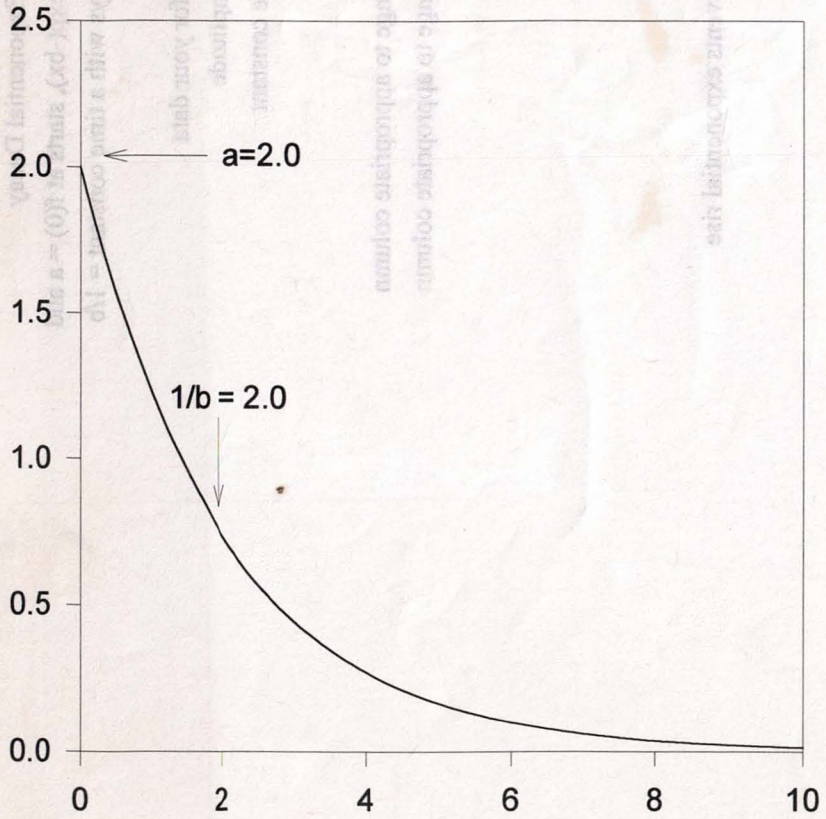
FIT

expdecay1.spw

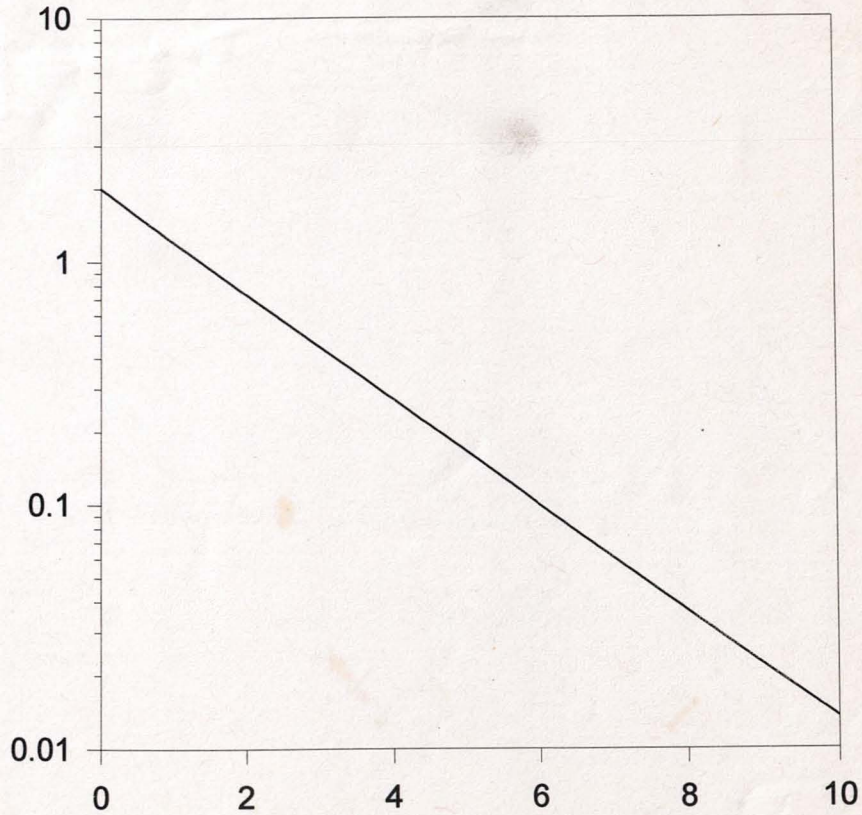
Single Exponential Decay

a = 2.0, b = 0.5

Linear Axis Scale



Logarithmic Y Axis Scale



SPW\MATH\FIT

```
[Parameters] ;Single Exponential Decay
                ;f=a exp(-bx), starts at f(0) = a and
                ;decays with a time constant = 1/b

;Modify these values for your data
a=1             ;amplitude
b=.1            ;rate constant

[Variables]
x=col(1)        ;change to appropriate column
y=col(2)        ;change to appropriate column

[Equations]
f = a*exp(-b*x)
fit f to y

[Constraints]
b > 0           ;prevents exponential rise
```

