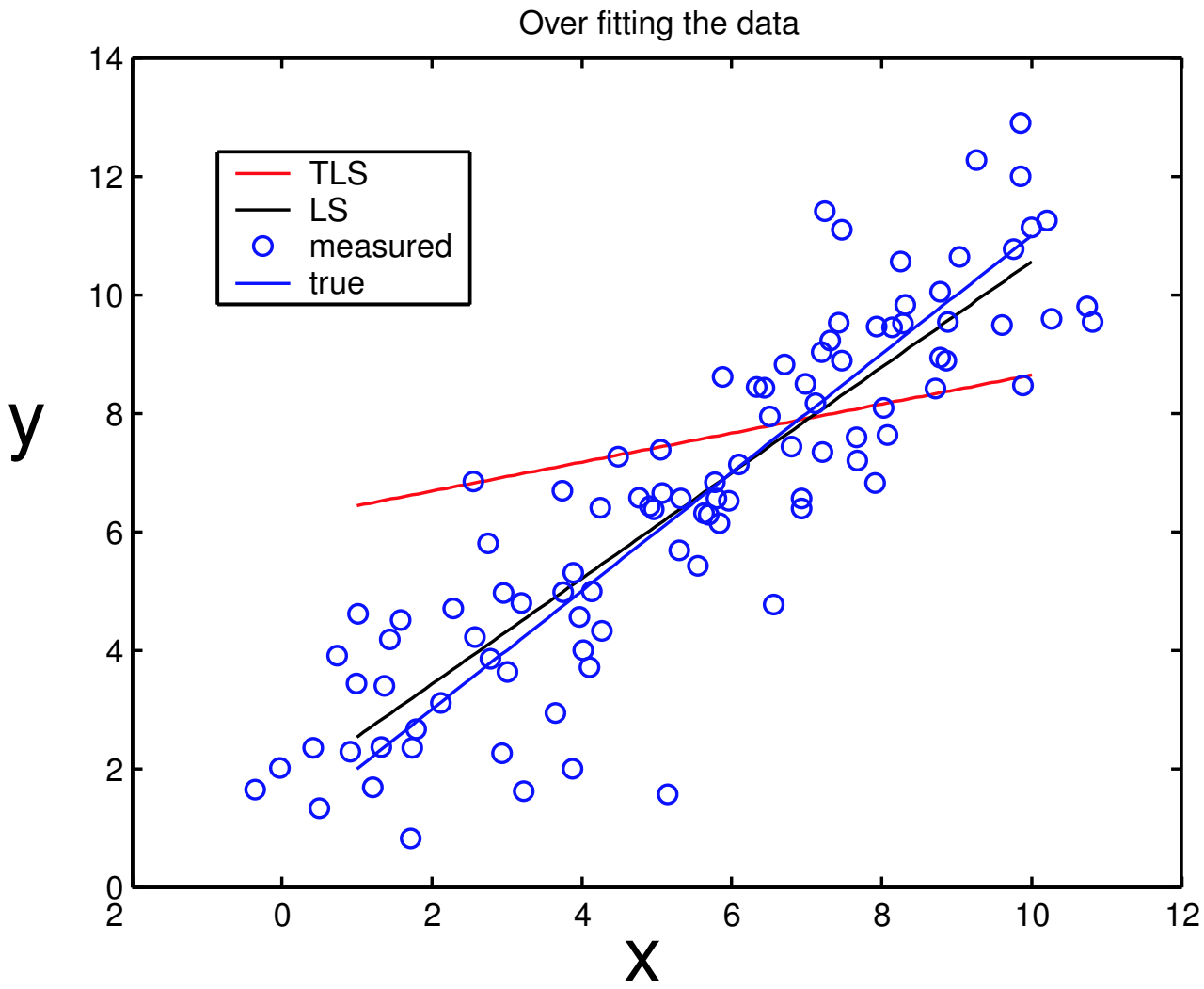
where x_i and y_i are measured with error and we wish to estimate a and b. This can be written as:

$$y = [1, x] \begin{bmatrix} b \\ m \end{bmatrix}$$

This is a problem because we *know* that the first column of A does not have any noise on it.

In this case TLS has too much freedom to fit the data.

Over Fitting data?



TLS over fits the data when one column really is noise free.

If we know a column is noise free, there are slightly more complicated algorithms for finding those solutions.

Ok, what are the new assumptions

1. We know which elements of A have no noise
2. The noise is element wise IID with equal covariance

Hold on: What do you mean the all the noises have equal covariance. How often does that happen in real life?

