A NONLINEAR REGRESSION SOLVER FOR GAMS

ERWIN KALVELAGEN

ABSTRACT. This document describes a non-linear regression solver for GAMS.

1. Introduction

The non-linear regression solver NLS for GAMS calculates estimates θ for the non-linear statistical model:

$$y = f(X, \theta) + \varepsilon \tag{1}$$

The solver calculates

$$\min_{\theta} \sum_{i=1}^{n} [y_i - f(X_i, \theta)]^2 \tag{2}$$

using the nonlinear least squares package NL2SOL[16].

It is possible to use a GAMS NLP solver to provide initial values and through an option it is even possible to bypass NL2SOL and only use the solution provided by the NLP solver. In that case only statistics such as standard errors are calculated. As these quantities are not readily available using GAMS directly this solver provides therefore added-value even in case the solver functionality is not used.

GAMS brings to the table a number of important features in applied non-linear regression modeling:

- A powerful language for data manipulation. Compared to statistical systems GAMS is especially suited for non-regular data and for very large sparse structures.
- A number of powerful NLP solvers. The built-in nonlinear least squares solver NL2SOL in this package may fail to converge for certain models. In that case we can try to help it by using the GAMS NLP solvers to find a good solution to the least squares problem. When we pass this (near) optimal solution to NL2SOL it will have an easy task. This also applies for certain problems that may require a global solver due to presence of local optima. Furthermore, the GAMS NLP solvers can use bounds to protect difficult functions such as log(.) or divisions. These bounds may provide better reliability and also better performance. In this respect most statistical packages are less well-endowed: they often provide just one or two unconstrained solvers.
- GAMS brings you *automatic differentiation*, so regression models can use exact derivatives without asking the user to provide them. This can increase performance, reliability and a convenience factor that is not often

Date: November 2007.

found in statistical packages. The package R provides a package *Deriv* that implements automatic differentiation.

For an overview of nonlinear regression see [2, 35].

2. Usage

The model specified in GAMS should contain a dummy objective function (the solver will use a sum of squares objective internally) and a set of equations describing the fit. E.g.:

```
obj..     sse =n= 0;
fit(i)..  y(i) =e= b1*(1-exp[-b2*x(i)]);

b1.l = 500;
b2.l = 0.0001;

option nlp=nls;
model ols1 /obj,fit/;
solve ols1 minimizing sse using nlp;
```

Here `x(i)` and `y(i)` are parameters, and `b1` and `b2` are free variables. Note that the role of coefficients is reversed: b_1 and b_2 are coefficients in the statistical model but variables in the optimization model. Similarly x and y are parameters for the optimization model but variables in the statistical model.

It is always a good idea to assign starting values nonlinear variables in an optimization model. In many cases the user will know some reasonable numbers that make sense as starting points. If you don't provide a starting point GAMS may use a default of zero which is often a very poor choice as initial point for the optimization algorithm.

The algorithm implements an unconstrained least squares method so bounds on the variables are not allowed. As we may have used bounds to protect a previous solve by a GAMS NLP solver, we accept bounds on the variables but ignore them while issuing a warning message.

The estimates are returned as the levels of the variables. The marginals will contain the standard errors. The row levels reported are the residuals $\hat{\varepsilon} = y - \hat{y} = y - f(X, \hat{\theta})$. In addition a GDX file is written which will contain all regression statistics.

3. Linear models

For linear models

$$y = X\theta + \varepsilon \tag{3}$$

a linear regression solver `LS`[19] can be used. In fact it is better to use the linear solver as it is faster, more reliable and provides more statistical output for this particular class of models.

Many models that look non-linear can actually be reformulated into linear models. Firstly, all models that are nonlinear in X but linear in θ are just linear from a regression point of view. E.g. a model like:

$$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4 + b_5 x^5 \tag{4}$$

taken from the `wampler` data sets[39] from the NIST site `http://www.itl.nist.gov/div898/strd/lls/lls.shtml` is a polynomial problem but linear in the coefficients to estimate $b_0, ..., b_5$. See section 9.5 in [19].

Some models can be linearized by taking logarithms. E.g.

$$y = ae^{bx} \tag{5}$$

can be transformed to

$$\ln y = \ln a + bx \tag{6}$$

To be precise: this implies we used a multiplicative error:

$$y = ae^{bx}e^{\varepsilon} \tag{7}$$

A Cobb-Douglas production function of the form

$$Y = \gamma K^{\alpha} L^{\beta} \tag{8}$$

results in a linear model when taking logarithms:

$$\ln Y = \ln \gamma + \alpha \ln K + \beta \ln L \tag{9}$$

A hyperbolic relationship

$$y = \frac{x}{a + bx} \tag{10}$$

can be linearized as:

$$\frac{1}{y} = b + a\frac{1}{x} \tag{11}$$

4. Example: Fitting a CES production function

As an example consider a CES (Constant Elasticity of Substitution) production function, an important equation used in many economic models. A search in the GAMS model library shows a handful of models that have CES functions. A production function $Q = f(K, L)$ measures output given inputs consisting of the 'factors of production' (in our case we have 2 factors: labor L and capital K). A simple production function often used in the economic literature is the Cobb-Douglas production function [38]. It looks like:

$$Q = \lambda K^{\alpha} L^{\beta} \tag{12}$$

A more complicated function that is very well known is the CES production function. CES functions were introduced by [1], and are also called ACMS functions, after the authors. For more information on CES functions and their limitations in production theory see [17, 11]. The functional form of a CES production function is:

$$Q = \gamma \left[\delta L^{-\rho} + (1-\delta)K^{-\rho}\right]^{-\frac{\eta}{\rho}} \tag{13}$$

where L is labor, K is capital, Q is output. γ is called the 'efficiency parameter' ($\gamma > 0$), δ is the 'distribution parameter' ($0 < \delta < 1$), and ρ is the 'substitution parameter' ($-1 \le \rho \le \infty$). η denotes the degree of homogeneity of the function.

