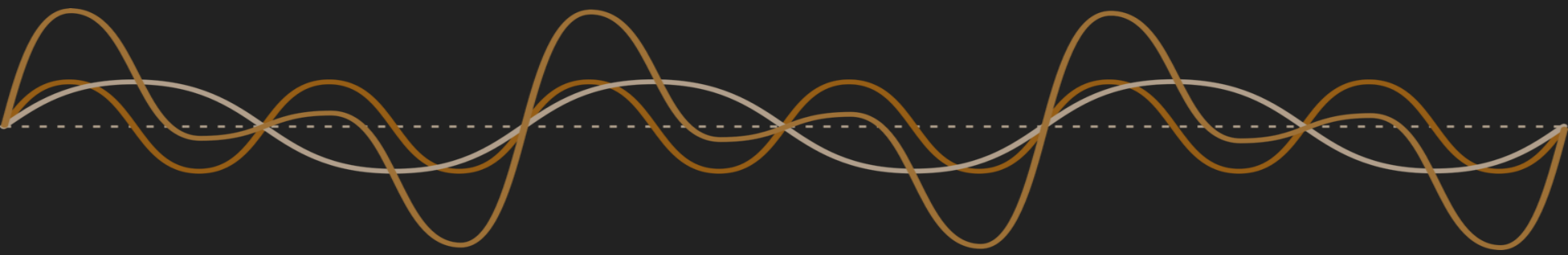
So, how do we do these curve fits, anyway?

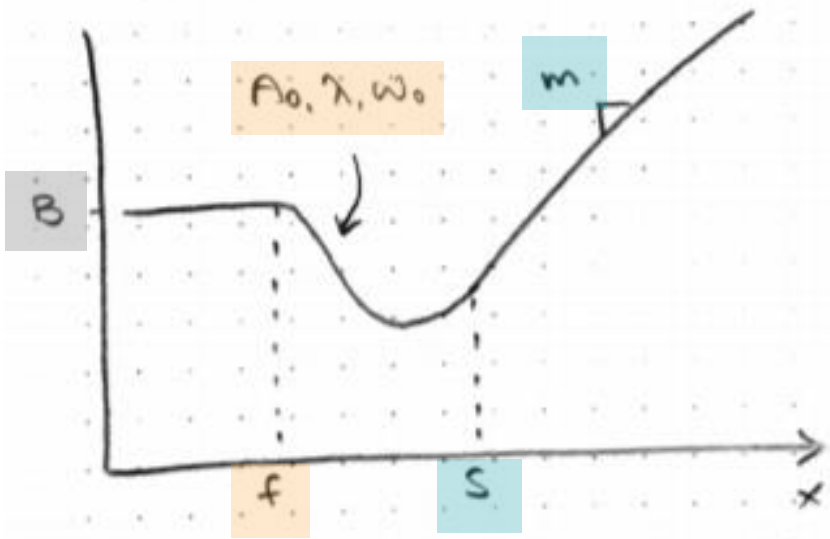


Rachel Rise
5 June
Stochastic Resonance Project

University of Colorado **Boulder**
Bioastronautics

The basic equation

$$y = B - \frac{A_0}{(x-f)^2}\left[\frac{r}{\sqrt{4r^2+w_0^2}}\right]\begin{cases}0, & x<f\\ 1, & x>f\end{cases} + m\begin{cases}0, & x<s\\ x-s, & x\geq s\end{cases}$$



B	–	BASELINE / SHAM THRESHOLD
A_0	–	DEPTH OF DIP (LINEAR)
λ	–	DEPTH OF DIP (NONLINEAR)
w_0	–	WIDTH OF DIP (NONLINEAR)
f	–	START OF DIP
m	–	SLOPE OF MASKING
s	–	START OF MASKING

Nonlinear constraints in fmincon

fmincon arguments (things we use are bold):

- **@(X) parameters to optimize**
- **optimization_function(X, other_arguments)**
- **Guesses**

Same as fminsearch

- Linear inequalities: matrix [A] and vector b such that [A]*X <= b
- Linear equalities: matrix [A_{eq}] and vector b_{eq} such that [A_{eq}]*X = b_{eq}
- **Lower and upper bounds on X:** vectors LB and UB
- **@nonlinear_constraint_function**
- Options (**suppress output**, tolerance, algorithm, etc.)