Expdcay2.spw

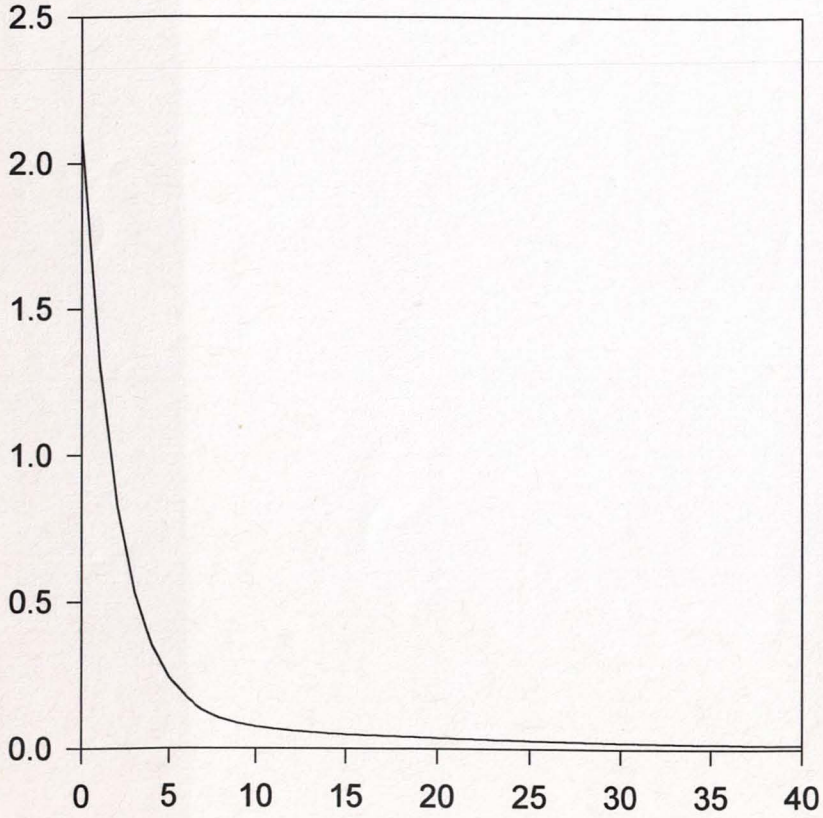
$y = a + b \cdot \exp(cx)$

Double Exponential Decay

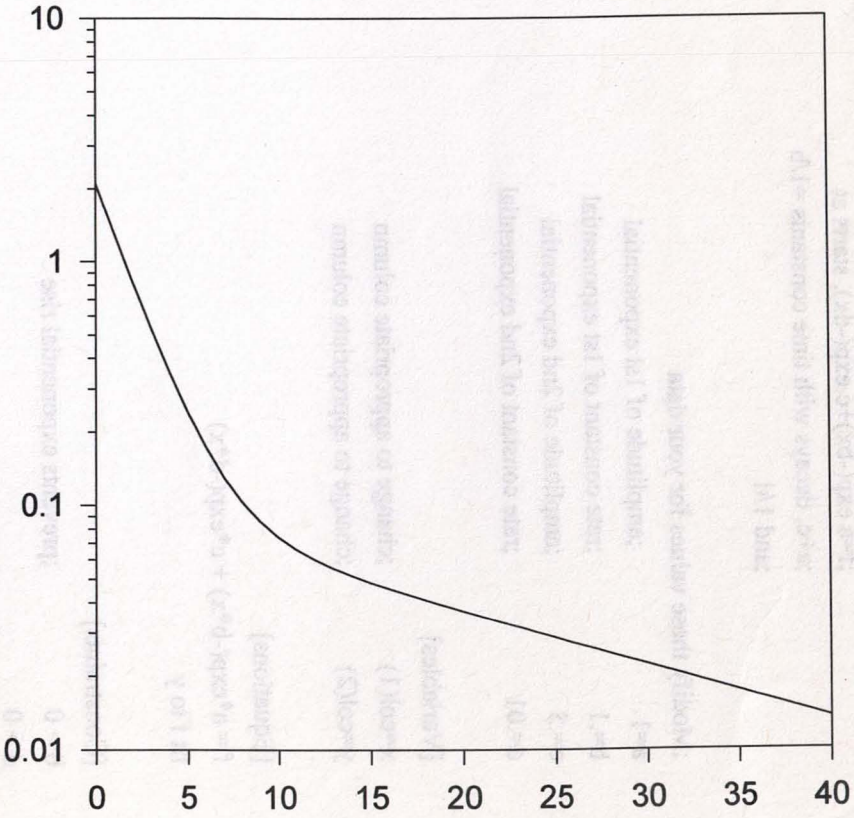
a = 2.0, b = 0.5

c = 0.1, d = 0.05

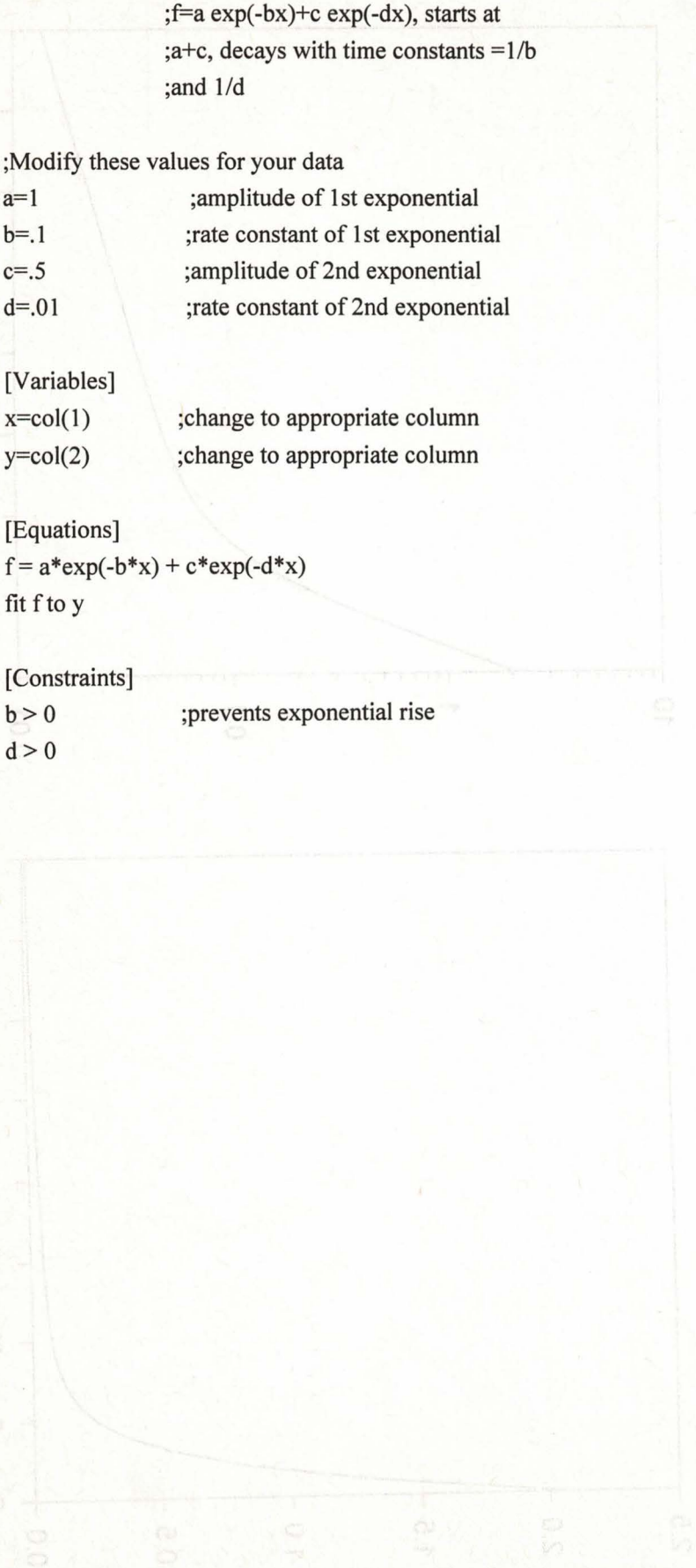
Linear Y Axis Scale



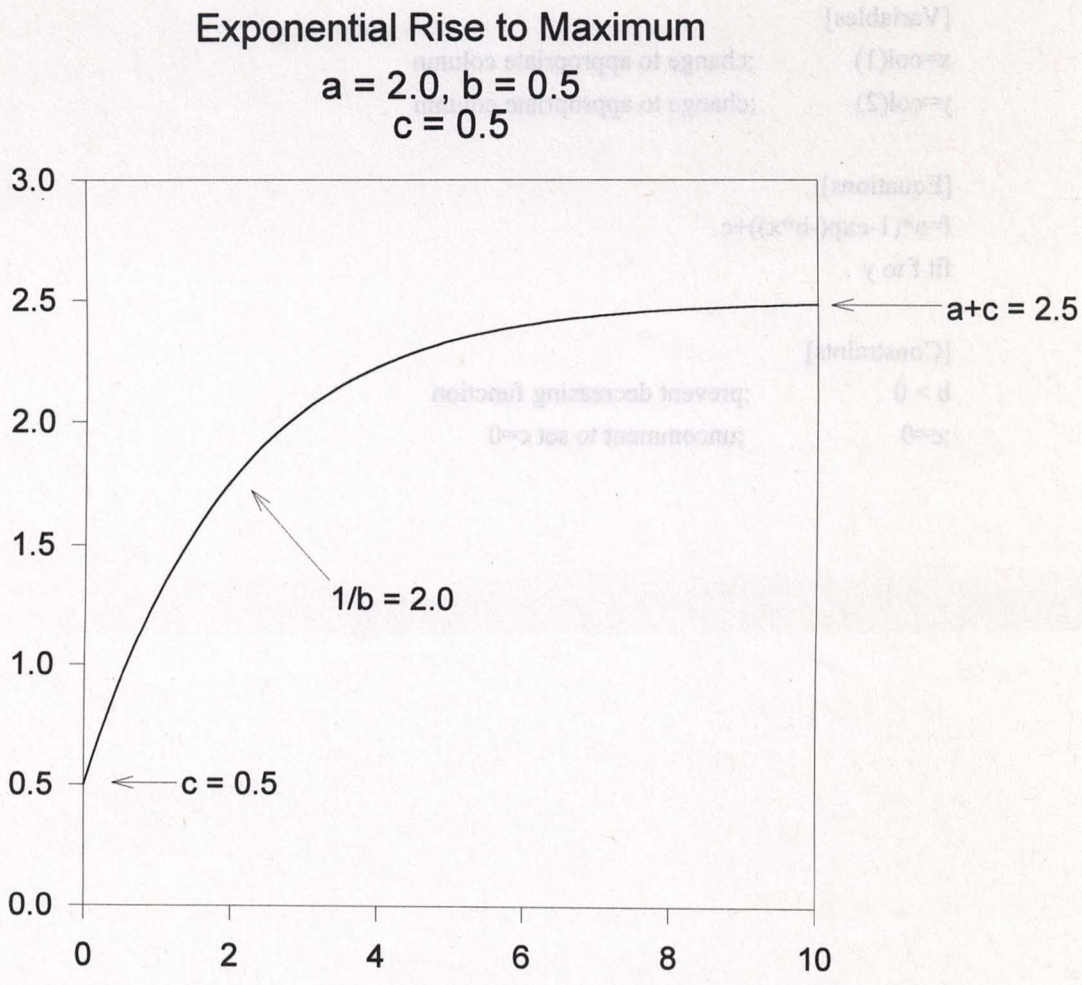
Logarithmic Axis Scale



```
[Parameters] ;Double Exponential Decay
             ;f=a exp(-bx)+c exp(-dx), starts at
             ;a+c, decays with time constants =1/b
             ;and 1/d

;Modify these values for your data
a=1          ;amplitude of 1st exponential
b=.1         ;rate constant of 1st exponential
c=.5         ;amplitude of 2nd exponential
d=.01        ;rate constant of 2nd exponential

[Variables]
x=col(1)     ;change to appropriate column
y=col(2)     ;change to appropriate column

[Equations]
f = a*exp(-b*x) + c*exp(-d*x)
fit f to y

[Constraints]
b > 0        ;prevents exponential rise
d > 0
```