Takings logs, and renaming some parameters, we can write this as:

$$\ln Q = \gamma - \frac{\eta}{\rho} \ln\left(\delta L^{-\rho} + (1-\delta)K^{-\rho}\right) \tag{14}$$

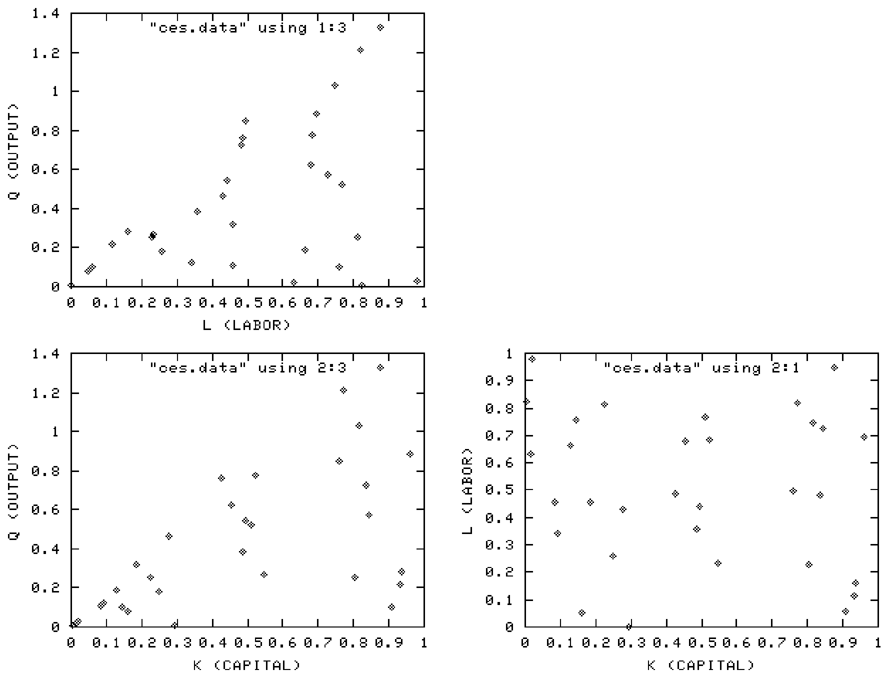

FIGURE 1. Scatter plots of the CES data set

As an aside it is noted that CES functions also have an application in Linear Programming theory. Interior point methods are often based on variants of a logarithmic barrier function. However, one can also devise an interior point algorithm for linear programming based on a CES function [26].

A data set from [13] is used as an example for our nonlinear regression problem. It is reproduced in table 1

The optimization model to be solved can be simply stated as:

$$\text{NLREG} \quad \underset{\gamma,\delta,\rho,\eta}{\text{minimize}} \sum_i r_i^2$$
$$\text{subject to} \quad \ln Q_i = \gamma - \frac{\eta}{\rho} \ln\left(\delta L_i^{-\rho} + (1-\delta) K_i^{-\rho}\right) + r_i$$

This model can be directly solved by a standard GAMS NLP solver such as CONOPT or MINOS:

4.0.1. *Model nls.gms.* [1]

```
$ontext

  Nonlinear least squares.

  Example: Estimation of a CES production function

  Data set: Table 22.4, page 724 of Griffiths, Hill and Judge,
     LEARNING AND PRACTICING ECONOMETRICS, Wiley, 1993.

```

[1] http://www.amsterdamoptimization.com/models/regression/nls.gms

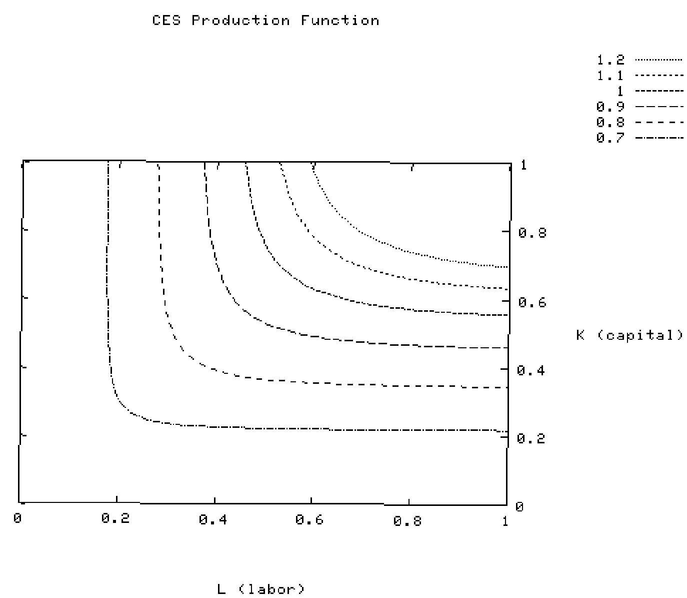

FIGURE 2. Contours of CES production function

L	K	Q	L	K	Q
0.228	0.802	0.256918	0.664	0.129	0.186747
0.258	0.249	0.183599	0.631	0.017	0.020671
0.821	0.771	1.212883	0.059	0.906	0.100159
0.767	0.511	0.522568	0.811	0.223	0.252334
0.495	0.758	0.847894	0.758	0.145	0.103312
0.487	0.425	0.763379	0.050	0.161	0.078945
0.678	0.452	0.623130	0.823	0.006	0.005799
0.748	0.817	1.031485	0.483	0.836	0.723250
0.727	0.845	0.569498	0.682	0.521	0.776468
0.695	0.958	0.882497	0.116	0.930	0.216536
0.458	0.084	0.108827	0.440	0.495	0.541182
0.981	0.021	0.026437	0.456	0.185	0.316320
0.002	0.295	0.003750	0.342	0.092	0.123811
0.429	0.277	0.461626	0.358	0.485	0.386354
0.231	0.546	0.268474	0.162	0.934	0.279431

TABLE 1. CES production function data set

```
  Erwin Kalvelagen, 2000

$offtext

set i 'observations' /i1*i30/;
set j 'parameters'   /L,K,Q/;
```

