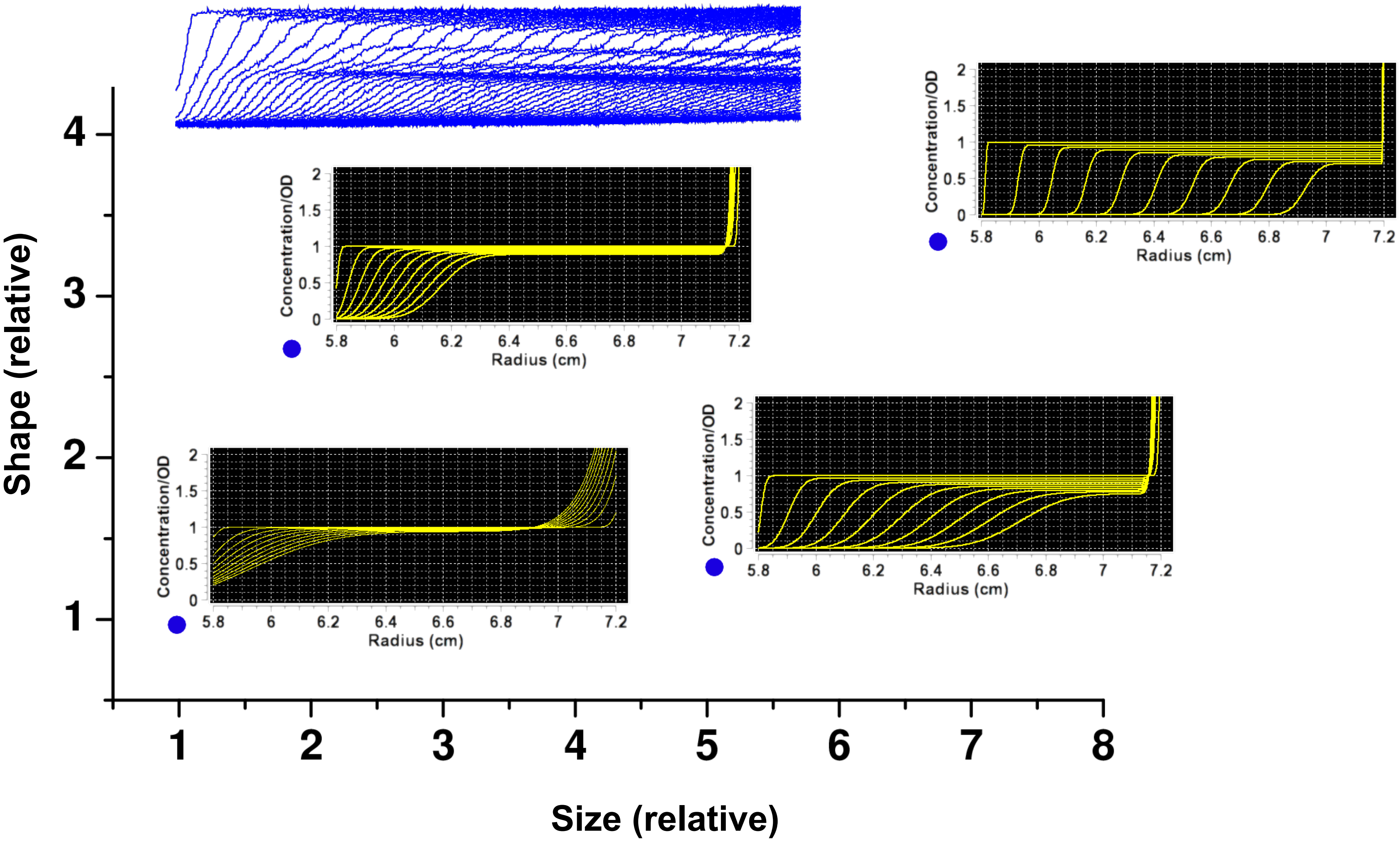
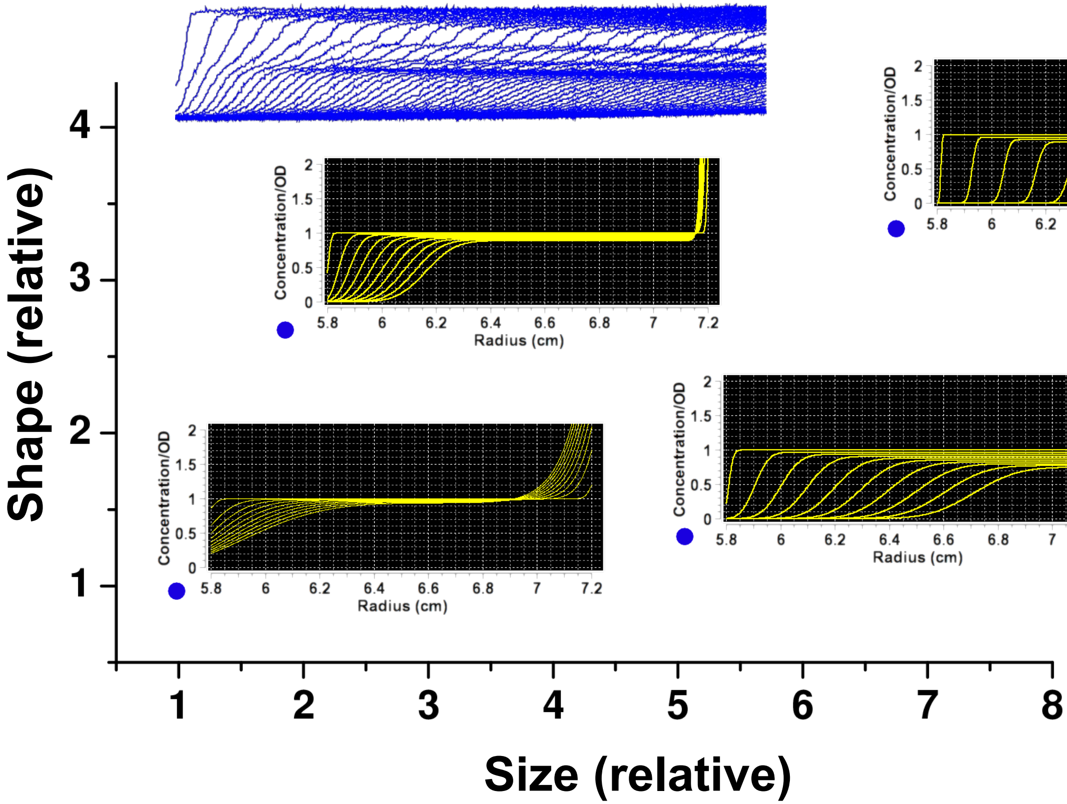
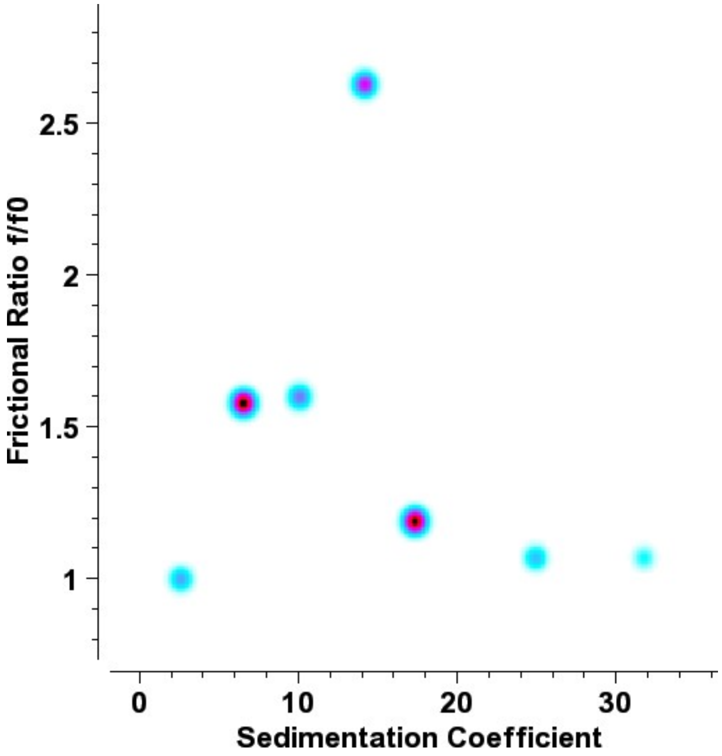
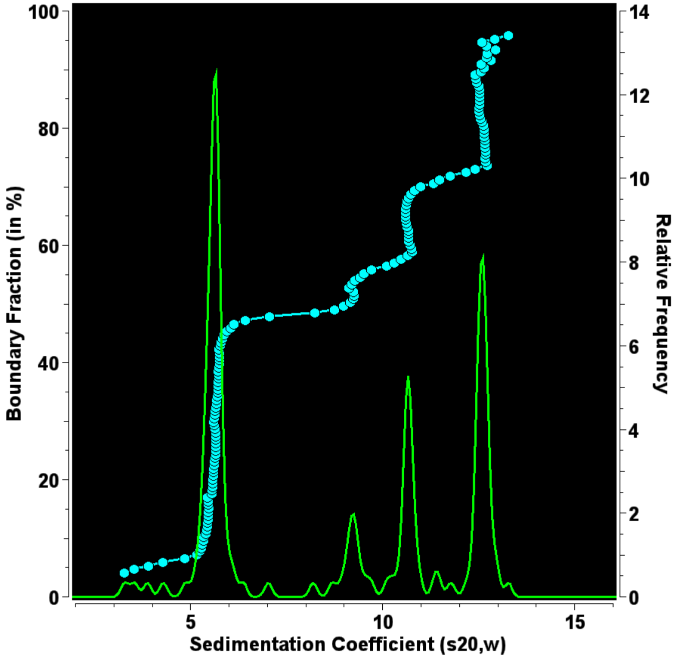
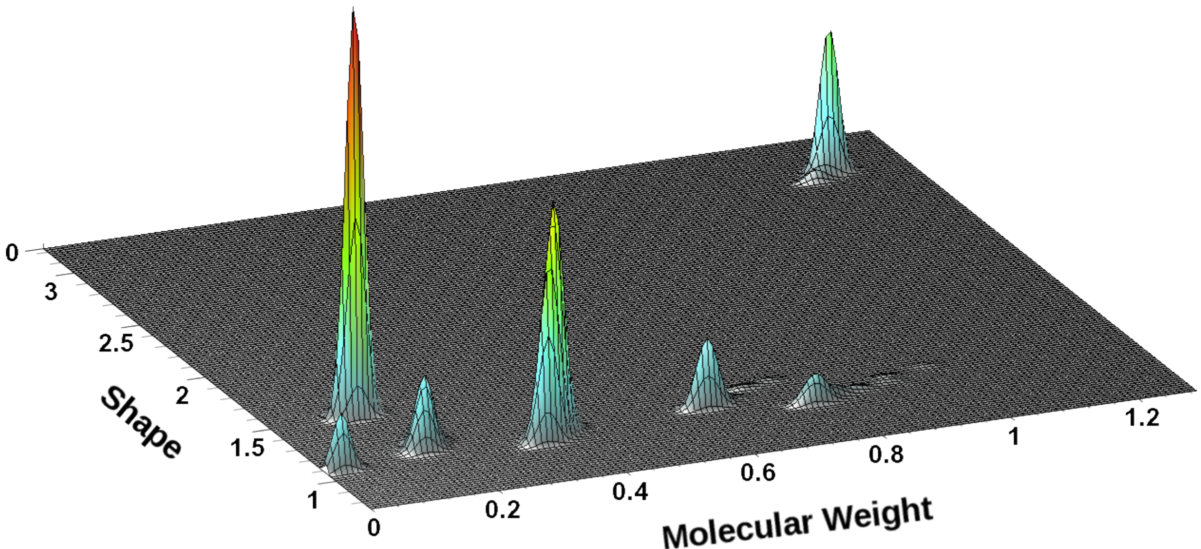