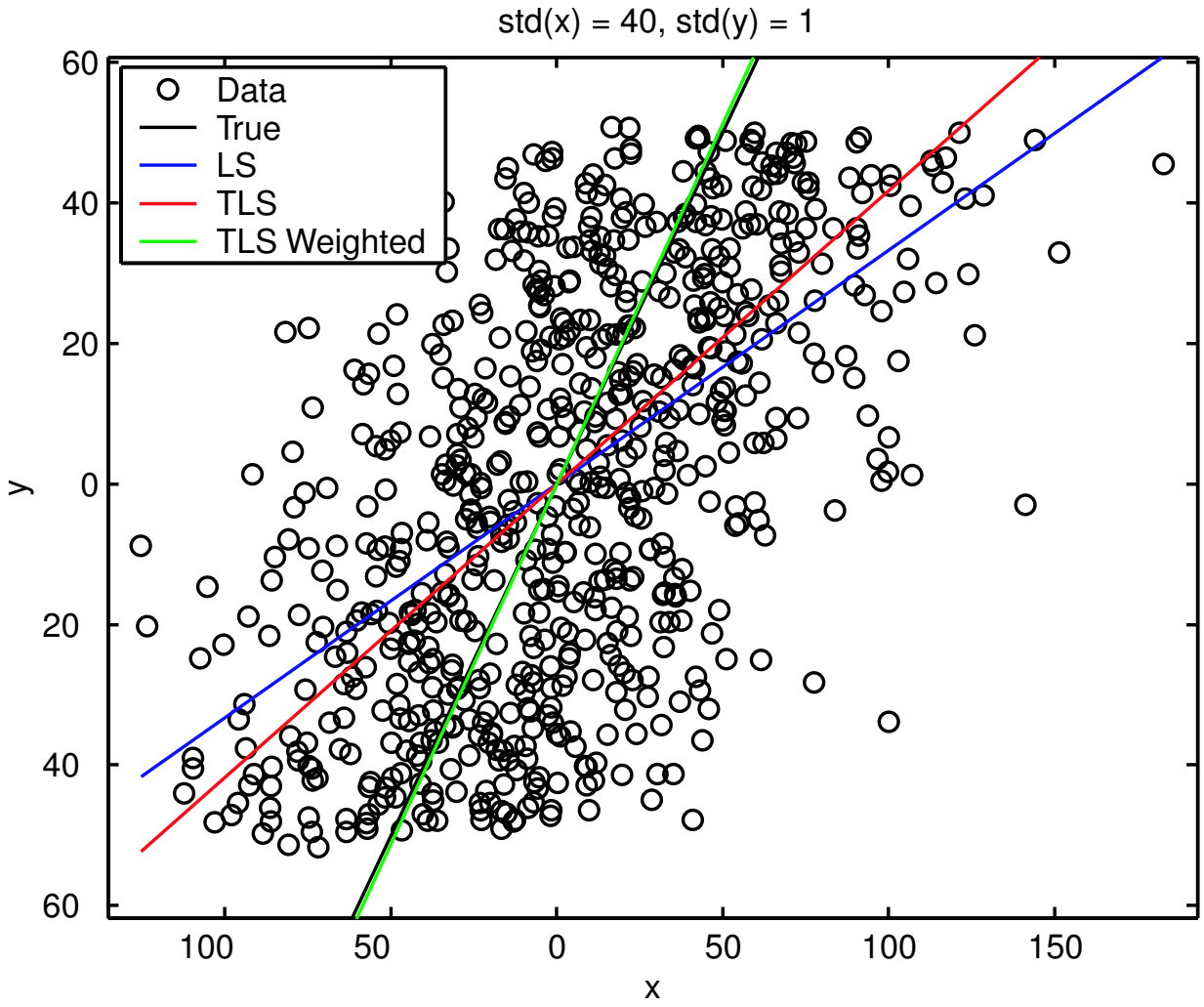
We get around this this by simply normalizing the columns:

$$[A, b]C\left(C^{-1}\begin{bmatrix} x \\ -1 \end{bmatrix}\right) \sim 0$$

Use the diagonal entries of C to weight each column by the inverse of its noise standard deviation. Geometrically, now the "perpendiculars" are slanted by the entries of C.

Even with TLS, you still have to *understand* your problem

Example of Column weighting in action



By scaling the columns to have the same size error, get a good fit of the data

Hey, that looks like those Force-Slip plots!

Algorithms

More general algorithms exist and are faster and etc. Come talk to me if you are interested.

- Noise variance changes with each sample
- Recursive solutions
- Really fast iterated solutions which assume small change in information
- Sparse solutions
- Multiple right hand sides
- etc, etc.

What if my model is nonlinear?

The problem becomes much more difficult in general.

Difficult means:

- Hard to show analytical bounds for error
- You probably need a faster computer
- You now have a winner topic for cocktail parties

Good things to check

- (local) Convexity
- Analytical gradients
- Number of points to keep things interesting

The Nonlinear Idea

Let $\boldsymbol{f}$ be the true nonlinear model:

$$\boldsymbol{f}(\boldsymbol{a}, \boldsymbol{x}) = \boldsymbol{b}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are vectors of true model values and $\boldsymbol{x}$ is a vector of parameters.