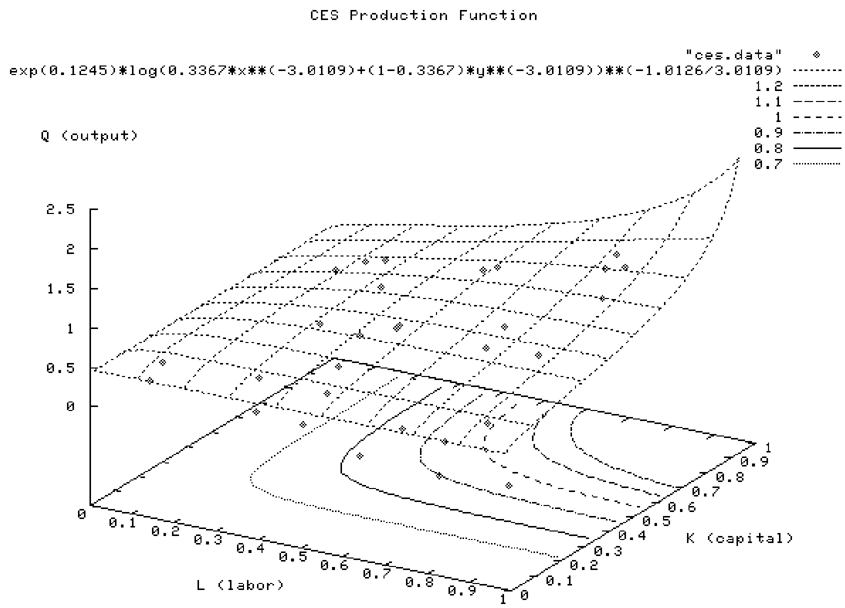

FIGURE 3. Surface of CES production function

```
table data(i,j)
         L          K          Q
i1       0.228      0.802      0.256918
i2       0.258      0.249      0.183599
i3       0.821      0.771      1.212883
i4       0.767      0.511      0.522568
i5       0.495      0.758      0.847894
i6       0.487      0.425      0.763379
i7       0.678      0.452      0.623130
i8       0.748      0.817      1.031485
i9       0.727      0.845      0.569498
i10      0.695      0.958      0.882497
i11      0.458      0.084      0.108827
i12      0.981      0.021      0.026437
i13      0.002      0.295      0.003750
i14      0.429      0.277      0.461626
i15      0.231      0.546      0.268474
i16      0.664      0.129      0.186747
i17      0.631      0.017      0.020671
i18      0.059      0.906      0.100159
i19      0.811      0.223      0.252334
i20      0.758      0.145      0.103312
i21      0.050      0.161      0.078945
i22      0.823      0.006      0.005799
i23      0.483      0.836      0.723250
i24      0.682      0.521      0.776468
i25      0.116      0.930      0.216536
i26      0.440      0.495      0.541182
i27      0.456      0.185      0.316320
i28      0.342      0.092      0.123811
i29      0.358      0.485      0.386354
i30      0.162      0.934      0.279431
;
```

```
parameters
    L(i)        'labor'
    K(i)        'capital'
    Q(i)        'output'
;

L(i) = data(i,'L');
K(i) = data(i,'K');
Q(i) = data(i,'Q');


variables
    gamma       'log of efficiency parameter'
    delta       'distribution parameter'
    rho         'substitution parameter'
    eta         'homogeneity parameter'
    residual(i) 'error term'
    sse         'sum of squared errors'
;

equations
    fit(i)     'the nonlinear model'
    obj        'objective'
;


obj..     sse =e= sum(i, sqr(residual(i)));
fit(i)..  log(Q(i)) =e=
             gamma - (eta/rho)*log[delta*L(i)**(-rho) + (1-delta)*K(i)**(-rho)]
             + residual(i);

* initial values
rho.l=1;
delta.l=0.5;
gamma.l=1;
eta.l=1;

model nls /obj,fit/;
solve nls minimizing sse using nlp;

display gamma.l, delta.l, rho.l, eta.l, sse.l;
```

This will provide the following results:

```
VARIABLE  gamma.L             =        0.124  log of efficiency parameter
VARIABLE  delta.L             =        0.337  distribution parameter
VARIABLE  rho.L               =        3.011  substitution parameter
VARIABLE  eta.L               =        1.013  homogeneity parameter
VARIABLE  sse.L               =        1.761  sum of squared errors
```

However we are missing output such as standard error that allows us to assess the quality of the fit. Other statistical packages such as CHAZAM [40] report these:

4.0.2. *Output of CHAZAM.* This run is used to verify the solution.

```
**********************************************************************
Hello/Bonjour/Aloha/Howdy/G Day/Kia Ora/Konnichiwa/Buenos Dias/Nee Hau/Ciao
Welcome to SHAZAM - Version 9.0  -  OCT 2000 SYSTEM=LINUX    PAR=   781
|_ * NONLINEAR LEAST SQUARES AND TESTING FOR AUTOCORRELATED ERRORS
|_ *
|_ * Example: Estimation of a CES production function
|_ *
|_ * Data set: Table 22.4, page 724 of Griffiths, Hill and Judge,
|_ *     LEARNING AND PRACTICING ECONOMETRICS, Wiley, 1993.
|_ *
|_ SAMPLE 1 30
|_ READ L K Q
   3 VARIABLES AND       30 OBSERVATIONS STARTING AT OBS       1

|_ GENR LOGQ=LOG(Q)
```

```
 |_ * Estimate the CES production function

 |_ NL 1 / NCOEF=4 PCOV ZMATRIX=Z COEF=BETA PREDICT=YHAT
 ...NOTE..SAMPLE RANGE SET TO:      1,     30
 |_ EQ LOGQ=GAMMA-(ETA/RHO)*LOG(DELTA*L**(-RHO)+(1-DELTA)*K**(-RHO))
 |_ COEF  RHO 1  DELTA .5  GAMMA 1  ETA 1
    3 VARIABLES IN   1 EQUATIONS WITH    4 COEFFICIENTS
         30 OBSERVATIONS

 REQUIRED MEMORY IS PAR=      22 CURRENT PAR=     781

 COEFFICIENT STARTING VALUES
 GAMMA      1.0000      ETA        1.0000      RHO        1.0000
 DELTA     0.50000
       100 MAXIMUM ITERATIONS, CONVERGENCE =   0.100000E-04

 INITIAL STATISTICS :

 TIME =      0.000 SEC.   ITER. NO.     0   FUNCT. EVALUATIONS     1
 LOG-LIKELIHOOD FUNCTION=   -45.75315
 COEFFICIENTS
    1.000000       1.000000       1.000000      0.5000000
 GRADIENT
   -25.82924       42.55772       5.501555      -6.222794

 INTERMEDIATE STATISTICS :

 TIME =      0.040 SEC.   ITER. NO.    15   FUNCT. EVALUATIONS    25
 LOG-LIKELIHOOD FUNCTION=  -0.6460435E-01
 COEFFICIENTS
   0.1246125       1.018006       2.750354      0.3581035
 GRADIENT
  -0.5488362      0.2883618      0.2721726E-01  -2.287144

 FINAL STATISTICS :

 TIME =      0.060 SEC.   ITER. NO.    25   FUNCT. EVALUATIONS    35
 LOG-LIKELIHOOD FUNCTION=  -0.3907423E-01
 COEFFICIENTS
   0.1244913       1.012594       3.010941      0.3366735
 GRADIENT
  -0.4790552E-03 -0.7407311E-04  0.1256401E-05 -0.1297012E-03
 ASYMPTOTIC COVARIANCE MATRIX
 GAMMA     0.47917E-02
 ETA       0.61486E-03  0.21055E-02
 RHO       0.50037E-01 -0.67552E-01   4.0663
 DELTA    -0.15592E-02  0.20209E-02 -0.13216      0.97548E-02
            GAMMA        ETA          RHO          DELTA

 MAXIMUM LIKELIHOOD ESTIMATE OF SIGMA-SQUARED =  0.58703E-01
 SUM OF SQUARED ERRORS =    1.7611
 GTRANSPOSE*INVERSE(H)*G  STATISTIC  -  =  0.11658E-08

       COEFFICIENT    ST. ERROR   T-RATIO

 GAMMA    0.12449     0.69222E-01  1.7984
 ETA       1.0126     0.45886E-01  22.068
 RHO       3.0109      2.0165      1.4931
 DELTA    0.33667     0.98766E-01  3.4088
 |_ END
 |_ STOP
```