Find: partial concentration, *s, D, MW, Kd,* k_{off} and f/f_0


Shape (relative)
4
3
2
1
Size (relative)
1
2
3
4
5
6
7
8
Concentration/OD
Radius (cm)
5.8
6
6.2
6.4
6.6
6.8
7
7.2
0
0.5
1
1.5
2


Frictional Ratio f/f0
2.5
2
1.5
1
Sedimentation Coefficient
0
10
20
30


Boundary Fraction (in %)
100
80
60
40
20
0
Relative Frequency
14
12
10
8
6
4
2
0
Sedimentation Coefficient (s20,w)
5
10
15


0
3
2.5
2
1.5
1
Shape
0
0.2
0.4
0.6
0.8
1
1.2
Molecular Weight

Survey of Numerical Modeling

We will touch on the following subjects:

- Model Building
  - Exact models
  - smoothing
- Theory of Fitting
  - What is a good fit and how can it be measured?
- Parameter Estimation
- Optimization – how do I fit the model to the data?
  - Linear vs. non-linear least squares
  - Linearization of non-linear systems
  - Brute force methods – Grid searches
  - Stochastic Methods – Genetic algorithms
- Effect of noise on analysis
- Statistical analysis and Monte Carlo approaches
- Parallel Implementation

Data Fitting

Modeling involves the description of some *observable data* (experimental measurements) using a mathematical *equation* that describes the underlying physical properties of the experiment.

First, we need to identify a general, mathematical model that can represent the observed data. The *parameters* of the model describe the specifics of the data.

Second, we need to determine the *values* of the parameters in the model that best fit our data. This is accomplished by a fitting algorithm that minimizes the difference between the data and model. Generally, an initial estimate is required that is then improved.

Finally, we need to estimate the error in the parameters we determined in the fitting process and obtain the *confidence intervals*.

How to build a Model

To build a model, one needs to understand the physical properties of the observed process. Many processes can be described by differential equations. When solved, these equations describe a linear or nonlinear model:

Example – radioactive decay:

Hypothesis: The rate of decay is proportional to the number of nuclei present.

The hypothesis can be formulated as a differential equation that describes the anticipated change:

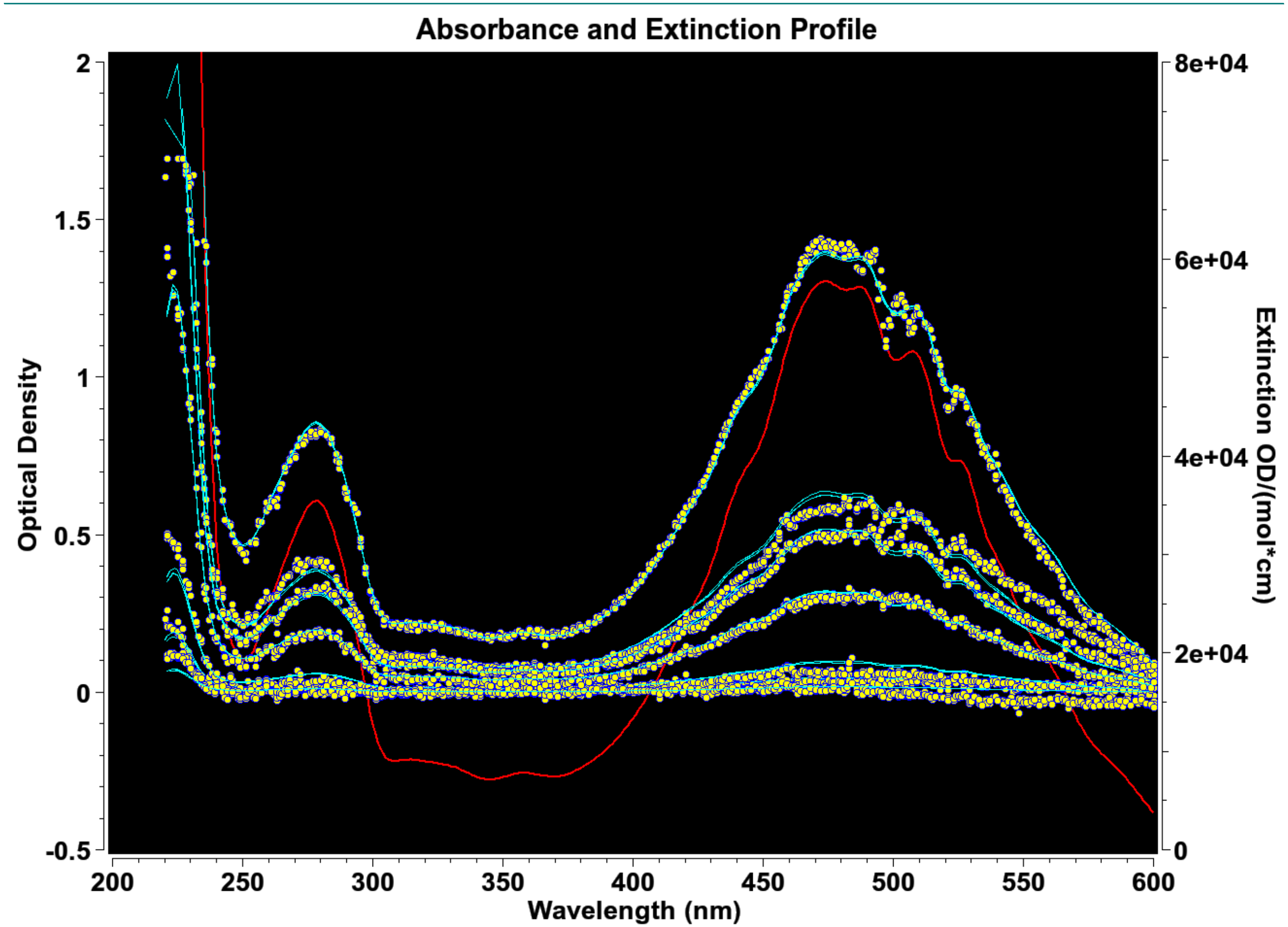
$$\frac{\partial N}{\partial t} = -a\ N$$

Integrate the differential equation over n and t: $\int_{0}^{n} \frac{dN}{N} = -a \int_{t_0}^{t} dt$