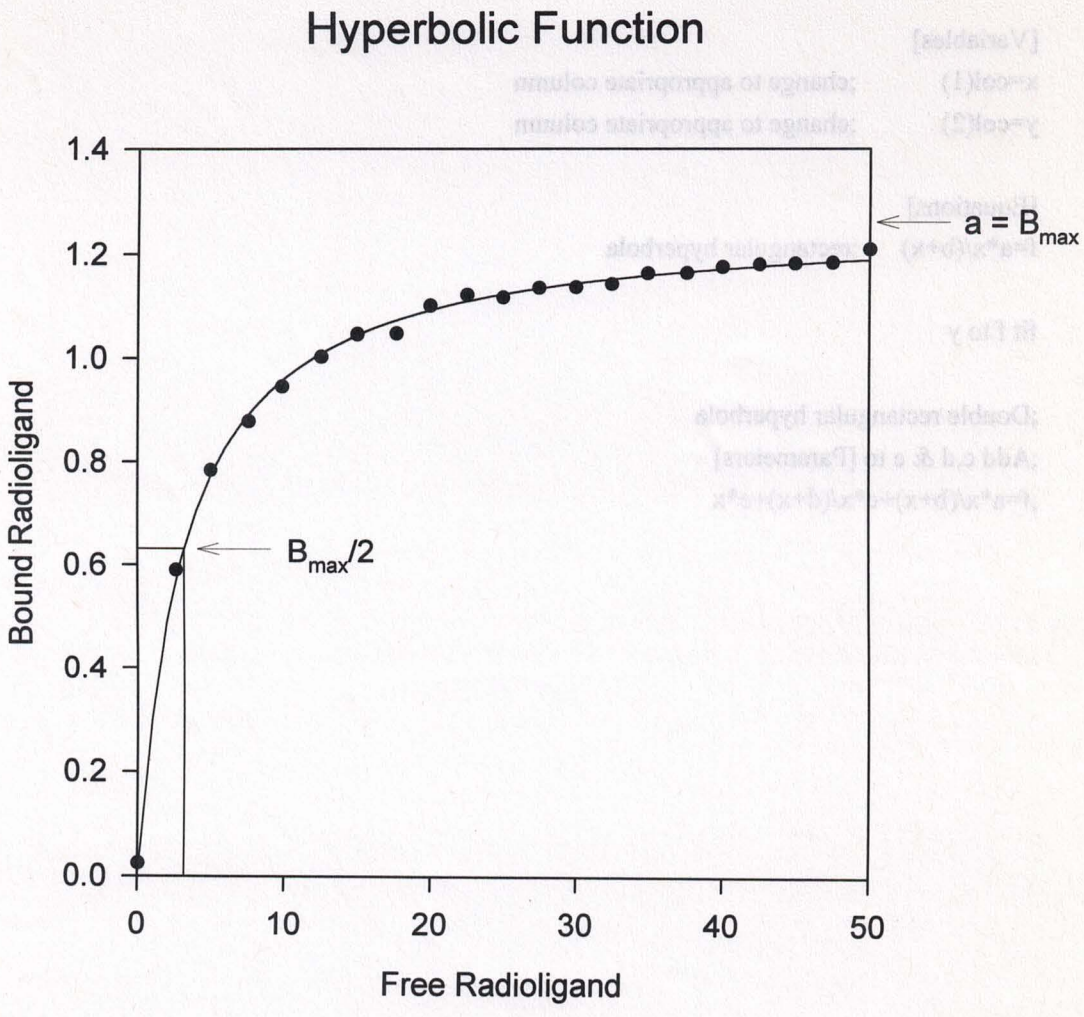


Exprise - spw


Exponential Rise to Maximum
a = 2.0, b = 0.5
c = 0.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
2
4
6
8
10
a+c = 2.5
1/b = 2.0
c = 0.5

```
[Parameters] ;Exponential Rise to Maximum
             ;f=a[1-exp(-bx)]+c, starts at c,
             ;rises to a+c with time constant=1/b

;Modify these values for your data
a=1          ;amplitude of exponential
b=.1         ;rate constant
c=1          ;zero intercept

[Variables]
x=col(1)     ;change to appropriate column
y=col(2)     ;change to appropriate column

[Equations]
f=a*(1-exp(-b*x))+c
fit f to y

[Constraints]
b > 0        ;prevent decreasing function
;c=0         ;uncomment to set c=0
```

Hyperbol.spw



Parameters] ;Hyperbolic Functions
;(Saturating Binding Curves)
;f=a*x/(b+x), starts at 0, rises to
;a (Bmax) and for x=b (KD) is a/2.

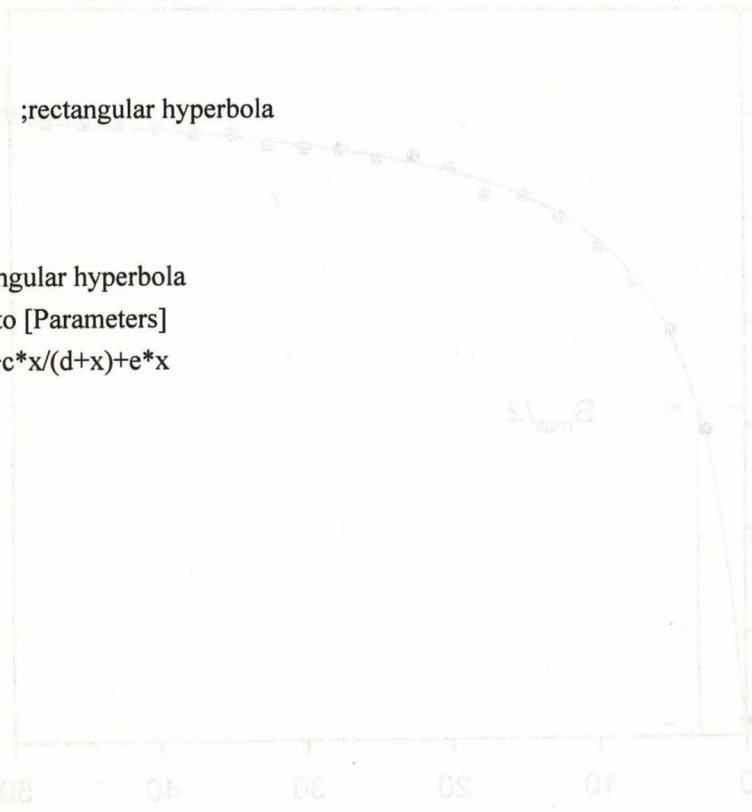
;Modify these values for your data
a=1 ;maximum value (Bmax)
b=.5 ;dissociation constant (KD)

[Variables]
x=col(1) ;change to appropriate column
y=col(2) ;change to appropriate column

[Equations]
f=a*x/(b+x) ;rectangular hyperbola

fit f to y

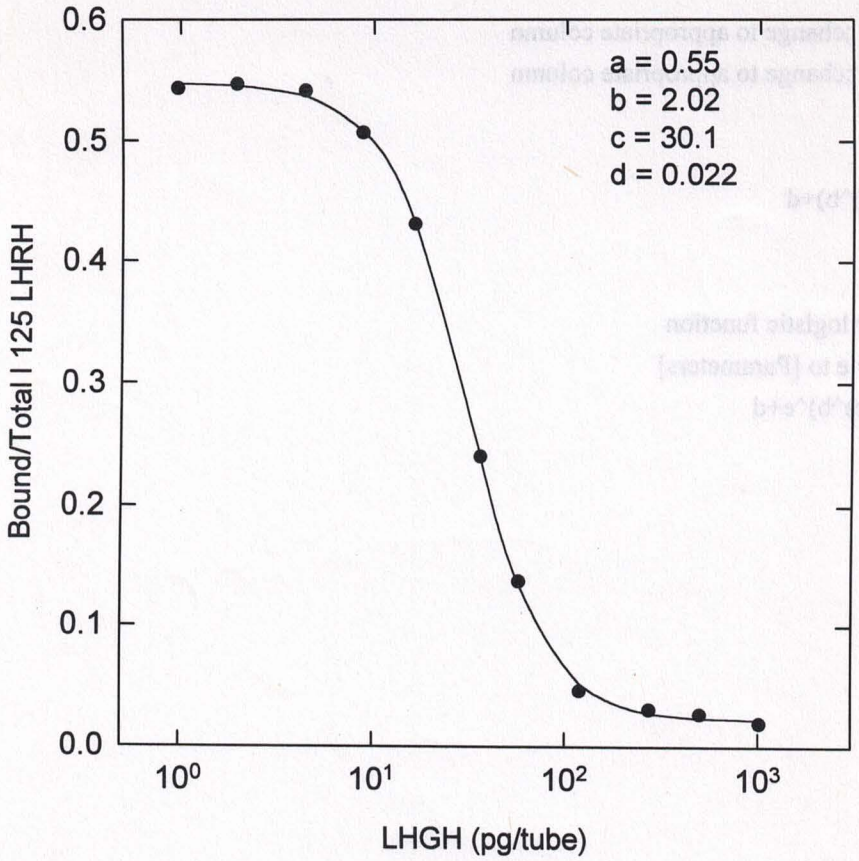
;Double rectangular hyperbola
;Add c,d & e to [Parameters]
;f=a*x/(b+x)+c*x/(d+x)+e*x