Assuming measurement errors:

$$\hat{\boldsymbol{a}} = \boldsymbol{a} + \Delta\boldsymbol{a}$$
$$\hat{\boldsymbol{b}} = \boldsymbol{b} + \Delta\boldsymbol{b}$$

$$\begin{array}{rl} \text{Minimize:} & \left\| \begin{array}{c} \Delta\boldsymbol{a} \\ \Delta\boldsymbol{b} \end{array} \right\| \\ \boldsymbol{x},\ \Delta\boldsymbol{a},\ \Delta\boldsymbol{b} & \\ \text{Subject to:} & \boldsymbol{f}(\hat{\boldsymbol{a}} - \Delta\boldsymbol{a}, x) = \hat{\boldsymbol{b}} - \Delta\boldsymbol{b} \end{array}$$

Nonlinear Least Squares

$$\underset{\boldsymbol{a},\, \boldsymbol{x}}{\text{Minimize:}} \quad \left\| \begin{array}{c} \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{x}) - \hat{\boldsymbol{b}} \\ \boldsymbol{a} - \hat{\boldsymbol{a}} \end{array} \right\|$$

Solutions to these problems iteratively approximate the nonlinear function as quadratic and solve a local linear least squares problem. If we let,

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{a} \end{bmatrix}$$

$$\boldsymbol{g}(\boldsymbol{\Theta}) = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{a}, \boldsymbol{x}) - \hat{\boldsymbol{b}} \\ \boldsymbol{a} - \hat{\boldsymbol{a}} \end{bmatrix}$$

Then iteratively we solve the problem,

$$\boldsymbol{J}^{\boldsymbol{i}} = \left.\frac{\partial \boldsymbol{g}(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}}\right|_{\boldsymbol{\Theta}^{\boldsymbol{i}}}$$

$$\boldsymbol{\Theta}^{i+1} = \boldsymbol{\Theta}^{i} + \alpha \boldsymbol{J}^{\dagger} \boldsymbol{g}^{i}(\boldsymbol{\Theta}^{i})$$

where i refers to the iteration number, $\dagger$ represents the least squares pseudoinverse and $0 < \alpha < 1$ is the backstepping parameter.

Structure

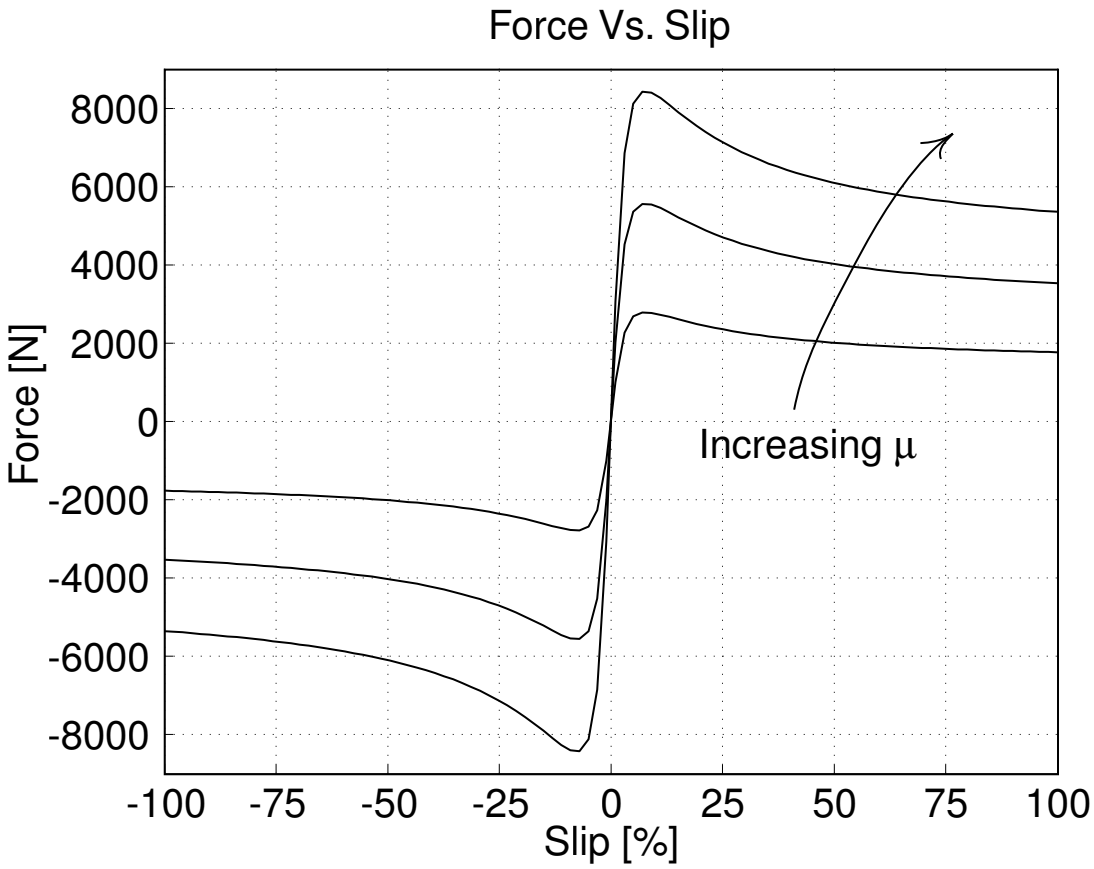
The gradient of the NLLS equation with respect to the regressors $\boldsymbol{\Theta} = [\boldsymbol{a}^T, \boldsymbol{x}^T]^T$ has the structure,

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial \boldsymbol{f}(\boldsymbol{a},\boldsymbol{x})}{\partial \boldsymbol{a}} & \frac{\partial \boldsymbol{f}(\boldsymbol{a},\boldsymbol{x})}{\partial \boldsymbol{x}} \\ \frac{\partial(\boldsymbol{a}-\hat{\boldsymbol{a}})}{\partial \boldsymbol{a}} & \frac{\partial(\boldsymbol{a}-\hat{\boldsymbol{a}})}{\partial \boldsymbol{x}} \end{bmatrix}$$

$$= \begin{bmatrix} B_{n\times n} & D_{n\times 2} \\ I_{n\times n} & 0_{n\times 2} \end{bmatrix}$$

The banded matrices here mean we have a ton of zeros. Since we know where they are, we can easily modify algorithms to skip them.