With the solution: $N(t) = N_0\ e^{-a(t-t_0)} + b$

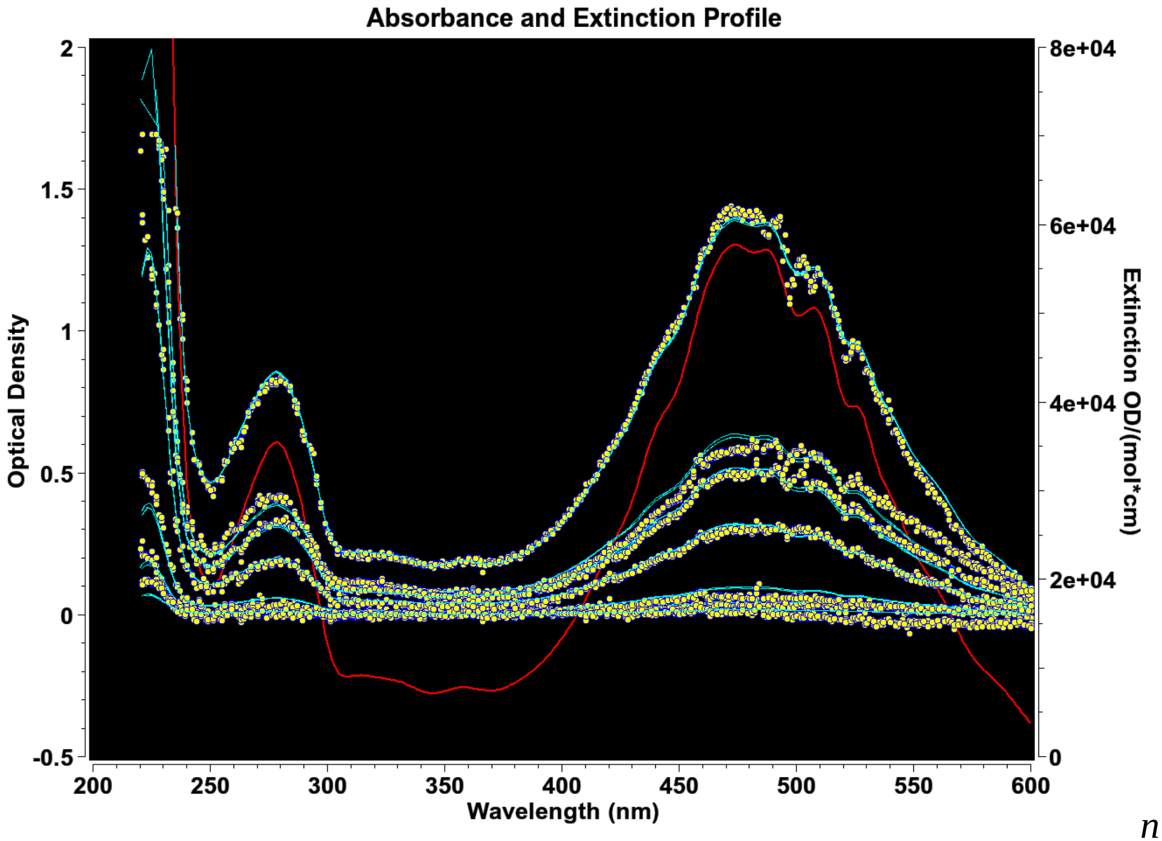
Giving rise to fitting parameters *N₀*, a and b

You start with some experimental data...



Absorption data from multiple concentrations fitted to a sum of Gaussian functions

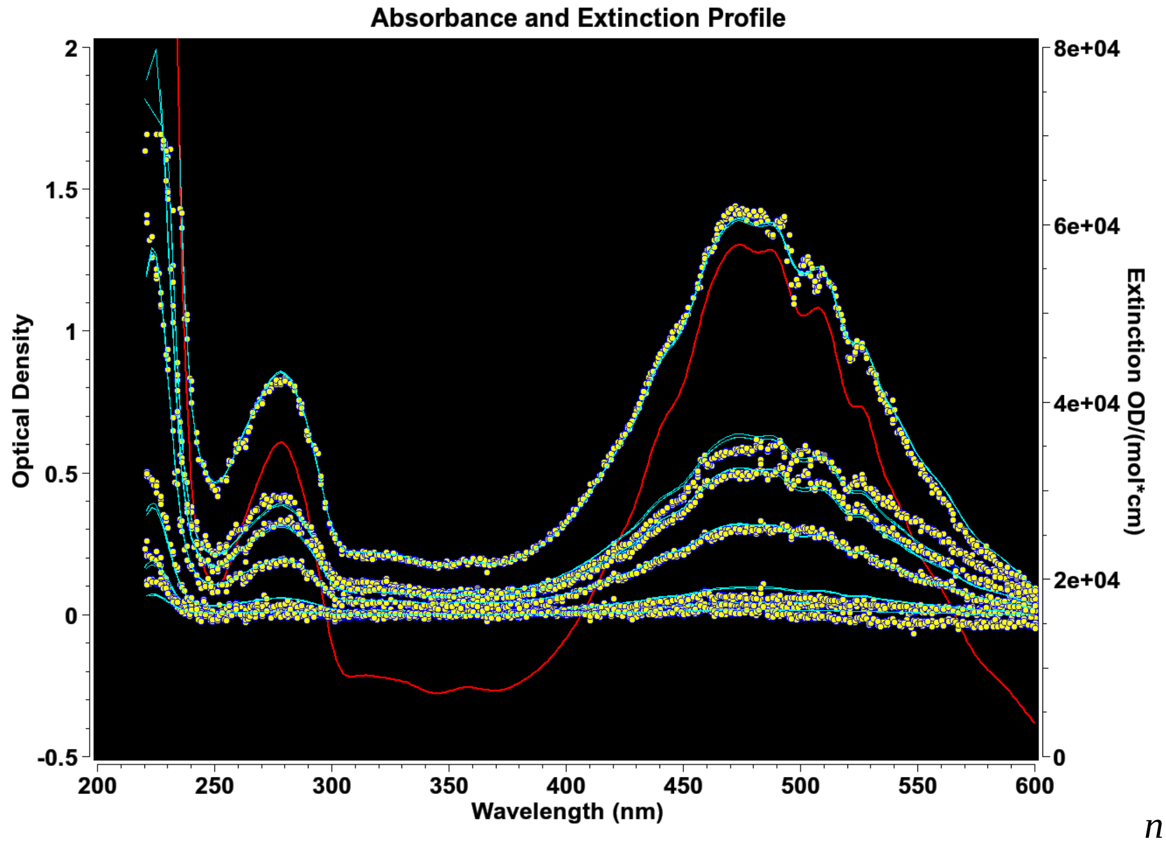
You start with some experimental data...



$$OD = a\sum_{i=1}^{n} c_i e^{\left[\frac{-(x_i-\mu_i)^2}{2\sigma_i^2}\right]} + b$$

Absorption data from multiple concentrations fitted to a sum of Gaussian functions

You start with some experimental data...



Parameters aren't really useful here (except outer amplitudes), model is only used to find a smoothing shape that can be scaled to the appropriate concentration

$$OD = a\sum_{i=1}^{n} c_i e^{\left[\frac{-(x_i-\mu_i)^2}{2\sigma_i^2}\right]} + b$$

Absorption data from multiple concentrations fitted to a sum of Gaussian functions

Method of Least Squares

Fitting Data to a Model by the Method of Least Squares:

Any observable process that influences the measurement needs to be accounted for in order for the model to yield meaningful results. The object is then to minimize the residuals between the model and the data:

$$MIN \sum_{i=1}^{n} \left(\frac{Data_i - (Model_i)}{uncertainty_i} \right)^2 = MIN \ \chi^2$$

Extracting parameters from a simulated solution by fitting the model to experimental data is called an *"Inverse Problem"*

Methods of Least Squares

$$MIN \sum_{i=1}^{n} \left(\frac{Data_i - (Model_i)}{uncertainty_i} \right)^2 = MIN \ \ \chi^2$$

Assumptions made in the Method of Least Squares:

- **The model is a truthful representation of reality**
- **All error is associated with the dependent variable. We can scale the reliability of each observation with an uncertainty factor σ_i.**
- **All experimental noise is considered to be of Gaussian distribution**

A non-parametric fit is used to smooth data for display, where the intrinsic model is of little interest, and hence the parameters may or may not be useful.

First-principles methods have meaningful parameters related to physical properties of the molecules, because the models are design to describe physical processes (light scattering, sedimentation, diffusion, etc).

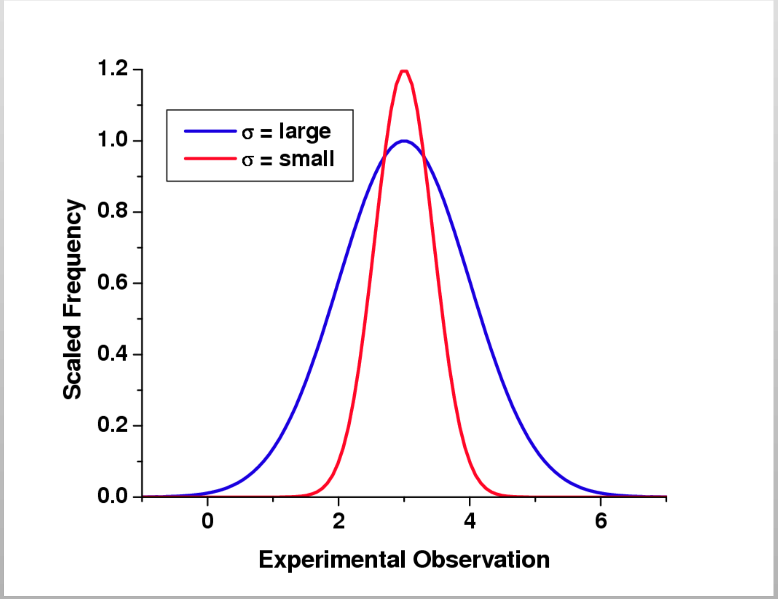
Experimental Uncertainties

The uncertainty of a measurement can be determined by repeating the experiment several times. Each time, a slightly different value is obtained for the experimental observation. Assuming a Gaussian distribution of errors in the measurement, one can determine the standard deviation σ of the distribution of measurement values, and use σ to set error bars on a measurement and to scale the contribution of a datapoint to the sum of the residuals.