The NLS solver tries to mimic the output that is provided by R, a popular statistical package in the public domain[32]:

```
> fm1<-nls(log(Q) ~ gamma-(eta/rho)*log(delta*L^(-rho)+(1-delta)*K^(-rho)),
+   c(rho=1,delta=0.5,gamma=1,eta=1),trace=TRUE,data=ces)
37.09680 :  1.0 0.5 1.0 1.0
2.295654 :  1.99986866 0.40119862 0.06698141 0.96205808
1.781744 :  3.0008698 0.3263035 0.1215438 1.0001874
1.761295 :  2.9001138 0.3418538 0.1233171 1.0143419
1.761125 :  3.0393385 0.3338332 0.1246505 1.0119903
```

```
1.761091 :  2.9857143 0.3382186 0.1242645 1.0129905
1.761081 :  3.0223077 0.3358118 0.1245782 1.0123984
1.761078 :  3.0038548 0.3371512 0.1244315 1.0127084
1.761077 :  3.0146639 0.3364070 0.1245208 1.0125321
1.761077 :  3.0088046 0.3368211 0.1244733 1.0126289
1.761077 :  3.0121094 0.3365907 0.1245004 1.0125748
1.761077 :  3.0102837 0.3367189 0.1244855 1.0126048
1.761076 :  3.0113034 0.3366476 0.1244938 1.0125881
1.761076 :  3.0107371 0.3366873 0.1244892 1.0125974
1.761076 :  3.0110526 0.3366652 0.1244918 1.0125922
1.761076 :  3.0108762 0.3366775 0.1244904 1.0125951
> summary(fm1)

Formula: log(Q) ~ gamma - (eta/rho) * log(delta * L^(-rho) + (1 - delta) *
    K^(-rho))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
rho    3.01088    2.32328   1.296   0.2064
delta  0.33668    0.13611   2.474   0.0202 *
gamma  0.12449    0.07834   1.589   0.1241
eta    1.01260    0.05068  19.979   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.2603 on 26 degrees of freedom

Number of iterations to convergence: 15
Achieved convergence tolerance: 9.961e-06

>
```

The model can be easily adapted to be solved with the NLS solver:

- replace the objective by a dummy objective
- remove all references to the variable `residual`
- add the statement `option NLP=LS;`.

4.0.3. *Model nls2.gms.* [2]

```
$ontext

  Nonlinear least squares.

  Example: Estimation of a CES production function

  Data set: Table 22.4, page 724 of Griffiths, Hill and Judge,
     LEARNING AND PRACTICING ECONOMETRICS, Wiley, 1993.

  Erwin Kalvelagen, 2000

$offtext

set i 'observations' /i1*i30/;
set j 'parameters'   /L,K,Q/;

table data(i,j)
          L          K           Q
i1      0.228      0.802      0.256918
i2      0.258      0.249      0.183599
i3      0.821      0.771      1.212883
i4      0.767      0.511      0.522568
i5      0.495      0.758      0.847894
i6      0.487      0.425      0.763379
i7      0.678      0.452      0.623130
i8      0.748      0.817      1.031485
i9      0.727      0.845      0.569498
i10     0.695      0.958      0.882497
i11     0.458      0.084      0.108827
```

[2] `http://www.amsterdamoptimization.com/models/regression/nls2.gms`

```
i12     0.981      0.021      0.026437
i13     0.002      0.295      0.003750
i14     0.429      0.277      0.461626
i15     0.231      0.546      0.268474
i16     0.664      0.129      0.186747
i17     0.631      0.017      0.020671
i18     0.059      0.906      0.100159
i19     0.811      0.223      0.252334
i20     0.758      0.145      0.103312
i21     0.050      0.161      0.078945
i22     0.823      0.006      0.005799
i23     0.483      0.836      0.723250
i24     0.682      0.521      0.776468
i25     0.116      0.930      0.216536
i26     0.440      0.495      0.541182
i27     0.456      0.185      0.316320
i28     0.342      0.092      0.123811
i29     0.358      0.485      0.386354
i30     0.162      0.934      0.279431
;


parameters
    L(i)         'labor'
    K(i)         'capital'
    Q(i)         'output'
;

L(i) = data(i,'L');
K(i) = data(i,'K');
Q(i) = data(i,'Q');


variables
    gamma        'log of efficiency parameter'
    delta        'distribution parameter'
    rho          'substitution parameter'
    eta          'homogeneity parameter'
    residual(i)  'error term'
    sse          'sum of squared errors'
;

equations
    fit(i)     'the nonlinear model'
    obj        'objective'
;


obj..     sse =n= 0;
fit(i)..  log(Q(i)) =e=
             gamma - (eta/rho)*log[delta*L(i)**(-rho) + (1-delta)*K(i)**(-rho)];

* initial values
rho.l=1;
delta.l=0.5;
gamma.l=1;
eta.l=1;

model nls /obj,fit/;
option nlp=nls;
solve nls minimizing sse using nlp;

display gamma.l, delta.l, rho.l, eta.l, sse.l;
```