My Pet Nonlinear Problem



$$F = C_x \left(\frac{V - R\omega}{V} \right)$$

Force Vs. slip approximately linear for low values of slip

This model is nonlinear in the measurements

If I use the NL technique, my estimator is *independent* of sample time

Rewrite Force Vs. Slip Equations

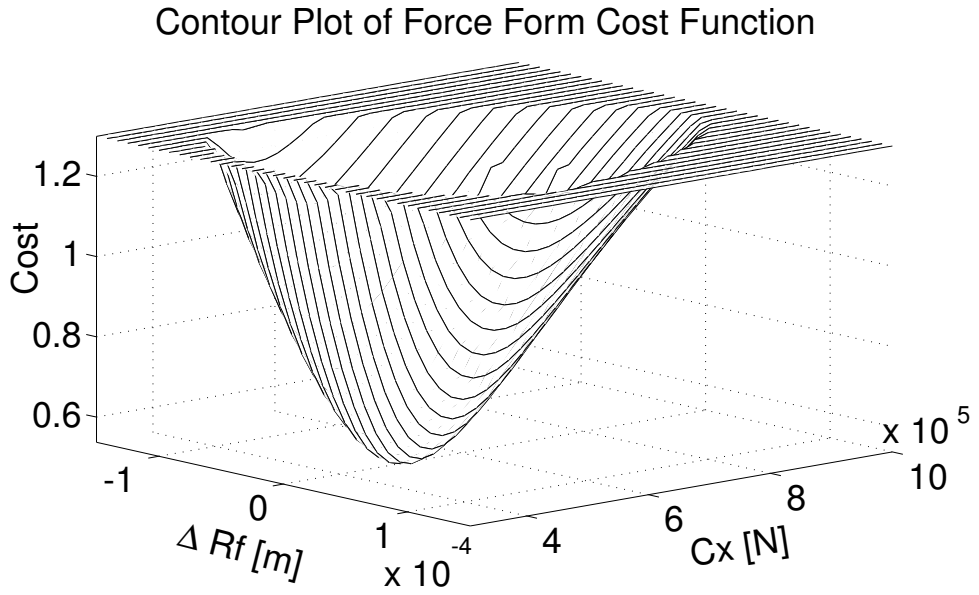
Write out the measurement errors explicitly

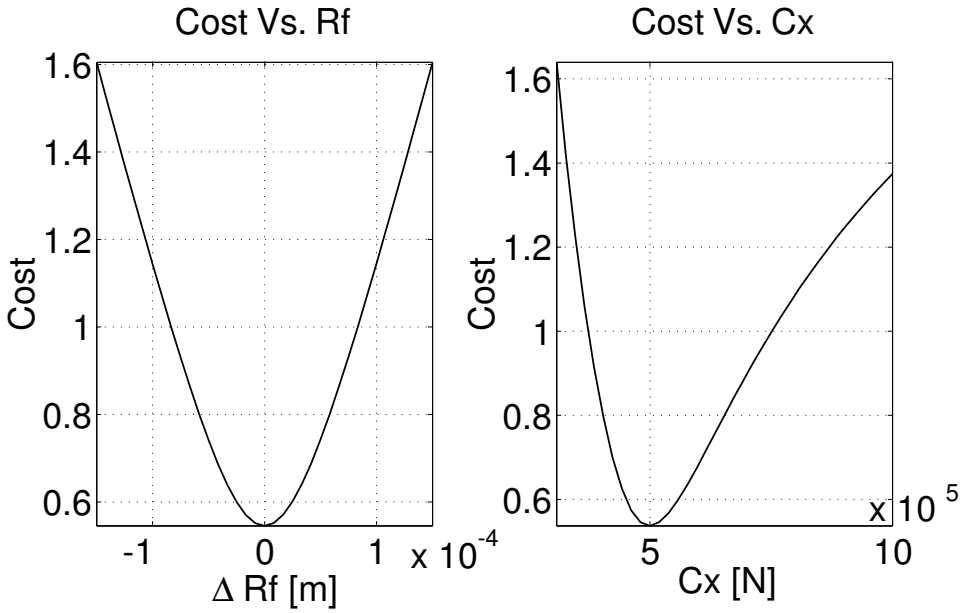
Minimize:
R_r, C_x $\quad \|\Delta\theta_r; \Delta\theta_f\|$

Subject to:

$$mR_r(\theta_r + \ddot{\Delta\theta_r}) = -C_x \left(\frac{R_r(\theta_r \dot{+} \Delta\theta_r) - R_f(\theta_f \dot{+} \Delta\theta_f)}{R_r(\theta_r \dot{+} \Delta\theta_r)} \right)$$