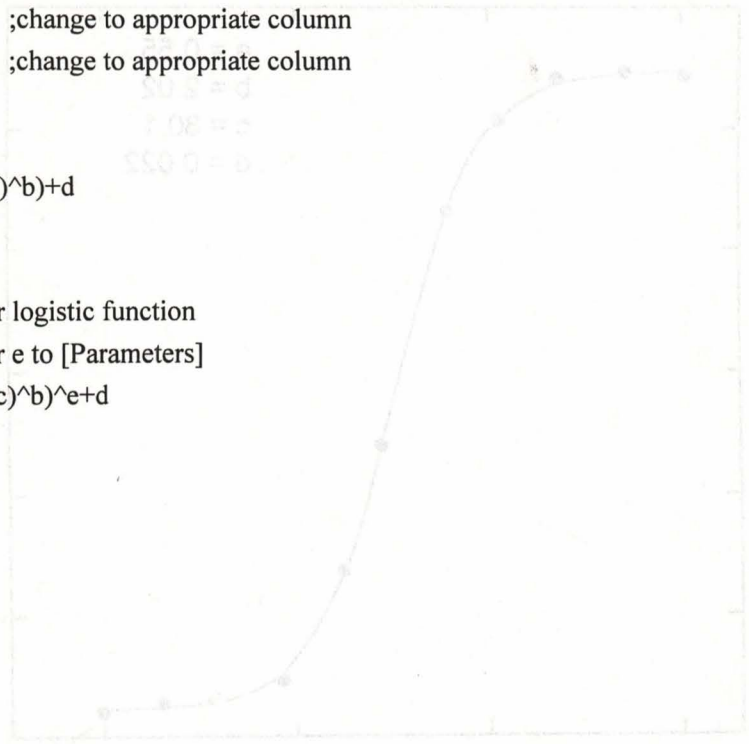
fit

logistic-spw

Logistic Function

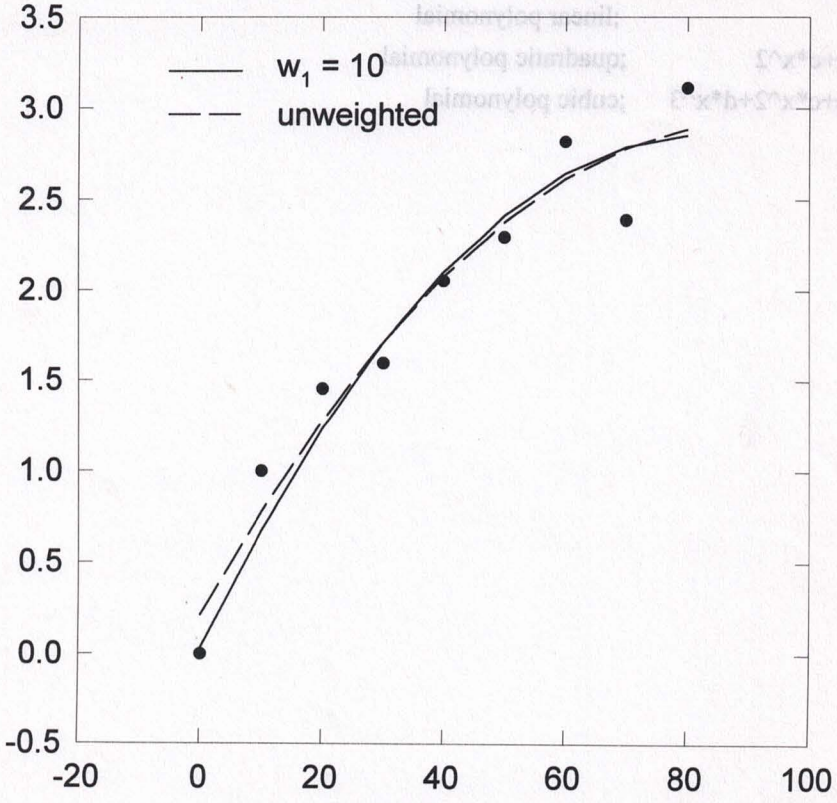


```
[Parameters] ;4 Parameter Logistic Function
             ;f=(a-d)/[1+(x/c)^b]+d, if b>0 it
             ;starts at a and falls to d.   If b<0
             ;it starts at d and rises to a.

;Modify these values for your data
a=.5         ;asymptotic maximum
b=1.5        ;slope parameter, b>0 gives slope<0
c=45         ;value at inflexion point
d=.01        ;asymptotic minimum

[Variables]
x=col(1)     ;change to appropriate column
y=col(2)     ;change to appropriate column

[Equations]
f=(a-d)/(1+(x/c)^b)+d

fit f to y
;Five parameter logistic function
;Add parameter e to [Parameters]
;f=(a-d)/(1+(x/c)^b)^e+d
```



poly.spw

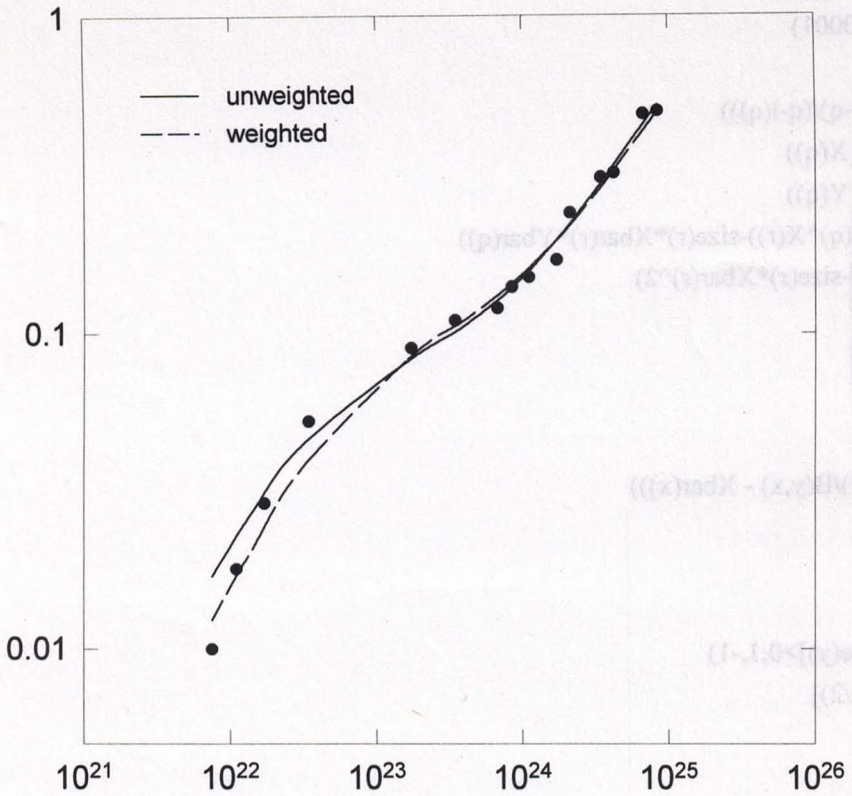
Quadratic Polynomial Fit
Forced Through Zero Using Weighting



```
[Parameters] ;Polynomial Curve Fits
                ;These fits may also be obtained
                ;using the Regression feature

a=1             ;zero intercept
b=2             ;slope
;c=1            ;used for alternate models below
;d=2            ;ditto

[Variables]
x=col(1)        ;change to appropriate column
y=col(2)        ;change to appropriate column

[Equations]
f = a+b*x                       ;linear polynomial
;f = a+b*x+c*x^2                ;quadratic polynomial
;f = a+b*x+c*x^2+d*x^3          ;cubic polynomial

fit f to y
```

SPW\MATH\

Advanced - SPW



A = 0.1
CD = -1
db = .1
x = col(1)*1e-24
y = col(2)
w = 1/col(2)^2

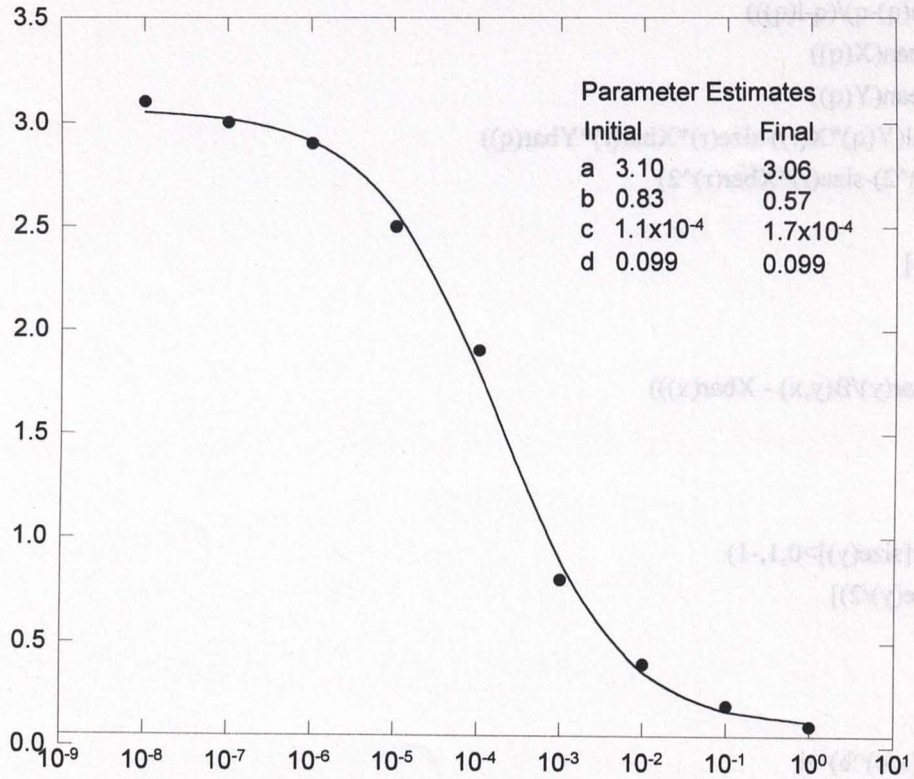
} Variables

f = 1-exp(-x*(db + CD*x)/(x+A))
fit f to y with weight w

```
[Variables]
        ;Automatic Computation of Initial
        ;Parameter Estimates for 4 Parameter
        ;Logistic Function

;Variables
x=col(1)
y=col(2)

;Initial Parameter Estimate Functions
u(q)= (max(q)+.0001)
l(q)= (min(q)-.0001)
X(q)=log(q)
Y(q)=log((u(q)-q)/(q-l(q)))
Xbar(q)=mean(X(q))
Ybar(q)=mean(Y(q))
B(q,r)=(total(Y(q)*X(r))-size(r)*Xbar(r)*Ybar(q))
 /(total(X(r)^2)-size(r)*Xbar(r)^2)

[Parameters]
a=u(y)
b=B(y,x)
c=10^(-(Ybar(y)/B(y,x) - Xbar(x)))
d=l(y)

;a=max(y)
;b=if(y[1]-y[size(y)]>0,1,-1)
;c=x[int(size(y)/2)]
;d=min(y)

[Equations]
f=(a-d)/(1+(x/c)^b)+d
fit f to y
```