The standard deviation can be calculated by using this formula:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}$$

where $\bar{x}$ is the average of all measurements.



Fitting Basics

For a straight line
we have: y = a + bx

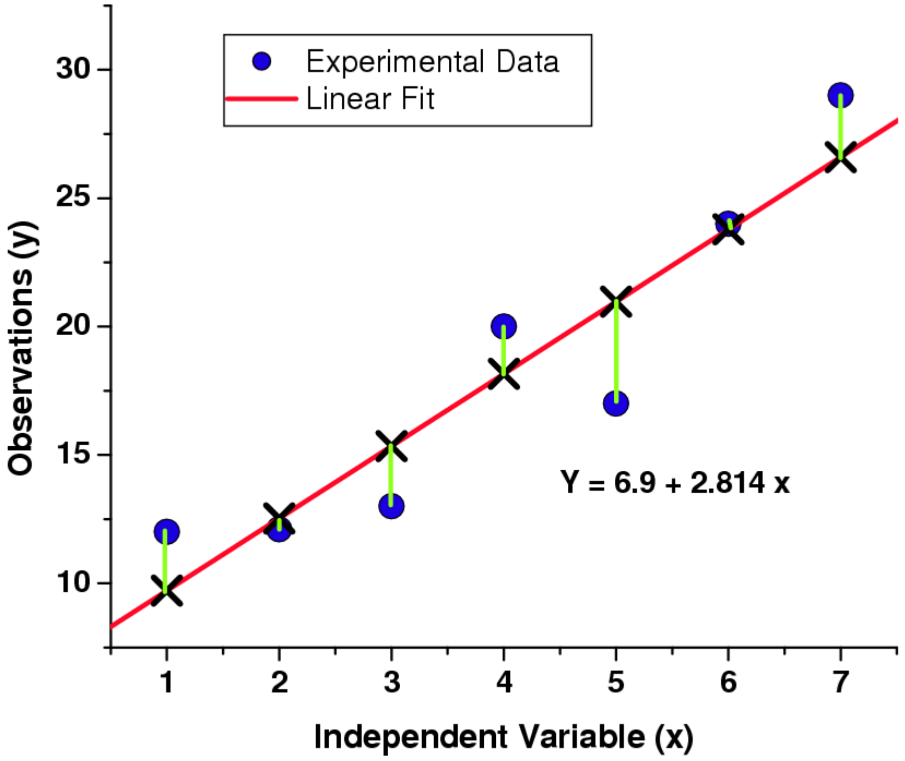
The least squares equation
Is given by:

$$\sum_{i=1}^{n}\left(\frac{D_i - M_i}{\sigma_i}\right)^2$$

The distances
are measured
perpendicular
to the data.

The object is to
find the equation
of the straight line
that minimizes the
distance between
the straight line and
the data points.

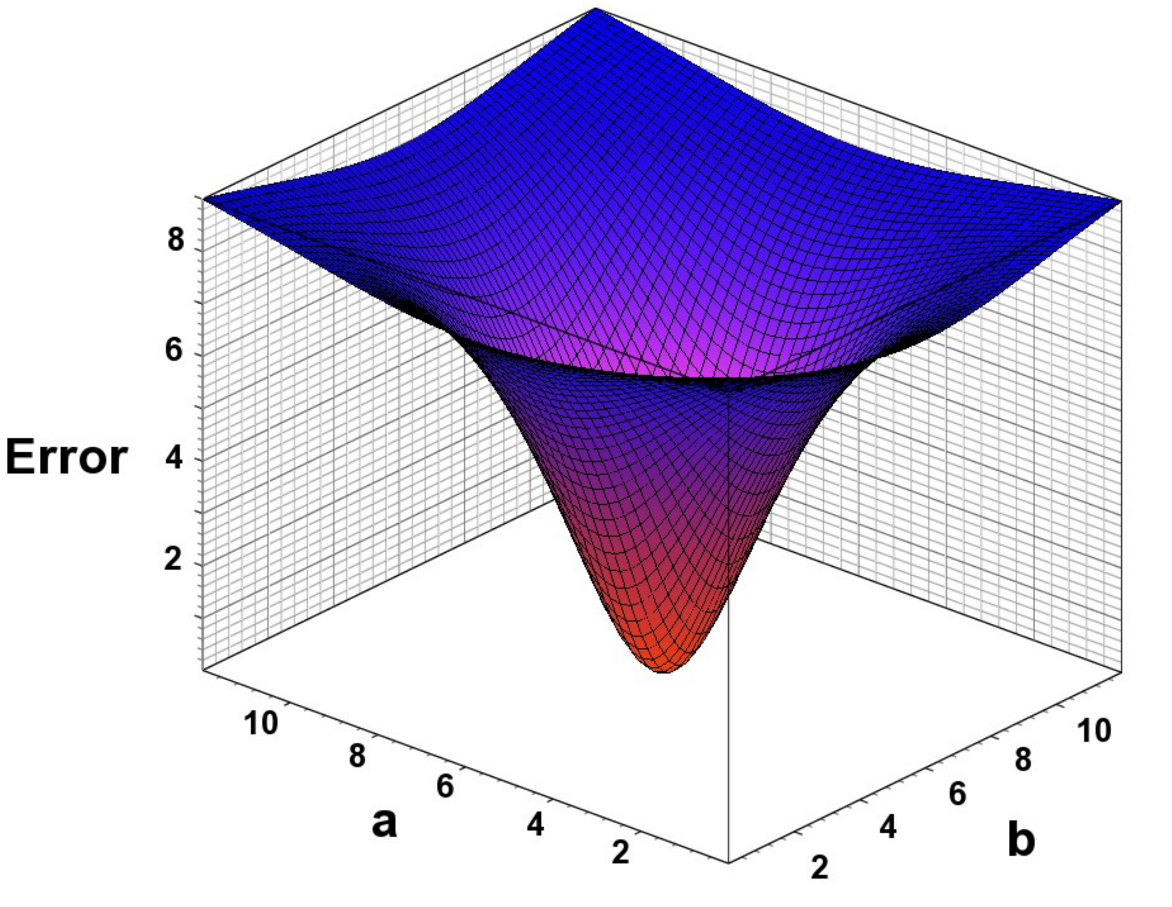
$$\sum_{i=1}^{n}\left(\frac{D_i - a - b\,x_i}{\sigma_i}\right)^2 = \chi^2(a,b)$$



Fitting Basics

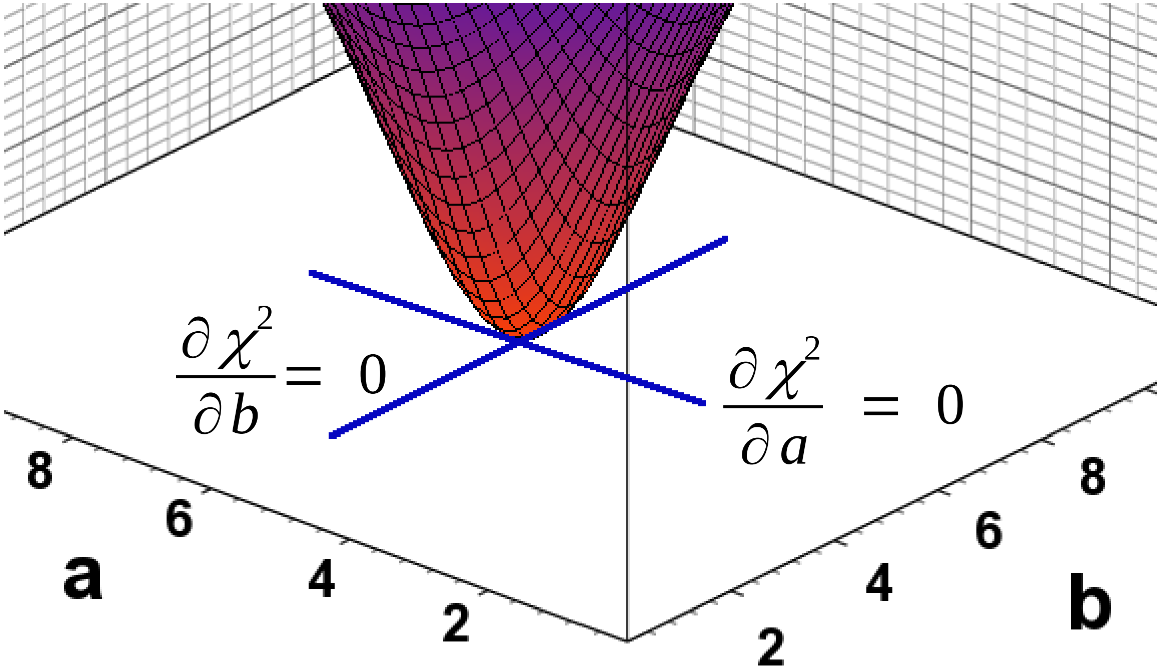
What is an error surface?

Each parameter combination a, b results in a unique error when fitted to the experimental data.
The optimal solution occurs where the error is the smallest.
Ideally, the error surface is continuously differentiable.