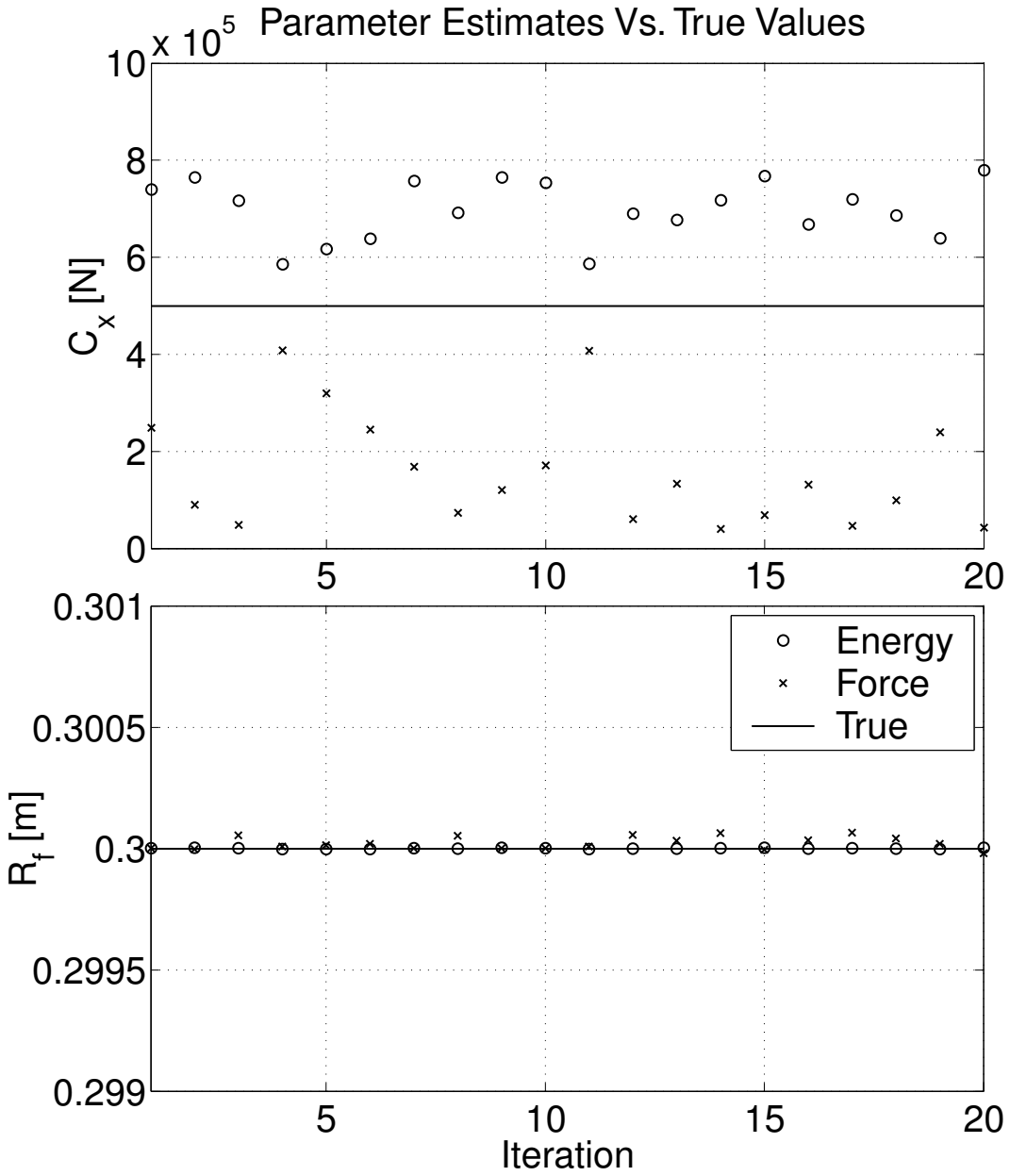
Cost Surfaces are Quasiconvex





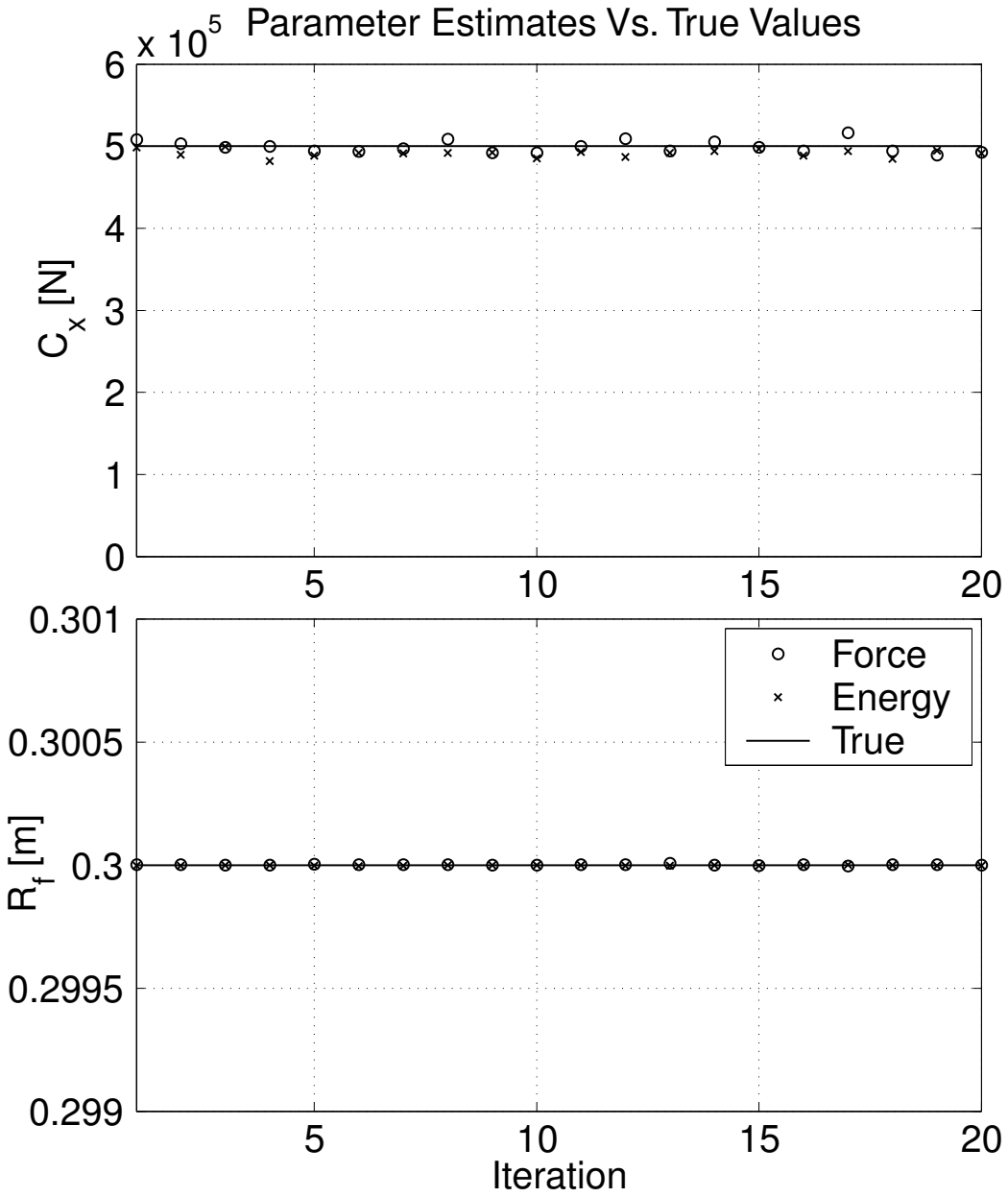
Locally quasiconvex cost function insures unique