This gives:

```
               S O L V E      S U M M A R Y

     MODEL   nls                 OBJECTIVE  sse
     TYPE    NLP                 DIRECTION  MINIMIZE
     SOLVER  NLS                 FROM LINE  90
```

```
**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      2 LOCALLY OPTIMAL
**** OBJECTIVE VALUE                1.7611

 RESOURCE USAGE, LIMIT          0.125      1000.000
 ITERATION COUNT, LIMIT         8         10000
 EVALUATION ERRORS              0             0

 ======================================================================
   Nonlinear Least Square Solver V1
   Erwin Kalvelagen, Amsterdam Optimization Modeling Group
   www.amsterdamoptimization.com
 ======================================================================

 Nonlinear Least Square Solver: NL2SOL

         Parameter      Estimate    Std. Error       t value      Pr(>|t|)
             gamma  -1.24491E-01   7.83443E-02  -1.58902E+00   1.24143E-01
             delta   3.36673E-01   1.36112E-01   2.47350E+00   2.02326E-02 *
               rho   3.01094E+00   2.32337E+00   1.29593E+00   2.06385E-01
               eta   1.01259E+00   5.06832E-02   1.99789E+01   2.66997E-17 ***
      ---
      Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Residual standard error:   2.60257E-01   on  26 degrees of freedom


 DLL version: _GAMS_GDX_237_2007-01-09
 GDX file: nls.gdx

                           LOWER          LEVEL          UPPER         MARGINAL

---- EQU obj               -INF           1.7611        +INF             .

  obj  objective

---- EQU fit  the nonlinear model

           LOWER          LEVEL          UPPER         MARGINAL

i1          1.3590        -0.3379         1.3590          .
i2          1.6950        -0.4234         1.6950          .
i3         -0.1930         0.3117        -0.1930          .
i4          0.6490        -0.1849         0.6490          .
i5          0.1650         0.2030         0.1650          .
i6          0.2700         0.4315         0.2700          .
i7          0.4730         0.1155         0.4730          .
i8         -0.0310         0.1440        -0.0310          .
i9          0.5630        -0.4576         0.5630          .
i10         0.1250        -0.0590         0.1250          .
i11         2.2180         0.0286         2.2180          .
i12         3.6330         0.0164         3.6330          .
i13         5.5860         0.2163         5.5860          .
i14         0.7730         0.3072         0.7730          .
i15         1.3150        -0.2755         1.3150          .
i16         1.6780         0.1344         1.6780          .
i17         3.8790        -0.0157         3.8790          .
i18         2.3010         0.0744         2.3010          .
i19         1.3770        -0.1166         1.3770          .
i20         2.2700        -0.5760         2.2700          .
i21         2.5390         0.0229         2.5390          .
i22         5.1501        -0.2322         5.1501          .
i23         0.3240         0.0301         0.3240          .
i24         0.2530         0.2131         0.2530          .
i25         1.5300         0.1619         1.5300          .
i26         0.6140         0.0186         0.6140          .
i27         1.1510         0.3062         1.1510          .
i28         2.0890         0.0677         2.0890          .
i29         0.9510        -0.2057         0.9510          .
i30         1.2750         0.0809         1.2750          .

                           LOWER          LEVEL          UPPER         MARGINAL
```

```
---- VAR gamma            -INF          -0.1245        +INF            0.0783
---- VAR delta            -INF           0.3367        +INF            0.1361
---- VAR rho              -INF           3.0109        +INF            2.3234
---- VAR eta              -INF           1.0126        +INF            0.0507
---- VAR sse              -INF           1.7611        +INF             .

  gamma  log of efficiency parameter
  delta  distribution parameter
  rho  substitution parameter
  eta  homogeneity parameter
  sse  sum of squared errors
```

5. Input formats

5.1. LS compatible format. This format is compact and is also used in the linear LS[19] solver. It has a dummy objective with a =N=, and a set of n =E= equations describing the fit:

```
obj..     sse =n= 0;
fit(i)..  y(i) =e= exp(-b1*x(i))/(b2+b3*x(i));
```

The solver will form automatically the least squares objective

$$\text{sse} = \sum_{i=1}^{n} \left[y_i - \frac{\exp(-b_1 x_i)}{b_2 + b_3 x_i} \right]^2 \tag{15}$$

and minimize this.

The above fragment completed with declarations is here:

```
*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------

variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= exp(-b1*x(i))/(b2+b3*x(i));

model nlfit /obj,fit/;

*-----------------------------------------------------------------------------
* initial values
*-----------------------------------------------------------------------------

b1.l = 0.1;
b2.l = 0.01;
b3.l = 0.02;

*-----------------------------------------------------------------------------
* solve
*-----------------------------------------------------------------------------

option nlp=nls;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;
```

For some complete examples see for instance sections , , and .

5.2. **NLP compatible format.** This format can be used when another NLP solver is used to find a solution:

```
obj..     sse =e= sum(i, sqr(r(i)));
fit(i)..  y(i) =e= b1 + b2*exp[-x(i)*b4] + b3*exp[-x(i)*b5] + r(i);
```

This is a traditional NLP formulation and a solver like CONOPT or MINOS can solve these models. This can be important if NL2SOL has troubles solving a particular model. E.g. in the following fragment we show how first CONOPT is called followed by NLS.

```
*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------

variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
    b4           'coefficient to estimate'
    b5           'coefficient to estimate'
    r(i)         'residuals'
;

equations
    fit(i)     'the non-linear model, NLP format'
    obj        'objective, NLP format'
;

obj..     sse =e= sum(i, sqr(r(i)));
fit(i)..  y(i) =e= b1 + b2*exp[-x(i)*b4] + b3*exp[-x(i)*b5] + r(i);

model nlfit /obj,fit/;

*-----------------------------------------------------------------------------
*  initial values
*-----------------------------------------------------------------------------


b1.l =    50;
b2.l =   150;
b3.l =  -100;
b4.l =     1;
b5.l =     2;

*-----------------------------------------------------------------------------
* solve first by conopt then feed solution to nls
*-----------------------------------------------------------------------------

option nlp=conopt;
solve nlfit minimizing sse using nlp;
option nlp=nls;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l,b5.l;
```

The solver link to NLS will remove the r_i variables from the model and replace the objective by:

$$\text{sse} = \sum_{i=1}^{n} \left(y_i - [b_1 + b_2 \exp(-x_i b_4) + b_3 \exp(-x_i b_5)]\right)^2 \qquad (16)$$

For a complete example see section 12.11.6.

6. Options

Options are to be specified in a text file called `nls.opt` which should be located in the current directory (or the *project directory* in case you run GAMS from the IDE under Windows).