Error surface for some function *y = F(a, b)*

Solving the Least Squares Equation



The minimum in the differences occur where the derivative of our objective function with respect to the parameters is zero, so we need to differentiate it with respect to the parameters of interest, a and b, and set the derivative to zero, and find the values of a and b that satisfy this requirement.

$$\sum_{i=1}^{n} \left(\frac{D_i - (a + b\,x_i)}{\sigma_i} \right)^2 = \chi^2(a, b)$$

$$\frac{\partial \chi^2}{\partial a} = 0 = -2 \sum_{i=1}^{n} \left(\frac{D_i - a - b\,x_i}{\sigma_i^2} \right)$$

$$\frac{\partial \chi^2}{\partial b} = 0 = -2 \sum_{i=1}^{n} \left(\frac{x_i (D_i - a - b\,x_i)}{\sigma_i^2} \right)$$

Solving the Least Squares Equation

This leads to a system of linear equations:

$$\sum_{i=1}^{n} \frac{a}{\sigma_i^2} + \sum_{i=1}^{n} \frac{b\,x_i}{\sigma_i^2} = \sum_{i=1}^{n} \frac{D_i}{\sigma_i^2}$$

$$\sum_{i=1}^{n} \frac{a\;x_i}{\sigma_i^2} + \sum_{i=1}^{n} \frac{b\,x_i^2}{\sigma_i^2} = \sum_{i=1}^{n} \frac{D_i\;x_i}{\sigma_i^2}$$

Or, in matrix form:

$$\begin{bmatrix} \sum_{i=1}^{n} \frac{1}{\sigma_i^2} & \sum_{i=1}^{n} \frac{x_i}{\sigma_i^2} \\ \sum_{i=1}^{n} \frac{x_i}{\sigma_i^2} & \sum_{i=1}^{n} \frac{x_i^2}{\sigma_i^2} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \frac{D_i}{\sigma_i^2} \\ \sum_{i=1}^{n} \frac{D_i x_i}{\sigma_i^2} \end{bmatrix}$$

Solving the Least Squares Equation

Let:

$$A = \begin{bmatrix} \sum_{i=1}^{n} \frac{1}{\sigma_i^2} & \sum_{i=1}^{n} \frac{x_i}{\sigma_i^2} \\ \sum_{i=1}^{n} \frac{x_i}{\sigma_i^2} & \sum_{i=1}^{n} \frac{x_i^2}{\sigma_i^2} \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \text{and } B = \begin{bmatrix} \sum_{i=1}^{n} \frac{D_i}{\sigma_i^2} \\ \sum_{i=1}^{n} \frac{D_i x_i}{\sigma_i^2} \end{bmatrix}$$

(where $x_1 = a$ (i.e., intercept) and $x_2 = b$ (i.e., slope))

In matrix notation: AX = B, with solution $A^{-1}AX = A^{-1}B = X$

The equations can be solved either by inverting A or by using Cramer's Rule:

$$x_1 = \frac{b_1 a_{22} - b_2 a_{12}}{a_{11} a_{22} - a_{12} a_{21}}, \quad x_2 = \frac{b_2 a_{11} - b_1 a_{21}}{a_{11} a_{22} - a_{12} a_{21}}$$

Goodness of fit

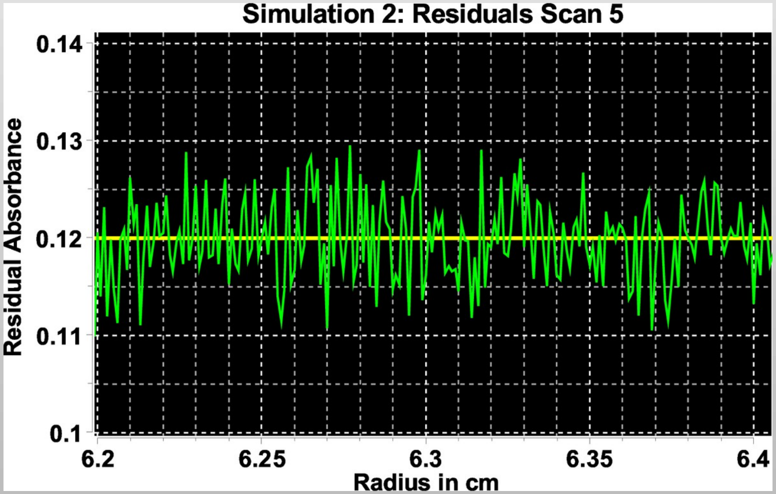
The quality of the fit is determined by the randomness of the residuals and the root mean square deviation (RMSD).

The randomness of the residuals can be measured by determining the runs. Runs (R) are the number of consecutive positive (*p*) or negative (*n*) residuals from the mean.

$$R_T = \frac{R-\overline{R}}{\sigma_R}$$

$$\overline{R} = \frac{2np}{n+p}+1$$

$$\sigma_R^2 = \frac{2np(2np-n-p)}{(n+p)^2(n+p-1)}$$



The *R*$_T$ value is a measure of randomness and can be compared to a normal table to find out the probability of the test being random

Linear Models

The equation of a straight line is considered a *linear equation*:

$$y = a + bx$$

This equation is *linear* in the coefficients that are fitted. The equation doesn't have to be that of a straight line to be considered linear.

$$y = a + bx + cx^2 + dx^3 + ex^4 + \ldots$$

As long as the coefficients are linear, the equation is considered a linear fitting equation, no matter how wildly nonlinear the terms of the independent variable are:

$$y = a + b(x - \sin(x^3)) + c\, e^{-(4-3x)} + d \ln(3x^4) + \ldots$$

In general, we can write for any linear equation:

$$y = a_1X_1 + a_2X_2 + a_3X_3 + a_4X_4 + \ldots$$

where X_i can be any nonlinear term.

Linear Models

Linearization of a nonlinear equation:

Turn $y = ae^{bx}$ **into a linear function of the form:**

$$\mathbf{y = a + bx}$$

take log on both sides:

Fitting the log of y reduces the nonlinear equation to a linear equation, y* = a* + bx, where y* = ln y and a* = ln a.

$$\ln y = \ln a + bx$$

This is not possible when a baseline is added:

$$y = ae^{bx} + c$$

Parameter Constraints

Sometimes, we may want to constrain the value of a parameter – for example, we don't want the amplitude of an exponential to turn negative during fitting:

$$y = a\, e^{bx} + c = e^{\ln(a) + bx} + c$$

By making the transformation to fitting the log of a number we can assure that the number itself will never be negative (negative amplitudes don't make sense in many physical models).

Nonlinear Regression

Why is it such a big deal if an equation is linear or nonlinear? It turns out that nonlinear functions need to be fitted using iterative approaches, while linear functions can be fitted in a single iteration, so it helps to have the objective function in a linear form. For nonlinear systems, a Jacobian is defined.

The idea: Iteratively improve the parameter estimates by following along the gradient of the error function in the direction of maximum "improvement". This requires knowledge of the partial derivatives for each parameter at each point in the experiment. We build the Jacobian matrix:

$$J = \begin{bmatrix} \left(\frac{\partial X}{\partial a_1}\right)_{x1} & \left(\frac{\partial X}{\partial a_2}\right)_{x1} & \cdots & \left(\frac{\partial X}{\partial a_n}\right)_{x1} \\ \left(\frac{\partial X}{\partial a_1}\right)_{x2} & \left(\frac{\partial X}{\partial a_2}\right)_{x2} & \cdots & \left(\frac{\partial X}{\partial a_n}\right)_{x2} \\ \cdots & \cdots & \cdots & \cdots \\ \left(\frac{\partial X}{\partial a_1}\right)_{xm} & \left(\frac{\partial X}{\partial a_2}\right)_{xm} & \cdots & \left(\frac{\partial X}{\partial a_n}\right)_{xm} \end{bmatrix}$$

Nonlinear Regression

Our equation is: $J * R = \Delta y$**, with** $R = a - g$

where J **is the Jacobian matrix,** g **is the current parameter estimate,** a **is the adjustment made to the parameter estimate in the current iteration, this is the value we need to find.** Δy **is the difference between the experimental data and the model**

Solve for a**:** $J^T J \; R = J^T \Delta y$**, substitute** $J^T \Delta y = B$**,**
Option 1: use inverse: $(J^T J)^{-1} J^T J \; R = R = (J^T J)^{-1} J^T \Delta y$
Option 2: $J^T J$ **is positive definite, so use Cholesky decomposition:**
$J^T J = LL^T$
$L(L^T R) = B$**, substitute** $L^T R = Z$**, to get** $LZ = B$ **and solve for** Z **using forward substitution, then solve for** R **using backward substitution:** $L^T R = Z$**, then solve for** a **to get the adjustment for the parameter.**

Iterate until converged.