solution for parameter estimates

Linear Simulation Results



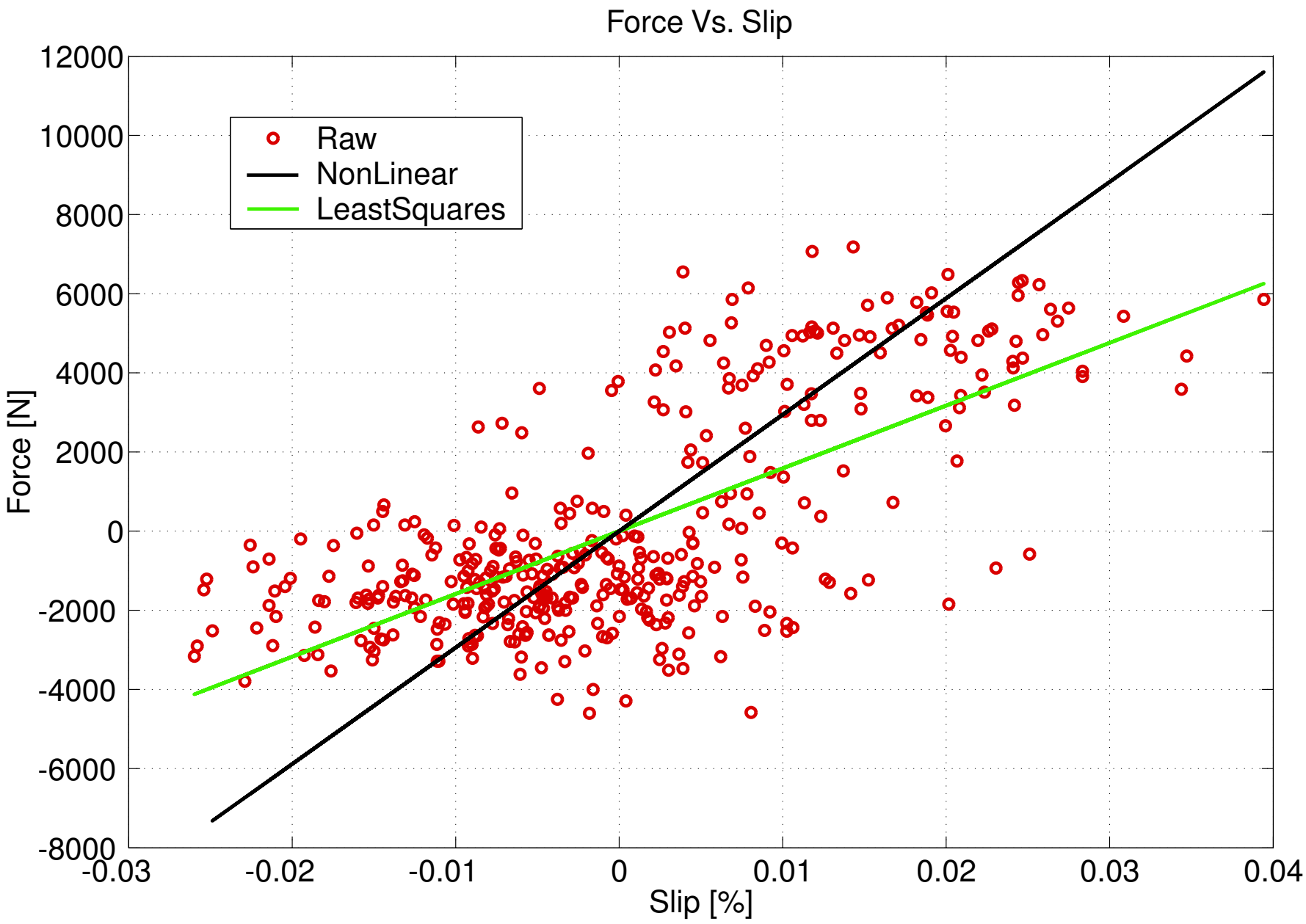
Linear LS returns biased parameter estimates in simulation