To signal the solver to read an option file, you'll need to specify *model*.`optfile = 1;` as in the following example:

```
option nlp=nls;
model m /all/;
m.optfile=1;
solve m minimizing sse using nlp;
```

It is possible to let GAMS write the option file from within a model. This allows you to make the `.gms` file self-contained. Here is an example:

```
$onecho > nls.opt
* This option is needed in MGH10.gms
mxfcal 10000
$offecho
```

The following options are recognized:

maxn i**:**
Maximum number of cases or observations. This is the number of rows (not counting the dummy objective). When the number of rows is very large, this is probably not a regression problem but a generic NLP model. To protect against those, we don't accept models with an enormous number of rows.
(Default = 1000)

maxp i**:**
Maximum number of coefficients to estimate. This is the number of columns or variables (not counting the dummy objective variable). When the number of variables is very large, this is probably not a regression problem but a generic NLP model. To protect against those, we don't accept models with an enormous number of columns.
(Default = 25)

gdx_file_name s**:**
Name of the GDX file where results are saved.
(Default = *nls.gdx*)

mxfcal i**:**
The maximum number of function evaluations.
(Default = 200)

mxiter i**:**
The maximum number of iterations. This overrides the GAMS *iterlim* option.
(Default = *iterlim* option from GAMS).

x0prt i**:**
If set to 1 this will print the initial values to the log file.
(Default = 0)

covreq i**:**
The covariance matrix is formed based on this setting. If $i = 0$ no covariance matrix is calculated. Then also no standard errors can be formed. The other

possible values are:

$$V = \begin{cases} \hat{\sigma}^2 H^{-1} J^T J H^{-1} & \text{if } i = 1 \\ \hat{\sigma}^2 H^{-1} & \text{if } i = 2 \\ \hat{\sigma}^2 (J^T J)^{-1} & \text{if } i = 3 \end{cases}$$

(Default = 3)

solprt i**:**

If set to one this option will print the final values from the solver together with the gradients and the scale vector d. The output will go to the log file.
(Default = 0)

statpr i**:**

If set to zero the summary statistics from the solver NL2SOL will not be printed.
(Default = 1)

xctol ϵ_x**:**

The x-convergence tolerance. If a Newton step is tried that has scaled relative change defined by

$$\Delta_r = \frac{\max_i d_i |x_i^k - x_i^{k-1}|}{\max_i d_i |x_i^k| + |x_i^{k-1}|}$$

is smaller than this tolerance

$$\Delta_r \leq \epsilon_x$$

and if this step yields at most twice the predicted function decrease, then NL2SOL returns with the message *X-convergence*. It appears likely that x^k is within some tolerance of a strong local minimizer x^*.
(Default = square root of the machine precision = 1.490116E-08)

rfctol ϵ_r**:**

Relative function convergence tolerance. If the actual function reduction is at most twice what was predicted and

$$\frac{f(x^k) - q^k(x^k - H^{-1}\nabla f(x^k))}{f(x^k)} \leq \epsilon_r$$

(i.e. the function reduction predicted by the quadratic model is $\leq \epsilon_r f(x^k)$) then return with the message *Relative function convergence*.
(Default = 1.0E-10)

afctol ϵ_a**:**

Absolute function convergence tolerance. If the absolute value of the current function value $f(x^k)$ is less than ϵ_a return with the message *Absolute function convergence*. The sum of squared errors is close to zero so further significant improvement is not possible. (Note that the function value is equal to the sum of squared errors divided by two.)
(Default = 1.0E-20)

outlev i**:**

This option controls the number and length of iteration summary lines printed to the log file. Zero means print no iteration log. A positive number $i > 0$ will print a detailed log line every i^{th} iteration. A negative number $i < 0$ will print a shorter log line every $|i|$ iteration. See section 8 for more

information.
(Default = -1)

parprt i**:**
If one then the non-default floating point settings are reported. If zero, printing is omitted.
(Default = 1)

covprt i**:**
Flag whether to print the covariance matrix to the log.
(Default = 0)

delta0 δ_0**:**
A factor used in choosing the finite-difference step size used in computing the covariance matrix when *covreq* is one or two. For component i, a step size

$$\delta_0 \max(|x_i|, 1/d_i) \operatorname{sign}(x_i)$$

is used, where d is the current scale vector.
(Default = square root of the machine precision = 1.490116E-08)

dfac γ**:**
A factor used in updating the scale vector d. The updates have the form

$$d_i^1 = \max(\sqrt{||J_i||^2 + \max(s_{i,i}, 0)}, \gamma d_i)$$

Then d_i is set to

$$d_i := \begin{cases} d_i^1 & \text{if } d_i^1 > jtol \\ \max(d_i^0, jtol) & \text{otherwise} \end{cases}$$

(Default = 0.6)

dinit ∂**:**
The value to which the scale vector d is initialized. It can be zero as an update will be applied.
(Default = 0.0d0)

d0init ∂_0**:**
The value to which d^0 is initialized.
(Default = 1.0)

jtinit τ_0**:**
The value to which *jtol* is initialized.
(Default = 1.0E-6)

tuner τ**:**
This options helps decide when to check for false convergence and to consider switching models. This is done if the actual function decrease from the current step is no more than τ times its predicted value.
(Default = 0.1)

skip_solver i**:**
If set to one or two the solver NL2SOL is skipped and the provided solution will be used to produce statistics (standard errors etc.). In general it is better to call NL2SOL even if the initial point is already optimal. If set to one there will be check for zero gradients. If set to two no checks will performed.

gradtesttol ϵ_t**:**
Tolerance used in checking gradient $\partial f/\partial x_i$ if option `skip_solver = 1` is

used.
(Default = 1.0E-2)

For the more technical options please refer to background information in [16].

7. Nonlinear Least Squares

The least square solver minimizes the quadratic function

$$\min_{\theta} \sum_{i=1}^{n} r_i(\theta)^2 \tag{17}$$

where r_i are the residuals $r_i = y_i - f(X_i, \theta)$.

The algorithm used will actually minimize

$$\min_{\theta} \frac{1}{2} \sum_{i=1}^{n} r_i(\theta)^2 \tag{18}$$

so the objective function values printed to the log are equal to

$$\frac{\text{sse}}{2} \tag{19}$$

A standard way of solving such problems is the Gauss-Newton method. Writing the problem as

$$\min F(x) = \frac{1}{2}||r(x)||_2^2 \tag{20}$$

then the gradient is

$$\nabla F(x) = J(x)^T r(x) \tag{21}$$

where $J(x)$ is the *Jacobian* matrix of first derivatives of $r(x)$. The *Hessian* is formed by:

$$\nabla^2 F(x) = H(x) = J(x)^T J(x) + \sum_{i=1}^{n} r_i(x) \nabla^2 r_i(x) \tag{22}$$

The Gauss-Newton method is based on the observation that for small residuals the second term is small so the Hessian can be approximated by its first term $J(x)^T J(x)$. A Newton iteration can be stated as:

$$x^{k+1} = x^k - \left[\nabla^2 F(x^k)\right]^{-1} \nabla F(x^k) \tag{23}$$

or using the approximation:

$$x^{k+1} = x^k - \left[J(x^k)^T J(x^k)\right]^{-1} J(x^k)^T r(x^k) \tag{24}$$

In practice such an iteration is performed as:

$$x^{k+1} = x^k + d^k \tag{25}$$

where d^k is solved from:

$$J(x^k)^T J(x^k) d^k = -J(x^k)^T r(x^k) \tag{26}$$

For large deviation problems it is often better to augment the Gauss-Newton model by a matrix S^k:

$$\left[J(x^k)^T J(x^k) + S^k\right] d^k = -J(x^k)^T r(x^k) \tag{27}$$

The optimization algorithm inside [16] adaptively switches between the Gauss-Newton Hessian approximation and the augmented approximation that uses a quasi-Newton update.