Optimization Methods

Linear Optimization:

Straight line fits

Generalized linear least squares - single iteration fitting of objective functions of the type:

$$y = a_0 + \sum_{i=1}^{n} a_i X_i , \quad -\infty < a_i < +\infty$$

NNLS (non-negative constrained least squares):

$$y = a_0 + \sum_{i=1}^{n} a_i X_i , \quad a_i \geq 0$$

Multidimensional spectrum analyses – brute force fitting approaches like grid searches (HPC recommended)

Non-parametric fits (B-splines, polynomial smoothing, etc)

Optimization Methods

Nonlinear Optimization using Gradient Descent Methods for functions of the type:

$$y = F(a_i, x_i)$$

Levenberg-Marquardt (stable, robust, works well even if initial guesses are rather far away from optimum)

Gauss-Newton methods

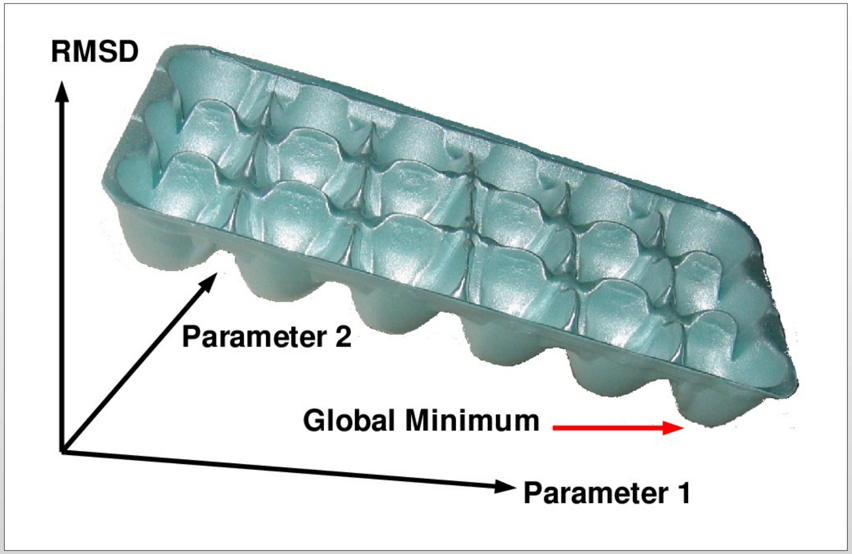
Quasi-Newton (works well near optimum)

Conjugate gradients

Tangent approximation methods (derivatives are not required)

Neural networks

The Optimization Challenge:



Problem with nonlinear least squares optimization:

For multi-component systems, the nonlinear least squares fitting algorithm gets easily stuck in local minima and the solution depends on the starting points. Problem gets worse with more parameters (i.e., multiple components).

Optimization Methods

Stochastic Methods

Monte Carlo
Simulated Annealing
Random walk
Genetic Algorithms

Optimization Methods

Comparison Stochastic vs Deterministic Fitting Methods:

Stochastic:

- **Large search space possible**
- **Generally slow converging**
- **Excellent convergence properties if given enough time**
- **Compute-intensive**
- **Suitable for many parameters**
- **Good for ill-conditioned error surfaces**
- **Derivatives not needed**

Deterministic:

- **Small search space**
- **Suitable for a few parameters only**
- **Well-conditioned error surface**
- **Very fast converging**
- **Requires derivatives**

Noise & Data Modeling Considerations

Fitting of noisy data prevents unique solutions – multiple solutions are possible.

We need to *minimize noise* when modeling data.

There are three ways to reduce or eliminate noise:

1. **fit the noise**
2. **maintain an exceptionally well tuned instrument**
3. **design your experiment to optimize the quality of the data**

There are two noise types:

1. Systematic noise: Signal comes from a systematic source that is not part of the parametric model (finger print on the lens of a camera) and is highly correlated with some feature of the experiment.

2. Stochastic (random noise): Noise is (hopefully) Gaussian in distribution and uncorrelated to any feature of the experiment

(1) can often be fitted and accounted for, which (2) must be minimized.

Noise & Data Modeling Considerations

A *first principles* approach allows us to model the experimental data by fitting it to a mathematical model. The model represents the physics of the experiment, and contains parameters of interest to the experimentalist. We need to find the values of these parameters by adjusting the model so it matches the data. This is a hard problem called an "*inverse problem*" that requires optimization (fitting) algorithms which aid us in adjusting the parameters so the model fits the data.

In UltraScan this is accomplished by a least squares fitting approach that compares each data point from the model with the corresponding point in the experimental data:

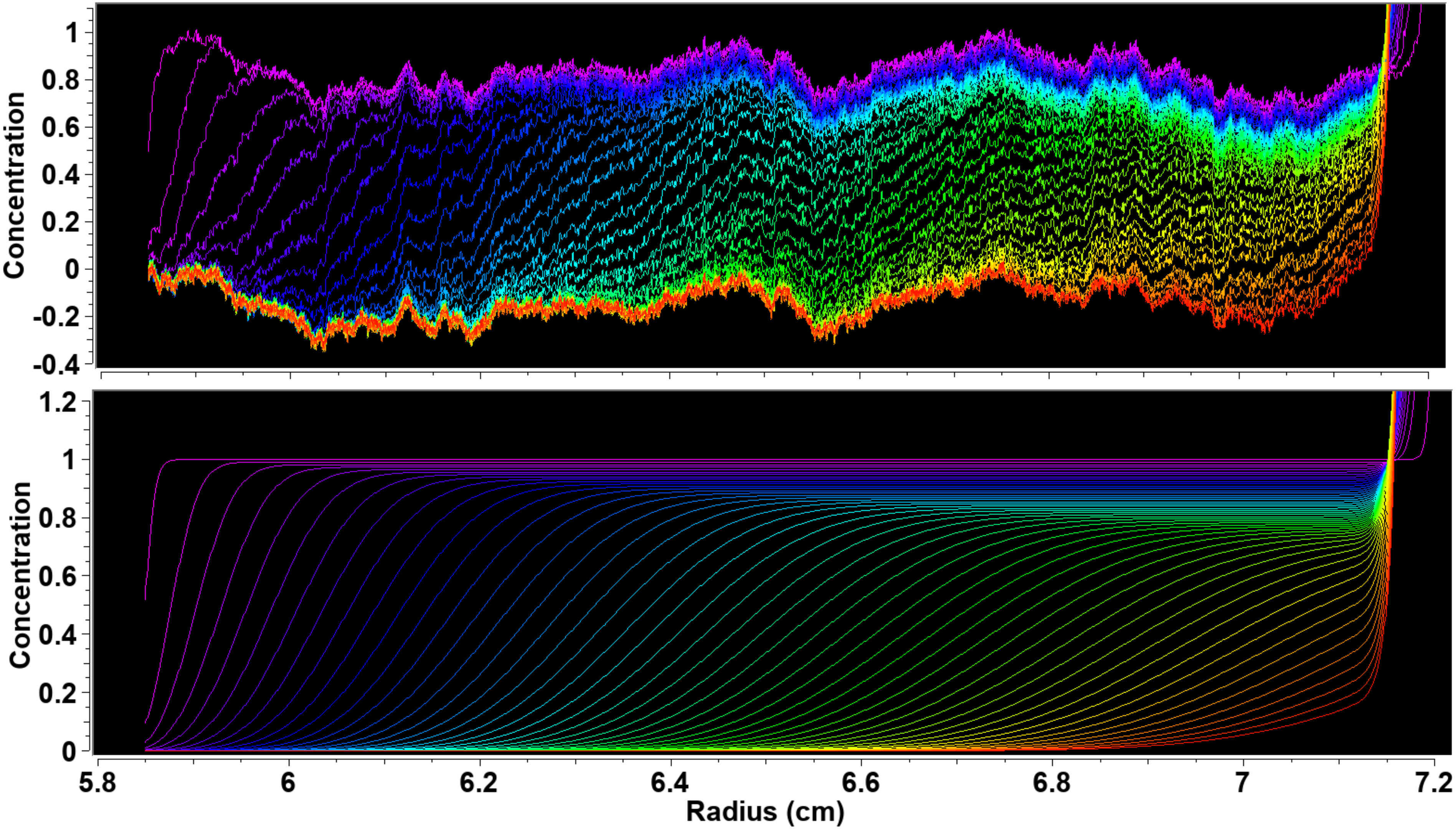
$$Minimize \sum_{i=1}^{N} \left(Data_i \; - \; Model_i \right)^2 \quad (i \text{ over radius and time})$$

Optimally, the difference is zero, but because of experimental noise this never happens, since the model is noise free.

Noise & Data Modeling Considerations

Fitting of noisy data prevents unique solutions – multiple solutions are possible.

We need to *minimize noise* when modeling data.



There are three ways to reduce or eliminate noise:

1. fit the noise
2. maintain an exceptionally well tuned instrument
3. design your experiment to optimize the quality of the data

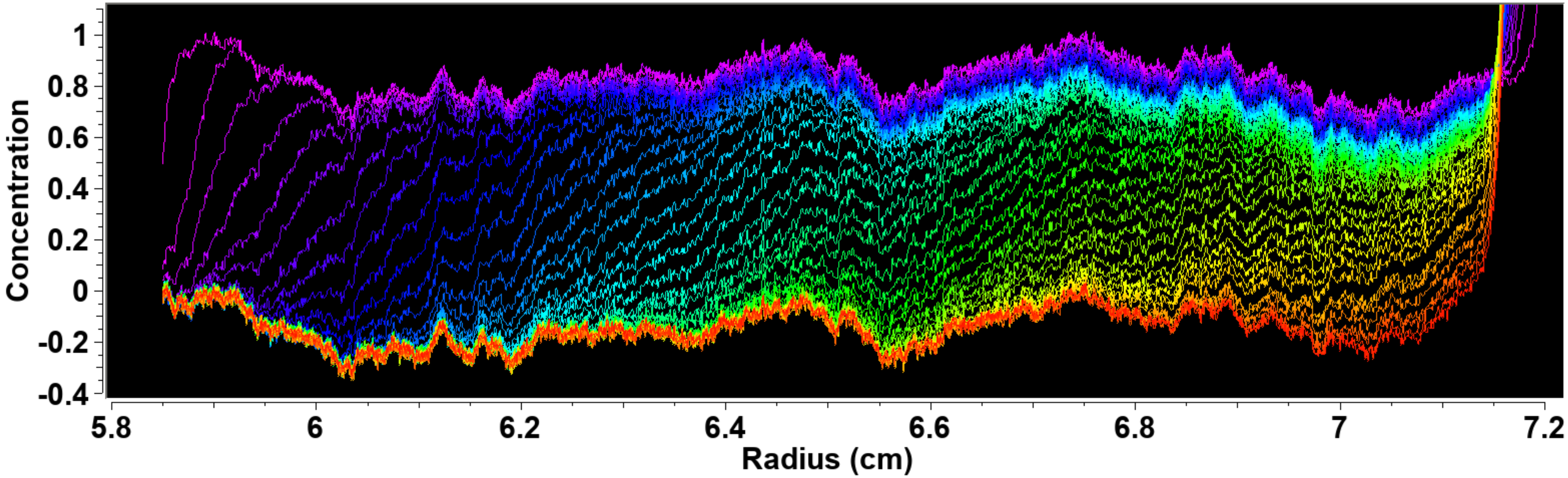
There are three noise types:

1. Time Invariant noise: Noise is different for each radial position, but the same offset for each scan, and hence time independent (finger prints).