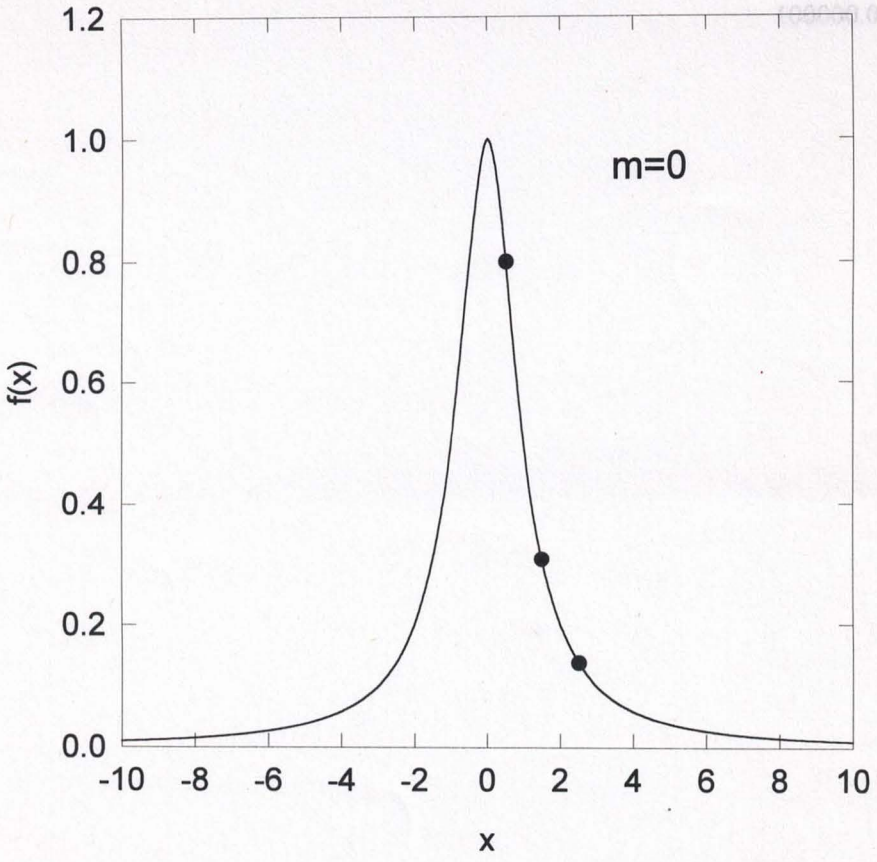
automat-spw

Logistic Function Fit Using Automatic
Initial Parameter Estimates

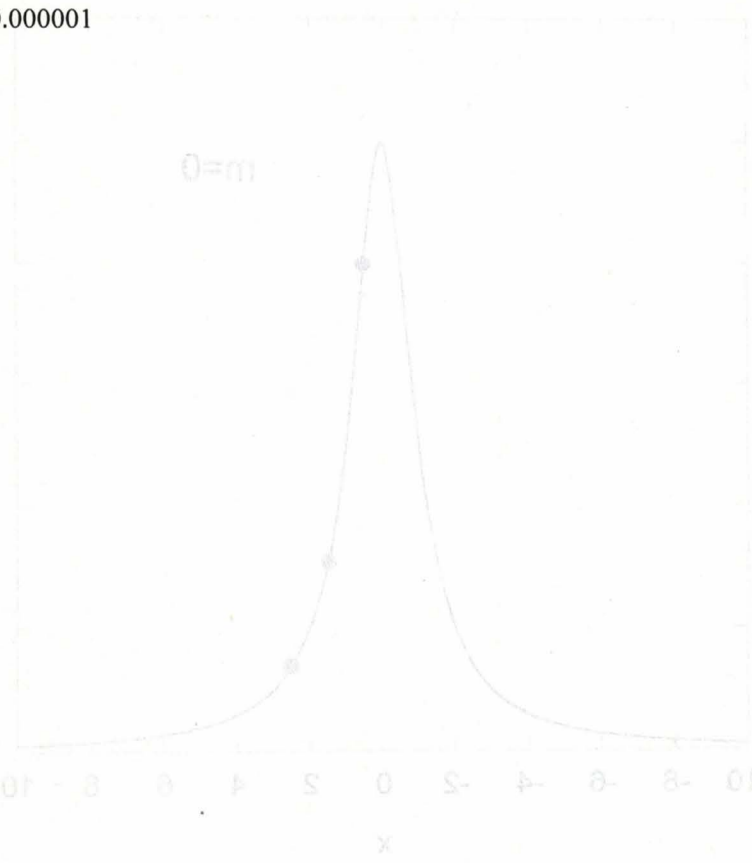


```
[Variables]
        ;Automatic Computation of Initial
        ;Parameter Estimates for 4 Parameter
        ;Logistic Function

;Variables
x=col(1)
y=col(2)

;Initial Parameter Estimate Functions
u(q)= (max(q)+.0001)
l(q)= (min(q)-.0001)
X(q)=log(q)
Y(q)=log((u(q)-q)/(q-l(q)))
Xbar(q)=mean(X(q))
Ybar(q)=mean(Y(q))
B(q,r)=(total(Y(q)*X(r))-size(r)*Xbar(r)*Ybar(q))
 /(total(X(r)^2)-size(r)*Xbar(r)^2)

[Parameters]
a=u(y)
b=B(y,x)
c=10^(-(Ybar(y)/B(y,x) - Xbar(x)))
d=l(y)

;a=max(y)
;b=if(y[1]-y[size(y)]>0,1,-1)
;c=x[int(size(y)/2)]
;d=min(y)

[Equations]
f=(a-d)/(1+(x/c)^b)+d
fit f to y
```