8. Solver log

A typical solver log will look like:

```
   it    nf      f        reldf      preldf     reldx

    0     1  0.740E+04
    1     3  0.545E+04  0.263E+00  0.245E+00  0.334E-01
    2     5  0.196E+04  0.640E+00  0.561E+00  0.787E-01
    3     6  0.110E+04  0.438E+00  0.104E+01  0.270E+00
    4     7  0.274E+03  0.752E+00  0.761E+00  0.238E+00
    5     8  0.257E+03  0.623E-01  0.613E-01  0.113E+00
    6     9  0.257E+03  0.104E-02  0.102E-02  0.910E-02
    7    10  0.257E+03  0.174E-05  0.167E-05  0.478E-03
    8    11  0.257E+03  0.352E-08  0.352E-08  0.246E-04
    9    12  0.257E+03  0.266E-13  0.267E-13  0.406E-07

Relative function convergence.

 function     0.256524E+03   reldx        0.406402E-07
 func. evals      12         grad. evals      10
 preldf       0.266585E-13   npreldf      0.266585E-13
```

The column headers indicate the following:

it: Iteration number
nf: Number of function/gradient evaluations.
f: The current solver objective value f^k. This is half of the sum of squares.
reldf: Relative difference in solver objection function: $(f^{k-1} - f^k)/f^{k-1}$.
preldf: The predicted relative difference in solver objection function.
reldx: The scaled relative change in the variables x.

The log may look different if the option *outlev* is used.

The solver may request function and gradient evaluations separately. The GAMS nonlinear interpreter provides both in one call. We therefore keep track when the vector of variable changes x (the solver provides this information) and if this is not the case we can return gradients already calculated in the previous function value request without calling the interpreter for another evaluation. This means that the actual number of gradient evaluations is equal to the number of function evaluations.

The return status is printed next. Hopefully it contains a form of the word *Converged*. Here are the possible codes:

X-convergence: The scaled relative difference between the current parameter vector x and a locally optimal parameter vector is very likely at most *xctol*.

Relative function convergence: The relative difference between the current function value and its locally optimal value is very likely at most *rfctol*.

X- and relative function convergence: Both x- and relative function convergence.

Absolute function convergence: The current function value is at most *afctol* in absolute value.

Singular convergence: The hessian near the current iterate appears to be singular or nearly so, and a step of length at most *lmax0* is unlikely to yield a relative function decrease of more than *rfctol*.

False convergence: The iterates appear to be converging to a noncritical point. This may mean that the convergence tolerances *afctol*, *rfctol*, *xctol* are too small for the accuracy to which the function and gradient are being computed, that there is an error in computing the gradient, or that the function or gradient is discontinuous near x.

Function evaluation limit: Function evaluation limit reached without other convergence.

Iteration limit: Iteration limit reached without other convergence.

9. Statistics

When solving a nonlinear least squares problem the following statistics are printed:

```
          Parameter      Estimate    Std. Error        t value      Pr(>|t|)
                 b1   1.67451E+00   8.79896E-02   1.90307E+01   2.24178E-41 ***
                 b2  -1.39274E-01   4.11820E-03  -3.38191E+01   5.97082E-71 ***
                 b3   2.59612E-03   4.18565E-05   6.20242E+01   8.50890-107 ***
                 b4  -1.72418E-03   5.89319E-05  -2.92572E+01   6.13314E-63 ***
                 b5   2.16648E-05   2.01298E-07   1.07626E+02   5.78904-141 ***
       ---
       Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

  Residual standard error:   1.63545E-01   on  146 degrees of freedom
```

The standard errors for the estimates θ_i are calculated from the asymptotic variance-covariance matrix

$$V = \hat{\sigma}^2 (J^T J)^{-1} \tag{28}$$

as described in [23] (so we can compare the standard errors with the benchmark reference results) and [10]. Other packages may use a different definition:

$$V = \hat{\sigma}^2 H^{-1} \tag{29}$$

Here J is the *Jacobian* matrix and H the *Hessian* matrix. Using an option NLS can use a different definition of the variance-covariance matrix.

The quantity $\hat{\sigma}^2$ is the estimated error variance:

$$\hat{\sigma}^2 = \frac{\text{sse}}{\text{df}} = \frac{\sum_{i=1}^n \hat{\varepsilon}^2}{n-p} \tag{30}$$

where n is the number of observations and p is the number of estimates.

The standard errors are simply the square roots of the diagonal elements of V:

$$\text{se}_i = \sqrt{v_{i,i}} \tag{31}$$

The test statistic or t-values are calculated as:

$$t_i = \frac{\beta_i}{\text{se}_i} \tag{32}$$

i.e. the estimates divided by their standard error.

The t values need to be compared to the Student's t distribution. We do this for you and produce socalled p-values. These values give probabilities for the two sided test H_0: $b_i = 0$ against H_1: $b_i \neq 0$. The formal calculation is done as:

$$p\text{-value} = \text{tdist}(|t_i|, n-p, 2) \tag{33}$$

Often a coefficient is called significant if the p value is ≤ 0.05. The final column forms a simple 'bar chart' for the significance levels. A significant coefficient (i.e. p value ≤ 0.005) is marked with one or more stars.

The residual standard error is the value σ defined above.

The Student t distribution is calculated using an implementation of the incomplete beta function from [7, 5]. The F distribution function is calculated via a chi-square distribution function which is based on the incomplete gamma function from [36]. These functions are also used in GAMS, see [20].

10. GDX output

The solver will write a GDX file named *nls.gdx* by default (the name can be changed using an option, see section 6). This GDX file will contain all of the summary statistics and in addition the variance-covariance matrix.