2. Radially Invariant noise: Noise is different for each scan, but each radial position is offset by the same amount throughout the scan (baseline variation)

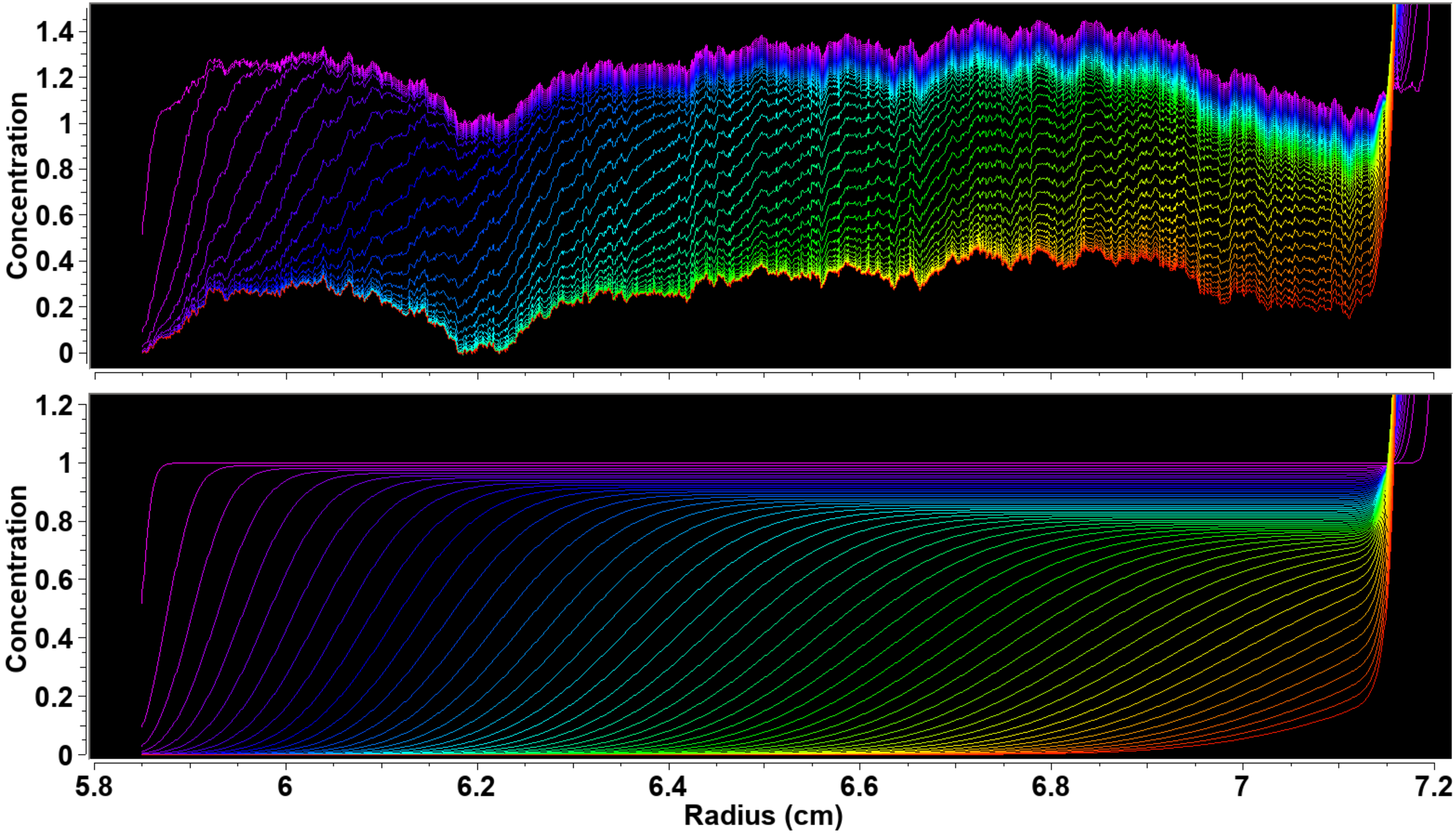
3. Stochastic (random noise): Noise is different for each radial and time point and it is (hopefully) Gaussian in distribution

(1) and (2) can be fitted by UltraScan and removed from the data



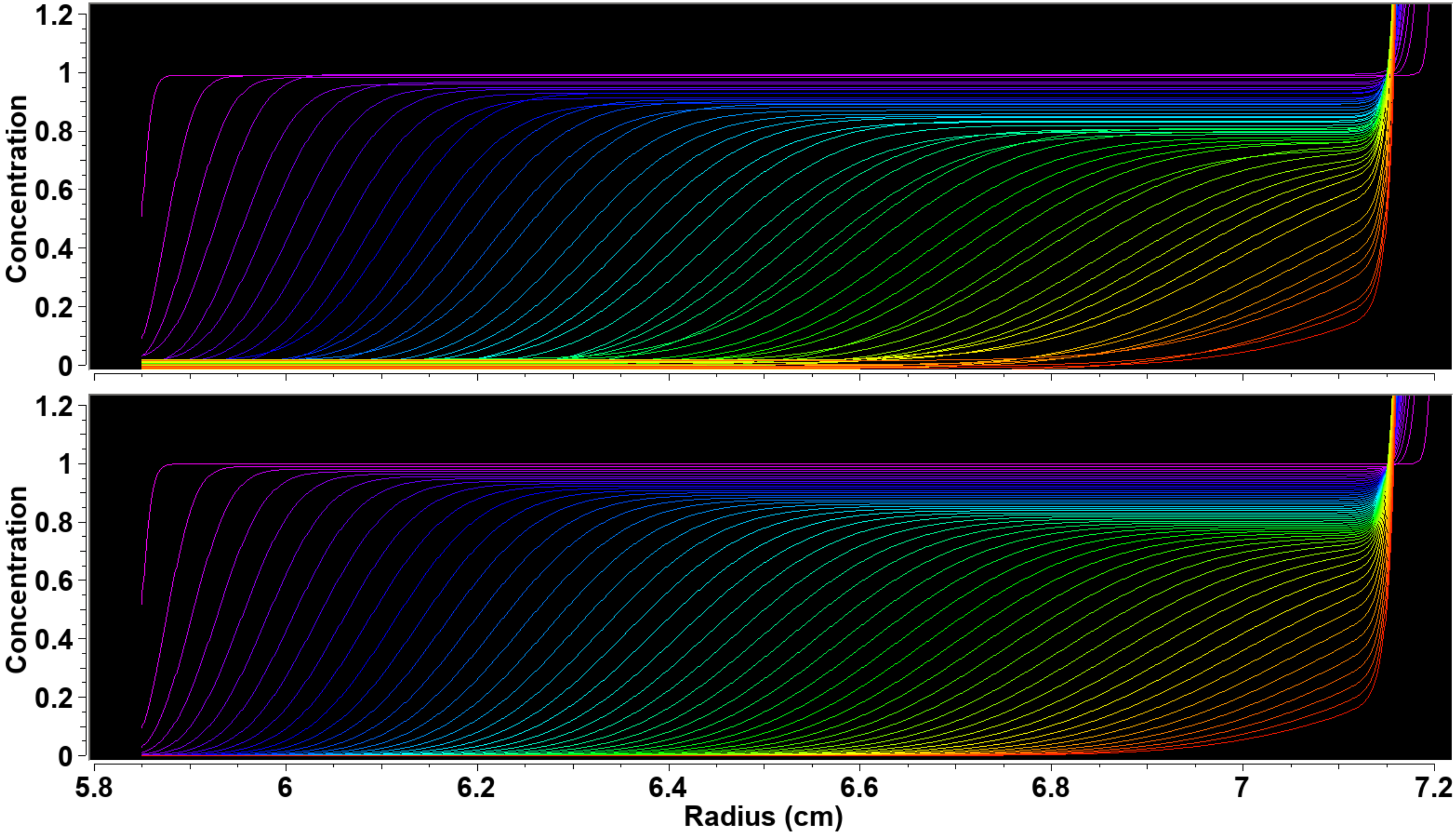
Noise & Data Modeling Considerations

Time Invariant noise: Noise is different for each radial position, but the same offset for each scan, and hence time independent.