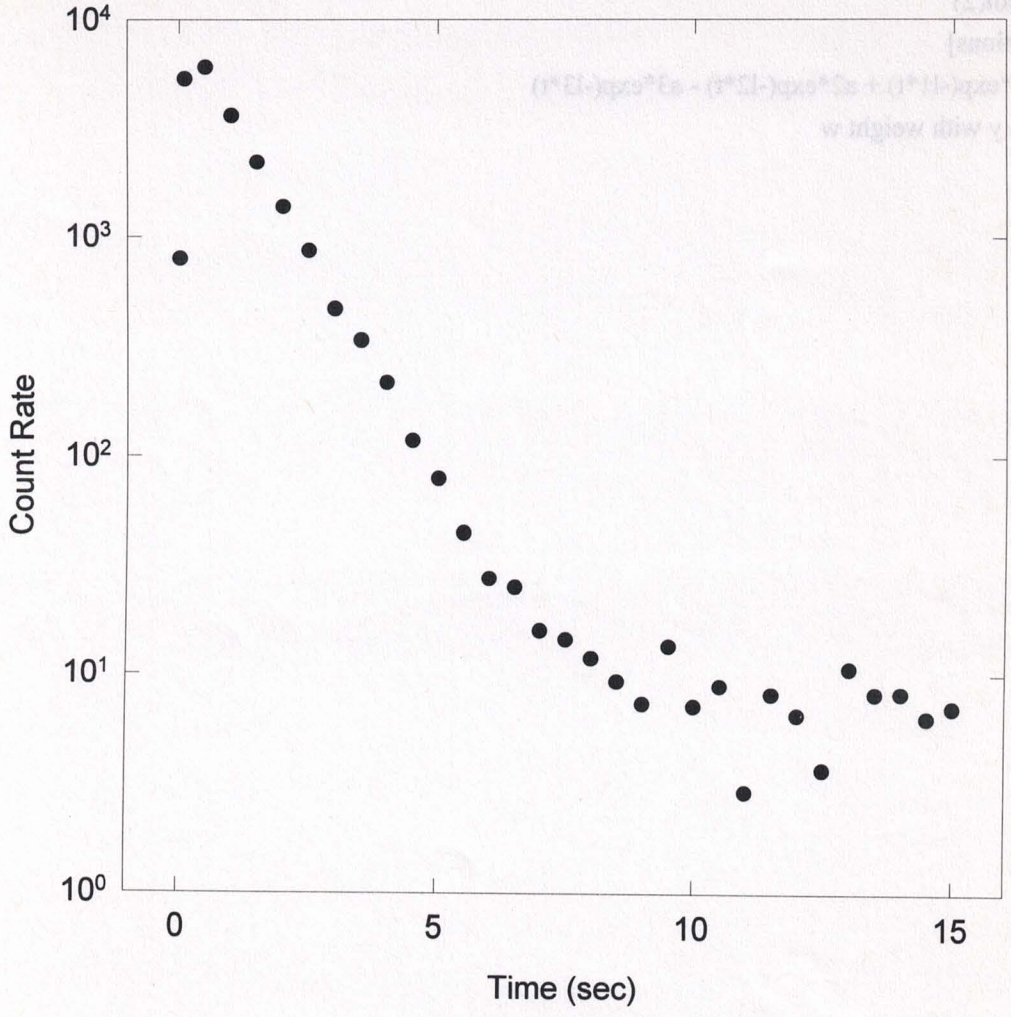
fit_exp1.spw



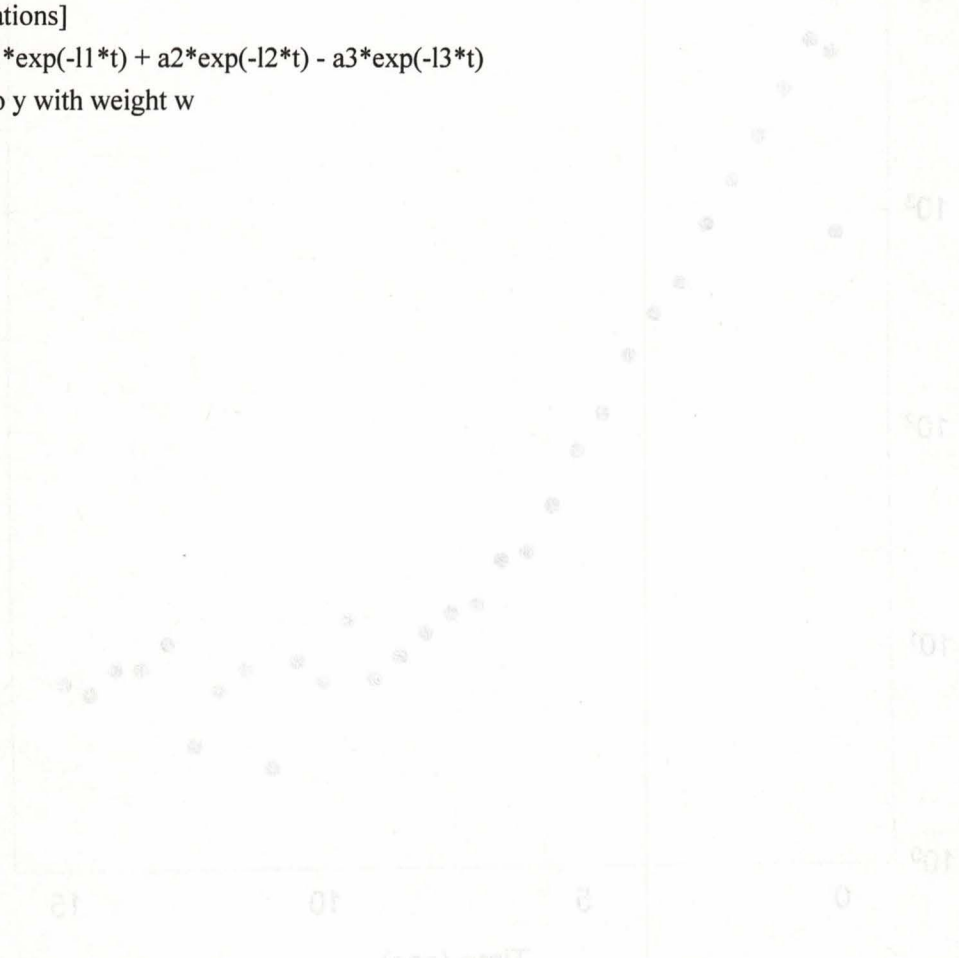
```
[Parameters]
m=1000
[Variables]
x=col(1)
y=col(2)
[Equations]
f=1/(1+(x-m)^2)
fit f to y
[Constraints]

[Options]
;stepsize=0.000001
;tolerance=0.000001
```



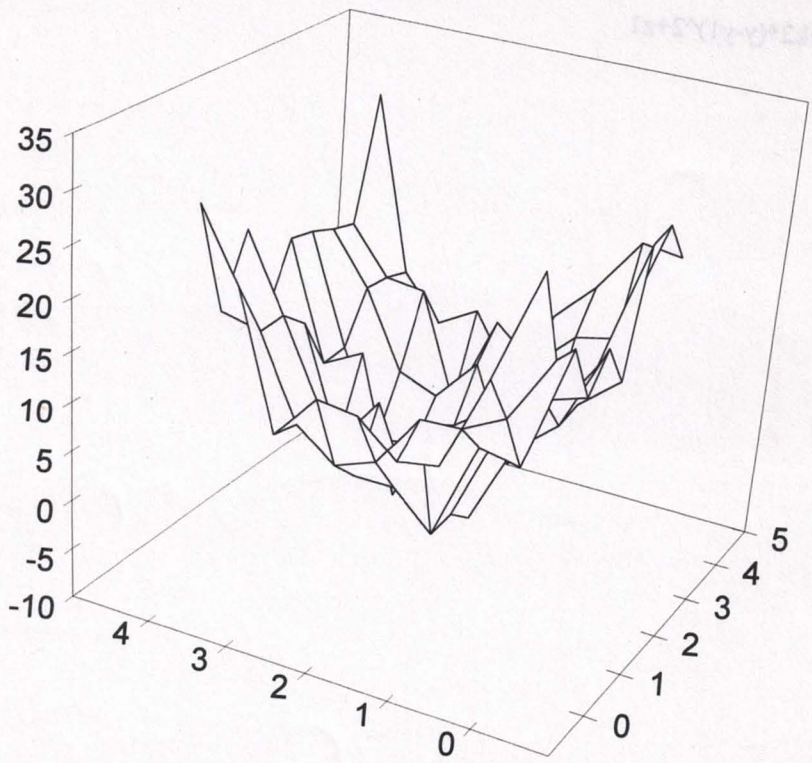
fit-exp2.spw



```
[Parameters]
a1=10000
a2=10
a3=10010
l1=1
l2=.01
l3=10
[Variables]
t=col(1)
y=col(2)
w=1/col(2)
[Equations]
f = a1*exp(-l1*t) + a2*exp(-l2*t) - a3*exp(-l3*t)
fit f to y with weight w
```



fit_exp3.spw



[Parameters]
k1=2
k2=4
x1=2
y1=2
z1=1

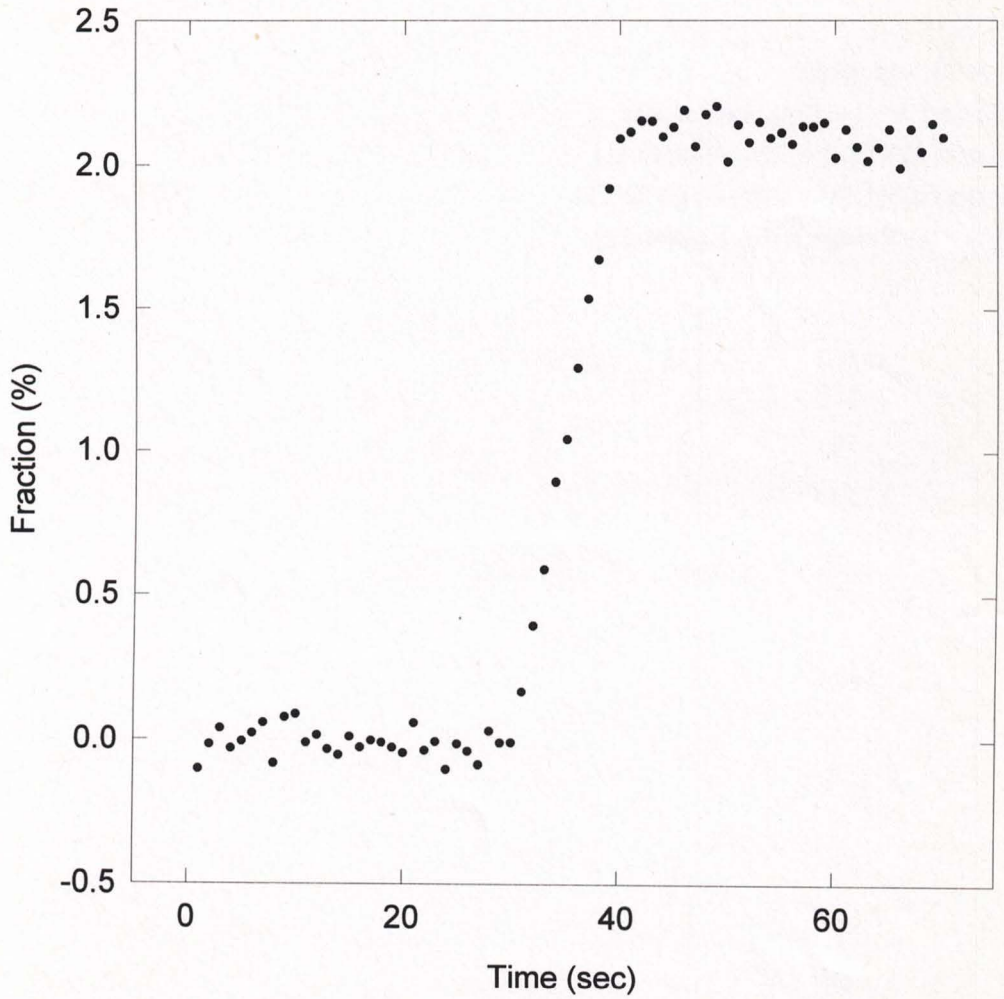
[Variables]
x=col(1)
y=col(2)
z=col(3)