Simulation Results

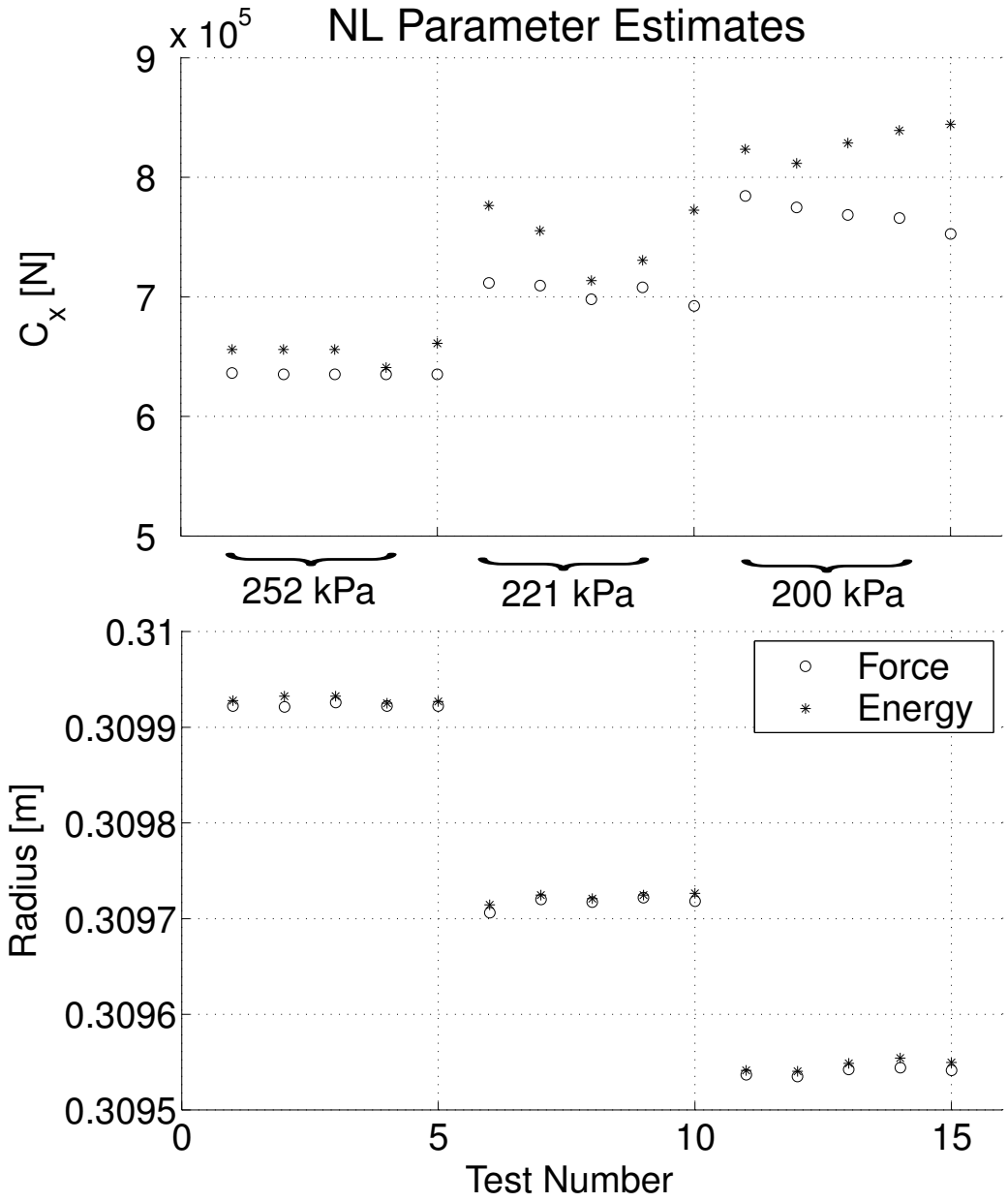


Nonlinear TLS returns much more consistent parameter estimates

Application Results 2



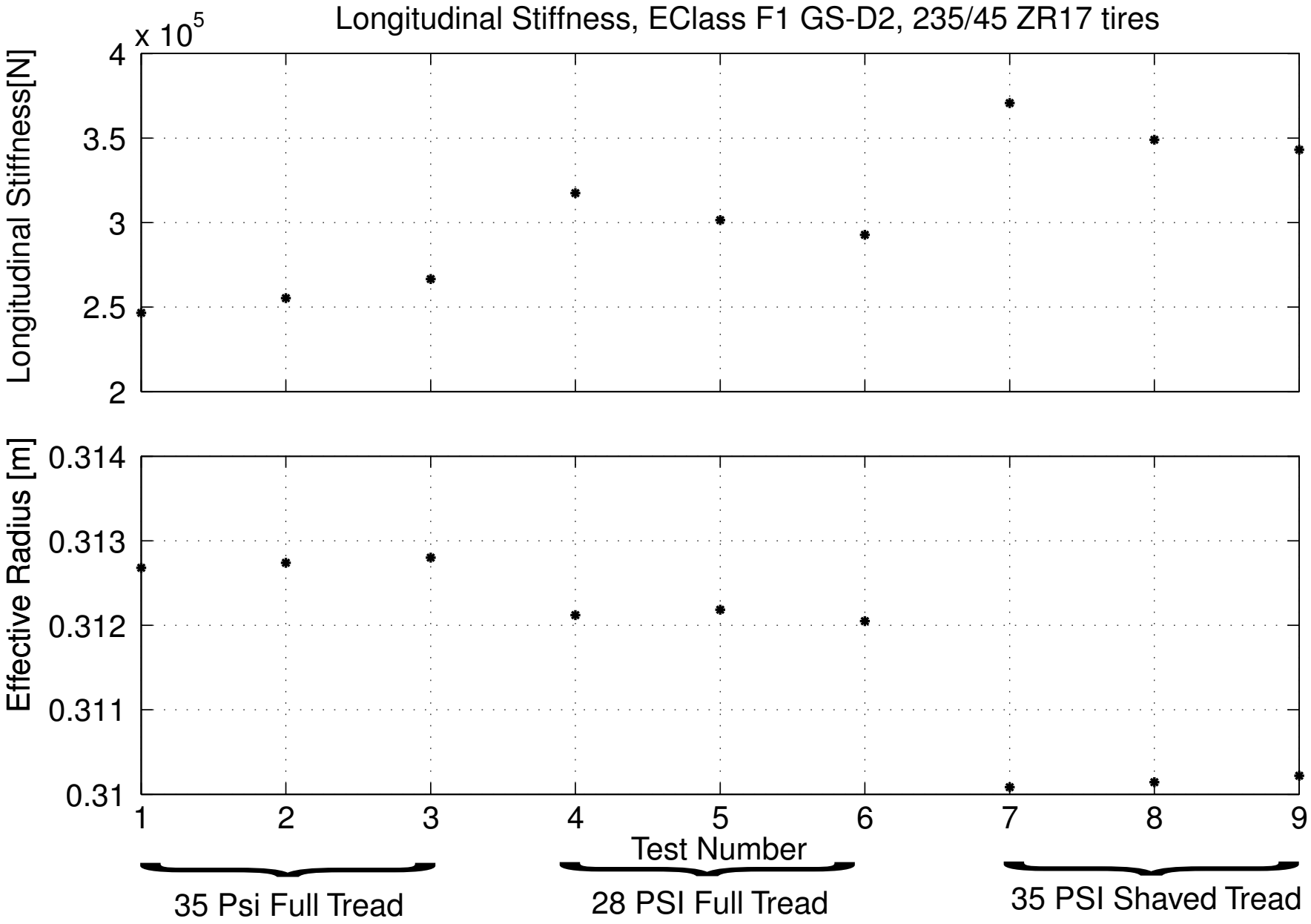
Test Data



Test data for three different tire pressures.

Each data point represents 60-90 seconds of data

Test Data 2


Longitudinal Stiffness, EClass F1 GS-D2, 235/45 ZR17 tires
x 10^5
Longitudinal Stiffness[N]
4
3.5
3
2.5
2
Effective Radius [m]
0.314
0.313
0.312
0.311
0.31
1
2
3
4
5
6
7
8
9
Test Number
35 Psi Full Tread
28 PSI Full Tread
35 PSI Shaved Tread

Conclusions

- We saw the TLS parameter estimator presented from linear algebra basics
- Like LS, TLS is a neat and mostly easy to use tool
- Like NLLS, NLTLS sometimes works really well
- Like LS, beware the model assumptions, they can kill!