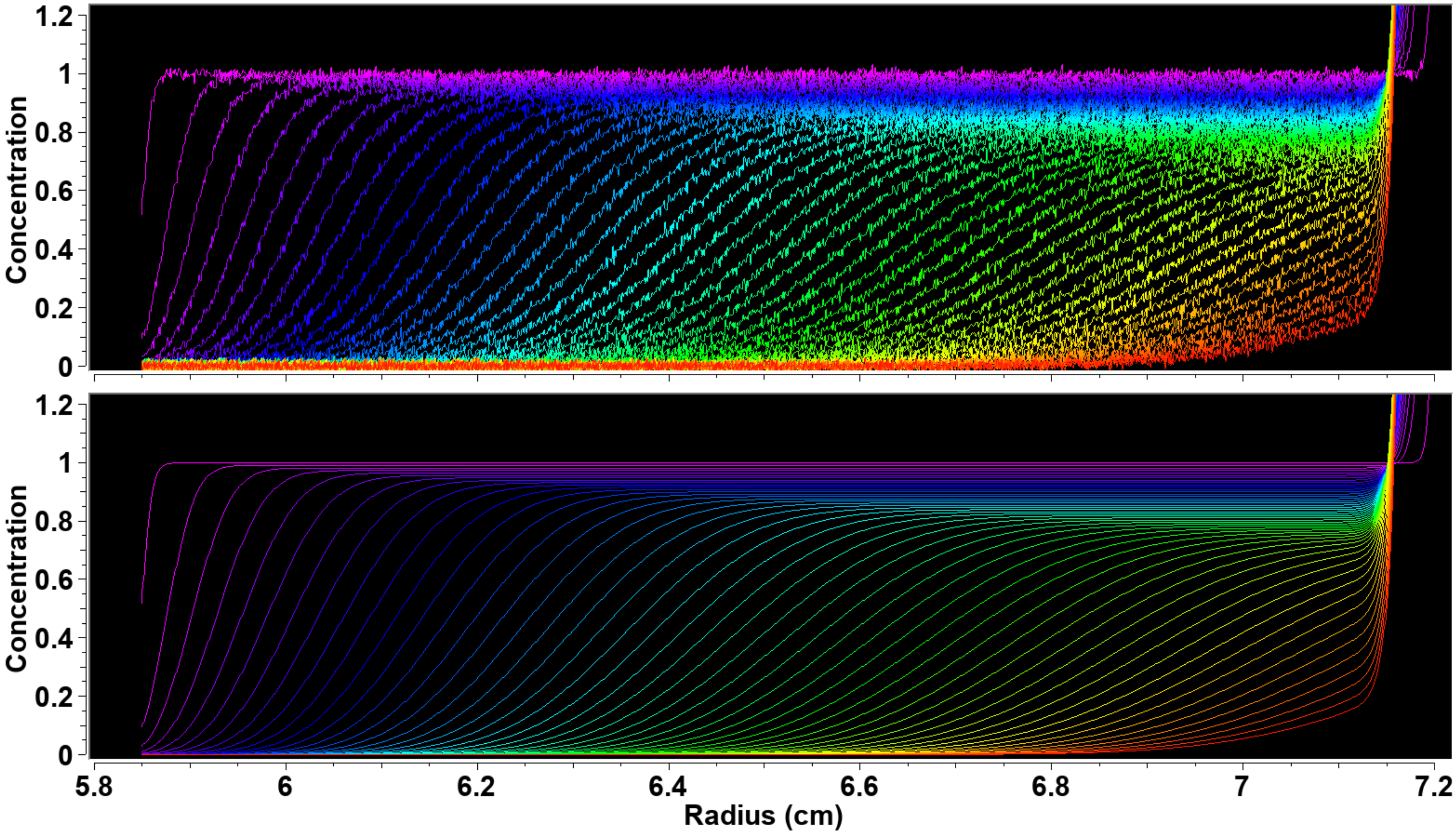
Noise & Data Modeling Considerations

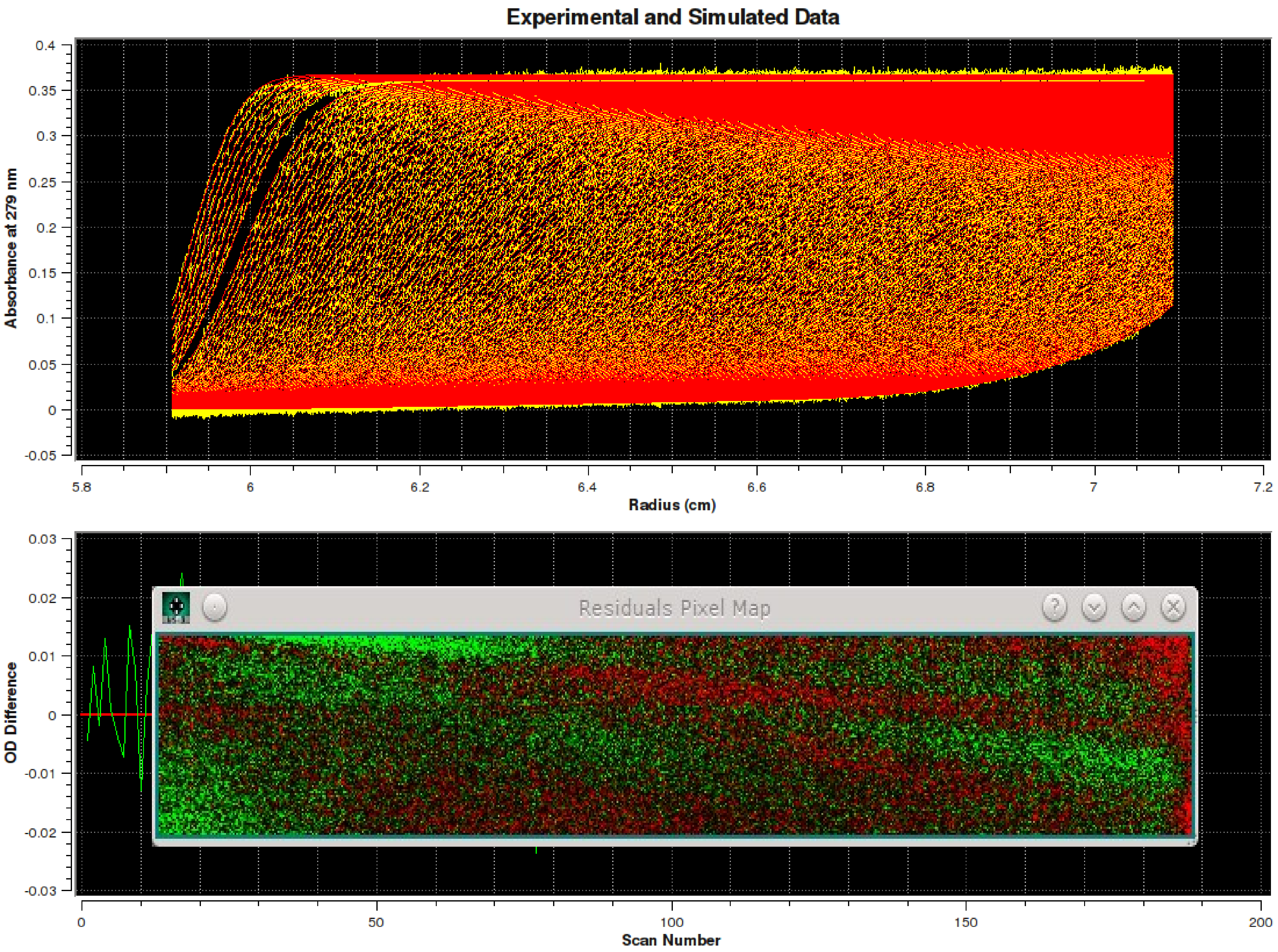
Radially Invariant noise: Noise is different for each scan, but each radial position is offset by the same amount throughout the scan



Noise & Data Modeling Considerations

Stochastic (random noise): Noise is different for each radial and time point and it is (hopefully) Gaussian in distribution:




Experimental and Simulated Data
Absorbance at 279 nm
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
5.8
6
6.2
6.4
6.6
6.8
7
7.2
Radius (cm)
OD Difference
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
Residuals Pixel Map
0
50
100
150
200
Scan Number

Homework 2

Homework (25 pts): For the dataset shown on the right calculate the equations for a and b for a straight line fit using Cramer's rule (y = c_1 + c_2x). Assume a standard deviation of 1 for each measurement. Show your work. Compare your answer by fitting with a plotting program. Show results. Extra credit (10 pts): write a Python or C++ program that solves this problem.

x	y	σ
2	10.6	1
4	12.1	1
6	14.5	1
8	20.8	1
10	17.3	1
12	24.7	1
14	29.1	1

$$A = \begin{bmatrix} \sum_{i=1}^{n} \frac{1}{\sigma_i^2} & \sum_{i=1}^{n} \frac{x_i}{\sigma_i^2} \\ \sum_{i=1}^{n} \frac{x_i}{\sigma_i^2} & \sum_{i=1}^{n} \frac{x_i^2}{\sigma_i^2} \end{bmatrix}, \quad X = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}, \quad \text{and } B = \begin{bmatrix} \sum_{i=1}^{n} \frac{D_i}{\sigma_i^2} \\ \sum_{i=1}^{n} \frac{D_i x_i}{\sigma_i^2} \end{bmatrix}$$

$$c_1 = \frac{b_1 a_{22} - b_2 a_{12}}{a_{11} a_{22} - a_{12} a_{21}}, \quad c_2 = \frac{b_2 a_{11} - b_1 a_{21}}{a_{11} a_{22} - a_{12} a_{21}}$$