[Equations]
f=k1*(x-x1)^2+k2*(y-y1)^2+z1
fit f to z

[Constraints]

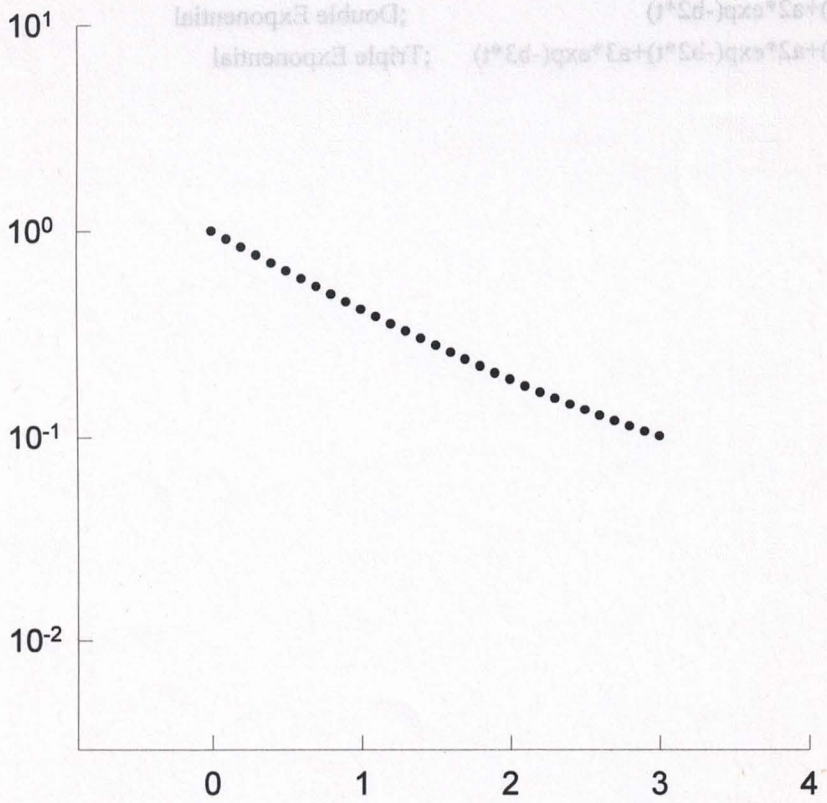
[Options]

fit-exp4.spw



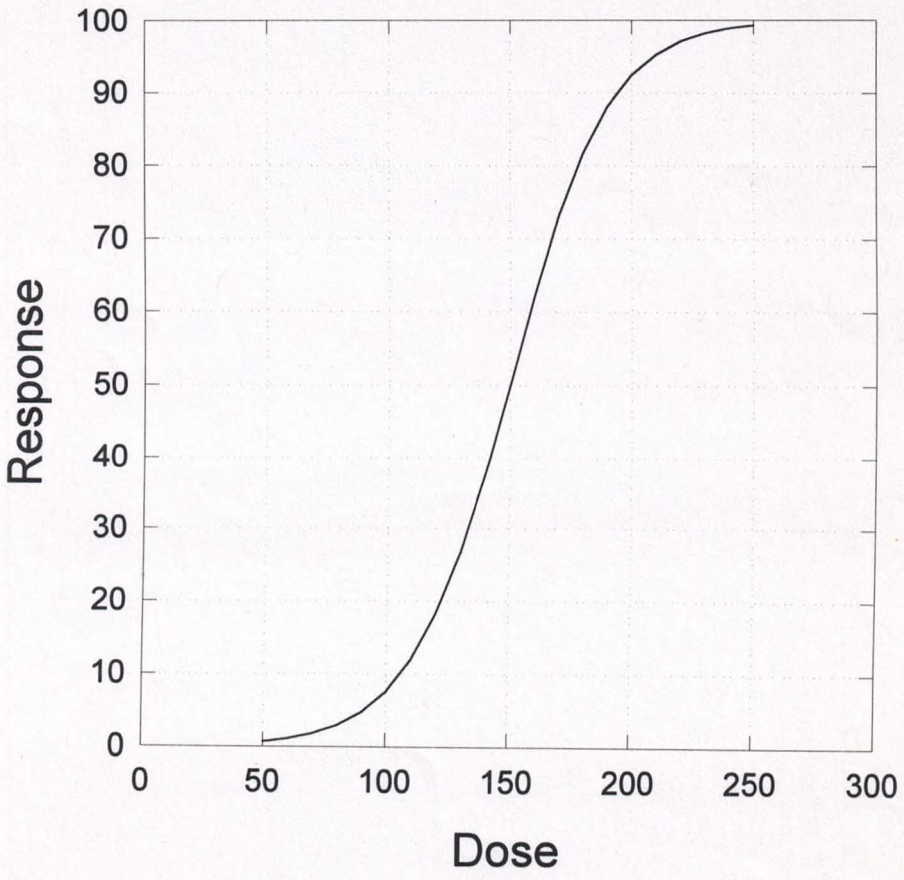
```
[Parameters]
x1 = 0.0
x2 = 0.0
x3 = 2.1
x4 = 2.1
T1 = 30
T2 = 40
[Variables]
y=col(2)
t=col(1)
[Equations]
t1 = 1
t4 = 70
f = if(t <= T1, region1(t),
        if(t <= T2, region2(t), region3(t)))
region1(t) = (x1*(T1-t) + x2*(t-t1))/(T1-t1)
region2(t) = (x2*(T2-t) + x3*(t-T1))/(T2-T1)
region3(t) = (x3*(t4-t) + x4*(t-T2))/(t4-T2)

fit f to y
```

fit-exp5.spw



```
[Parameters]
a1=1
b1=1
;a2=.1
;b2=.1
;a3=.01
;b3=.01
[Variables]
t=col(1)
y=col(2)
;t=col(3)
;y=col(4)
[Equations]
f=a1*exp(-b1*t)                                  ;Single Exponential
;f=a1*exp(-b1*t)+a2*exp(-b2*t)                   ;Double Exponential
;f=a1*exp(-b1*t)+a2*exp(-b2*t)+a3*exp(-b3*t)     ;Triple Exponential
fit f to y
[Constraints]

[Options]
```

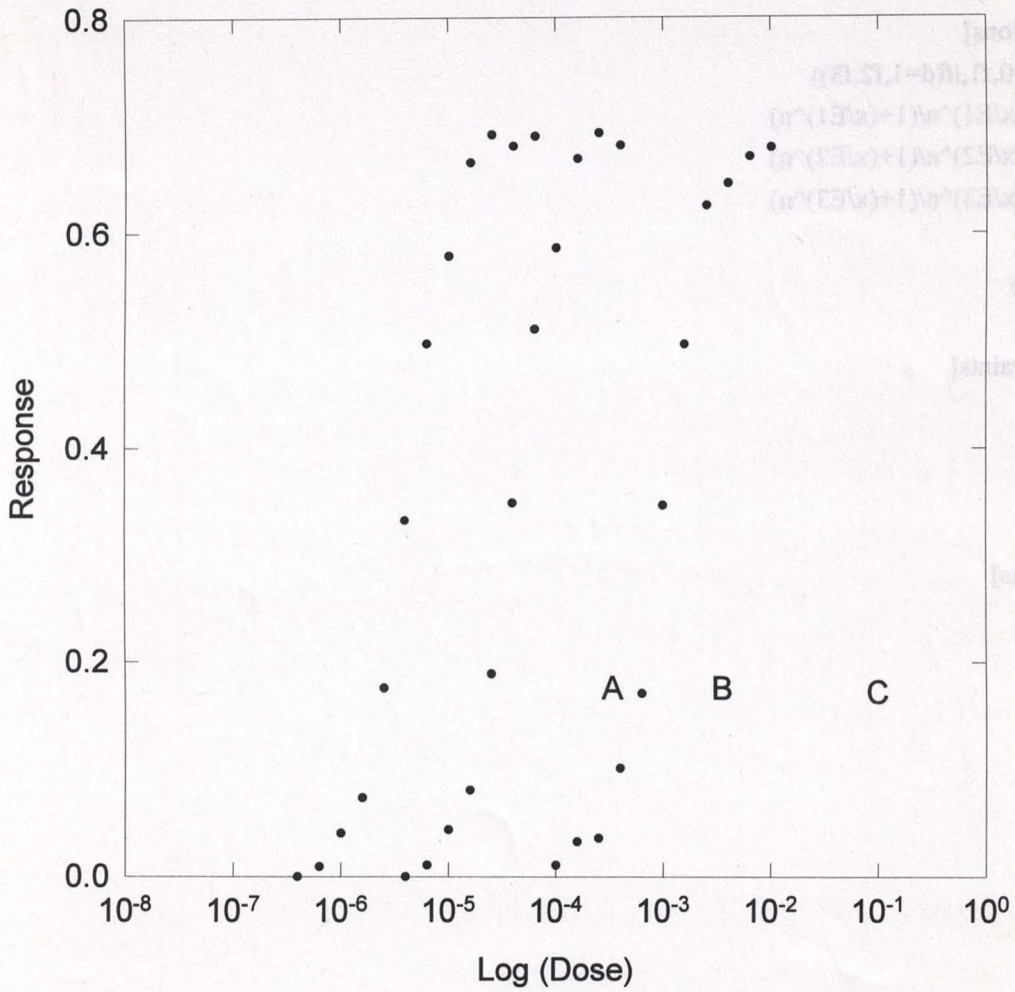
fit_exp6.spw


Response
Dose
0
10
20
30
40
50
60
70
80
90
100
0
50
100
150
200
250
300

[Parameters]
x=150
[Variables]
y=0
[Equations]
p1=100.96
p2=-.05010
p3=148.96
p4=-1.2080
f=p1/(1+exp(p2*(x-p3)))+p4-50
fit f to y
[Constraints]