The nonlinear constraint function

University of Colorado Boulder Bioastronautics

[c, ceq] = non_linear_constraint_fcn(X)

X must be only argument, other variables must be global

The constraints are as follows:

1) c(X) <= 0
   for example: if you want $2X^2 - 3X <= 15$ and X >= 4y

```
global y
c(1) = 2*X^2 - 3*X - 15;
c(2) = 4*y - X;
```

2) Similarly, ceq(X) = 0
   example: if you want $\ln(X) + 5X^3 = y$;

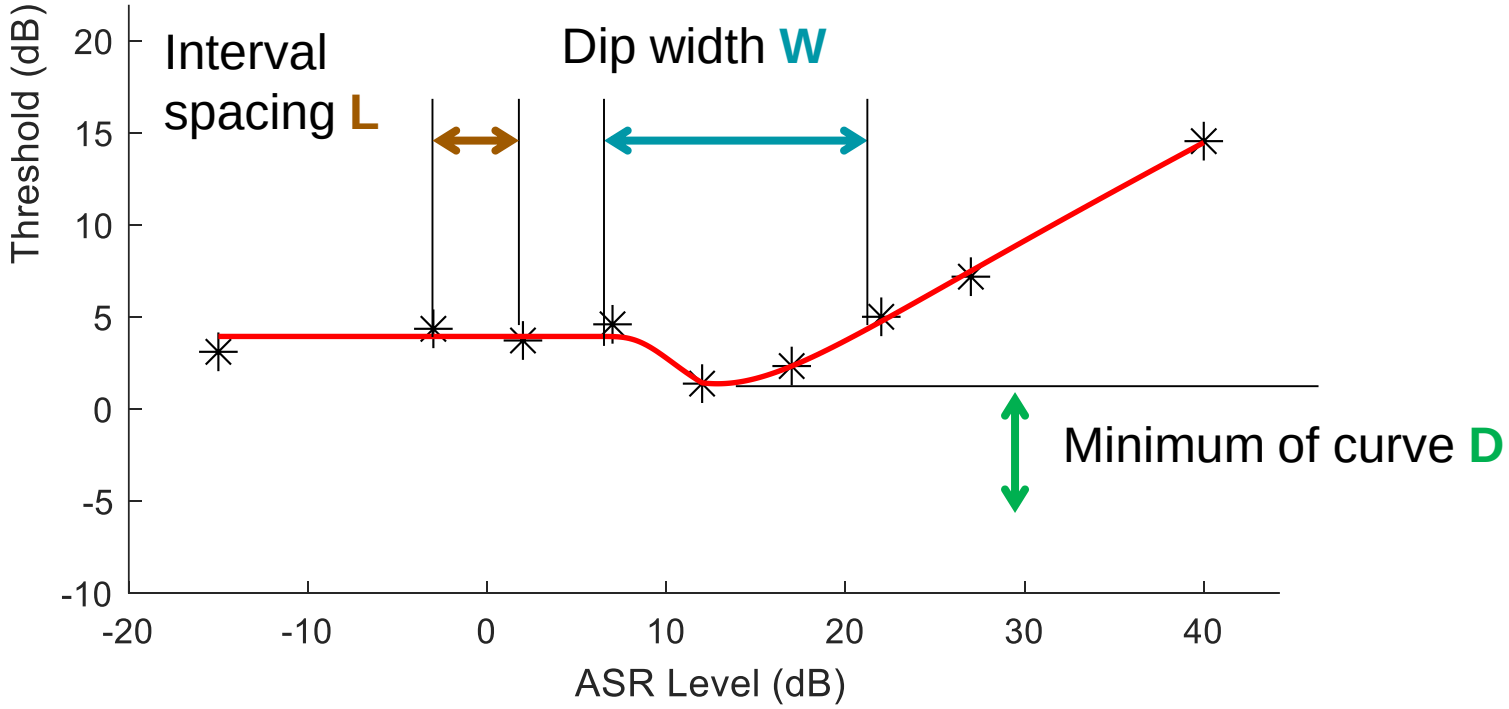
```
global y
ceq = log(X) + 5*X^3 - y;
```

Constraints in this problem

1) 1.5**L** ≤ **W** ≤ 5**L**

2) 0.9*(lowest measured threshold) ≤ **D**

3) Dip comes back up before highest SR noise level



Our implementation of fmincon

University of Colorado Boulder
Bioastronautics

```
global levels thresholds TestType SRType method
widthThreshold1 widthThreshold2

%% Enter loops get the data from each subject file
% Populate levels and thresholds
% Assign method, TestType and SRType

[params, J, flag] = fmincon(@(X)

        SR_curve_fit_freeB(X, levels, thresholds), guesses,

        [],[],[],[],                % Linear constraints

        LowerBounds,UpperBounds,    % Lower and upper bounds

        @SR_minWidth_maxDepth,      % Nonlinear constraints

        options);
```