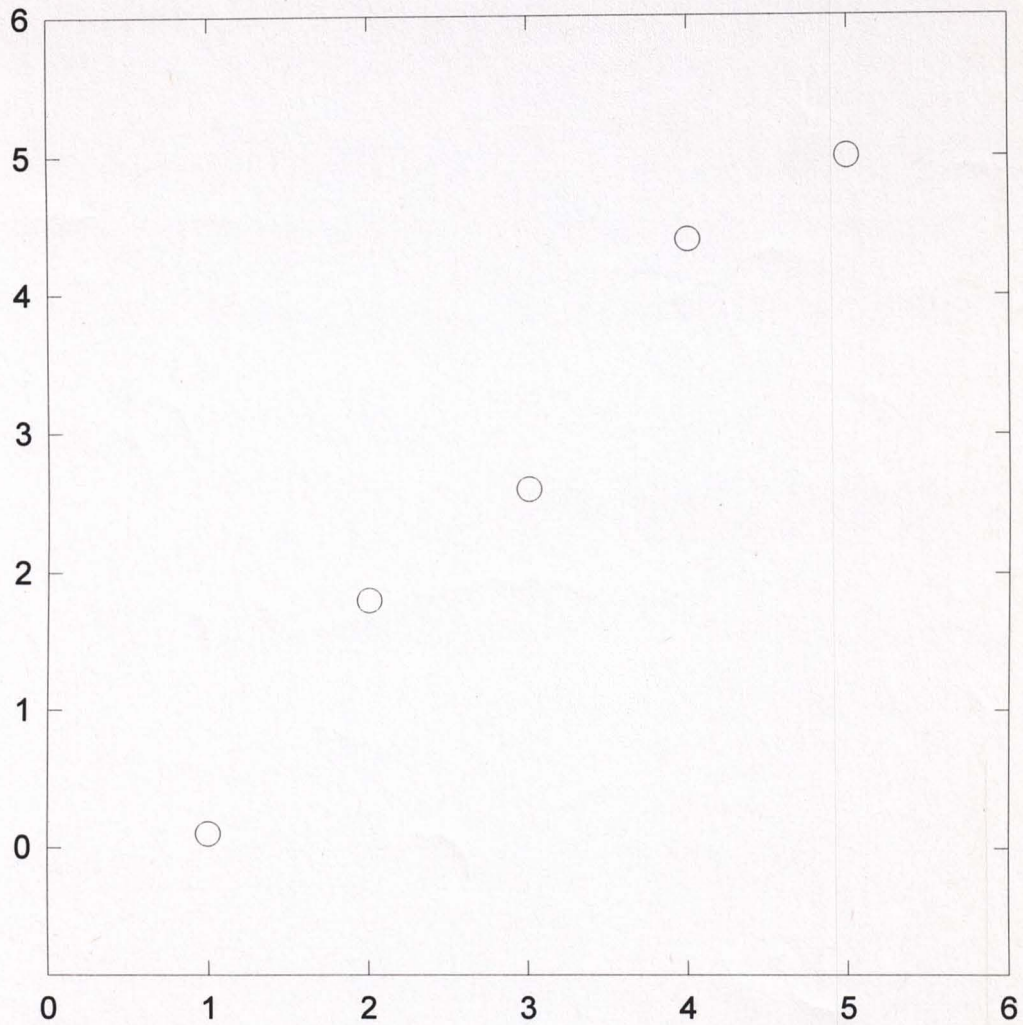
[Options]

fit_exp > spw

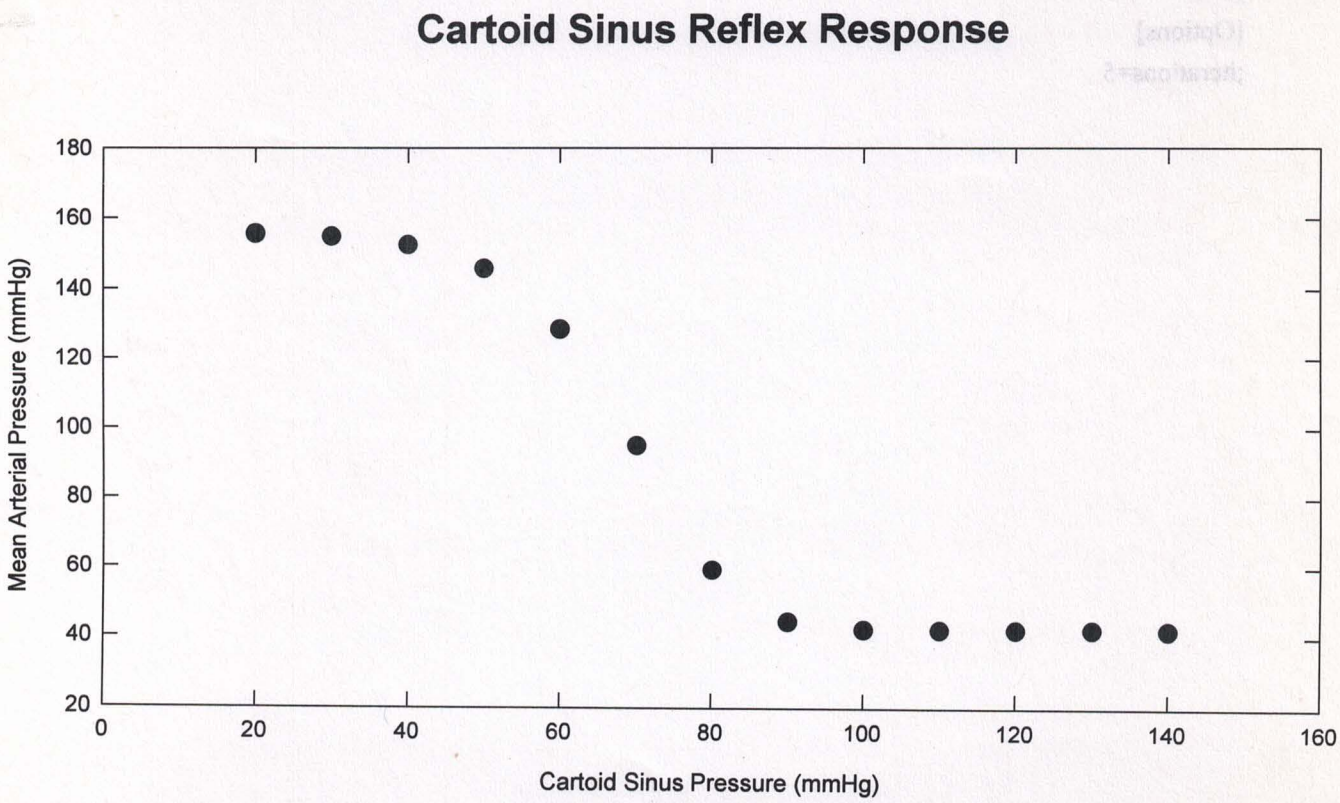

0.8
0.6
0.4
0.2
0.0
Response
A
B
C
10-8
10-7
10-6
10-5
10-4
10-3
10-2
10-1
100
Log (Dose)

```
[Parameters]
T=1
n=2
E1=1e-6
E2=1e-5
E3=1e-4

[Variables]
x=col(7)        ;concatonated x data
y=col(8)        ;concatonated y data
d=col(9)        ;function identifier

[Equations]
f=if(d=0,f1,if(d=1,f2,f3))
f1=T*(x/E1)^n/(1+(x/E1)^n)
f2=T*(x/E2)^n/(1+(x/E2)^n)
f3=T*(x/E3)^n/(1+(x/E3)^n)

fit f to y

[Constraints]
E1>0
E2>0
E3>0

[Options]
```

fit_tut1.spw



fit_TuT2.spw


Cartoid Sinus Reflex Response
Mean Arterial Pressure (mmHg)
180
160
140
120
100
80
60
40
20
0
20
40
60
80
100
120
140
160
Cartoid Sinus Pressure (mmHg)

```
[Parameters]
a=120    ;range for mean arterial pressure
b=0.1    ;slope coefficient
c=90     ;sinus pressure at maximum rate of change
d=40     ;minimum mean arterial pressure
;e=2
[Variables]
x=col(1)   y=col(2)
[Equations]
f=a/(1+exp(b*(x-c)))+d ;^e+d
fit f to y     ;fit statement
[Constraints]
;e>0
[Options]
;iterations=5
```