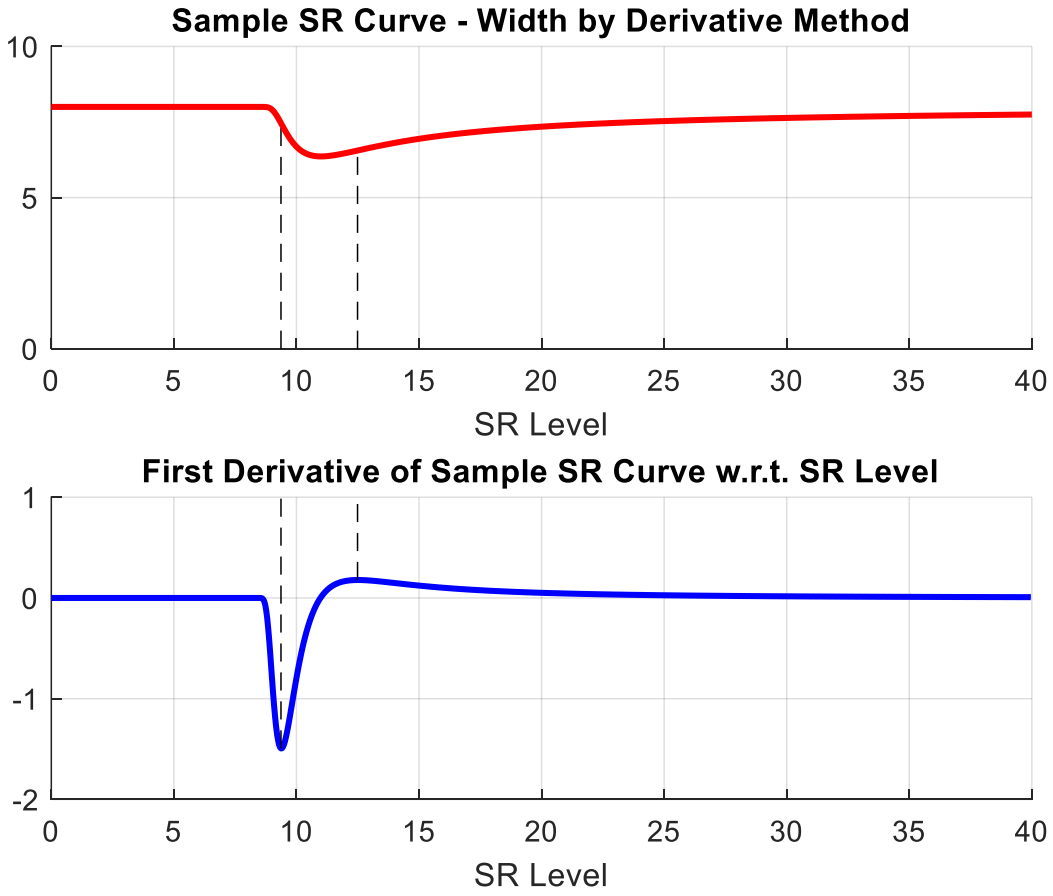
Nonlinear constraint function

University of Colorado Boulder
Bioastronautics

```
function [c, ceq] = SR_minWidth_maxDepth(X)

global levels thresholds TestType SRType method widthThreshold1
widthThreshold2

%% Evaluate what the curve is
SR_curve = ...;

%% Determine what the width of the curve is
x1 = ...;
x2 = ...;
width = x2 - x1;

%% The constraint equations
% Curve fit minimum >= 90% of lowest measured threshold
c(1) = 0.9*min(thresholds) - min(SR_curve)

% Width >= 1.5*levelSpacing
c(2) = 1.5*levelSpacing - width;          % levelSpacing based on SRType

% Width <= 5*levelSpacing
c(3) = width - 5*levelSpacing;
```

Method 1: Derivative

Minimum and maximum slope define width of dip



Method 2: Percentage (DO NOT USE)

Crossing a certain percentage of dip depth determines dip width

75% of dip depth

90% of dip depth

Method 3: Integral (DO NOT USE)

Quotient of dip area and depth determines width

Changes this time around

- New constraint on x2: must be less than max SR noise level
- Curve fit propagated out to 3x max SR noise level to better define x2