The content of the GDX file looks like:

```
C:\regression\docs>gdxdump  nls.gdx symbols
*  GDX dump of nls.gdx
*  Library in use : C:\PROGRA~1\GAMS22.6
*  Library version: GDX Library   ALFA 16Nov07 WIN.FR.NA 22.6 239.000.000.vis P3PC
*  File version   : _GAMS_GDX_237_2007-01-09
*  Producer       : nls.f90
*  File format    :    6
*  Compression    :    0
*  Symbols        :   13
*  Unique Elements:  264
   Symbol   Dim Type  Explanatory text
 1 confint    3  Par  confidence intervals
 2 covar      2  Par  variance-covariance matrix
 3 df         0  Par  Degrees of freedom
 4 estimate   1  Par  Estimated coefficients
 5 grad       1  Par  gradient
 6 jac        2  Par  Jacobian
 7 pval       1  Par  p values
 8 resid      1  Par  residuals
 9 resvar     0  Par  Residual variance
10 rss        0  Par  Residual sum of squares
11 se         1  Par  Standard errors
12 sigma      0  Par  Standard error
13 tval       1  Par  t values
C:\regression\docs>
```

Here follows a description for each of the items:

confint:
Confidence intervals for the estimates $\hat{\beta}$. The $1-\alpha\%$ confidence interval for estimate $\hat{\beta}_i$ is given by:

$$[\hat{\beta}_i - \text{se}_i t_{n-p;\frac{\alpha}{2}}, \hat{\beta}_i + \text{se}_i t_{n-p;\frac{\alpha}{2}}]$$

where $t_{n-p;\frac{\alpha}{2}}$ indicates the critical value for the Student's t distribution. To calculate these we use the algorithm from [15].

The confidence intervals are given for different α's. For a model that demonstrates how the confidence intervals can be retrieved see section 12.19.3.

covar:
The variance-covariance matrix. The indices are composed from the variable names in the model. An example of how to read the variance-covariance matrix is shown in section 12.19.1.

df:
Degrees of freedom: $\text{df} = n - p$ (i.e. the number of observations minus the number of parameters to estimate).

estimate:
The vector (of length p) of estimates $\hat{b}$. These are the same as returned in the solution.

grad:
The gradient ∇F of the objective function. See equation 21. A vector of length p.

jac:
The Jacobian ∇r of the residual function $r(\theta) = y - f(X, \hat{\theta})$. An $n \times p$ matrix.

pval:
A vector of length p with p-values given by (33).

resid:
A vector of length n with the residuals $\hat{\varepsilon} = y - \hat{y} = y - f(X, \hat{\theta})$.

resvar:
The residual variance $\frac{\text{RSS}}{\text{df}} = \frac{\text{RSS}}{\text{n-p}} = \hat{\sigma}^2$.

rss:
The residual sum of squares $\text{RSS} = \sum_{i=1}^{n} \varepsilon_i^2$

se:
The standard errors, vector of length p as defined by (31).

sigma:
Standard error of the regression model $\hat{\sigma}$ (30).

tval:
The t values, a vector of length p, as defined in (32).

11. Plotting

It is often desirable to get a better understanding of the fit using graphical tools such as scatter plots. Typical plots are scatter plots to assess the relation between the independent and dependent variables. Another interesting post-regression graph is to plot residuals to see if they are approximately normally distributed.

11.1. **Gnuplot scatter plots. Gnuplot** is a popular charting package among GAMS users. It can be downloaded from `http://www.gnuplot.info/`. A convenient way to run it from GAMS is to let it write a PNG file and then call a viewer associated with PNG files to display it. For the **Chwirut2** model in section 12.4.2, we used the following code:

```
*
* plot results
*

file pltdat /Chwirut2.dat/;
loop(i,
    put pltdat data(i,'x'):17:5,data(i,'y'):17:5/;
);
putclose;


file plt /Chwirut2.plt/;
put plt;
put "b1=",b1.l:0:16/;
```

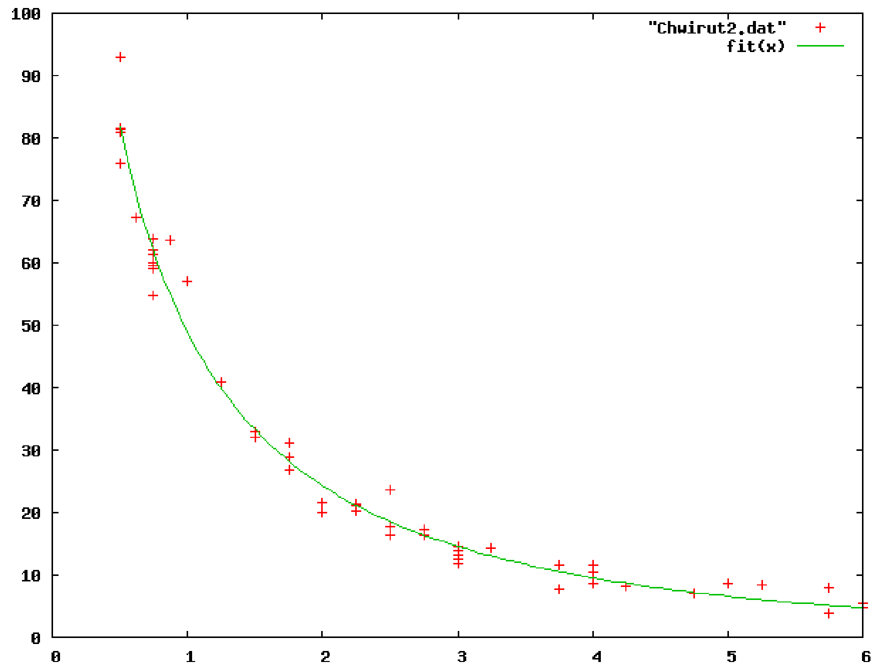

FIGURE 4. Scatter plot using Gnuplot

```
put "b2=",b2.l:0:16/;
put "b3=",b3.l:0:16/;
put "fit(x)=exp(-b1*x)/(b2+b3*x)"/;
putclose 'set term png'/
         'set output "Chwirut2.png"'/
         'plot "Chwirut2.dat",fit(x)'/;


execute '=wgnuplot.exe Chwirut2.plt';
execute '=shellexecute Chwirut2.png';
```

In the first part we write a data file with our data points (x, y). In the second part we write a command file `pontius.plt` for use with Gnuplot. The commands start with defining our fitted function $f(x) = \exp(-b_1 x)/(b_2 + b_3 x)$ where we substitute the estimates for b_1, b_2 and b_3. Then we tell Gnuplot to generate a PNG file. The plot command instructs Gnuplot to plot both the data points and the fitted function. Finally we call gnuplot followed by a call to `shellexecute` which will call the program associated with PNG files.

11.2. **Gnuplot residual plots.** It is not very difficult to produce a plot of the residuals:

```
*
* plot results
*

file pltdat /Chwirut2res.dat/;
loop(i,
    put pltdat data(i,'x'):17:5,fit.l(i):17:5/;
);
putclose;
```

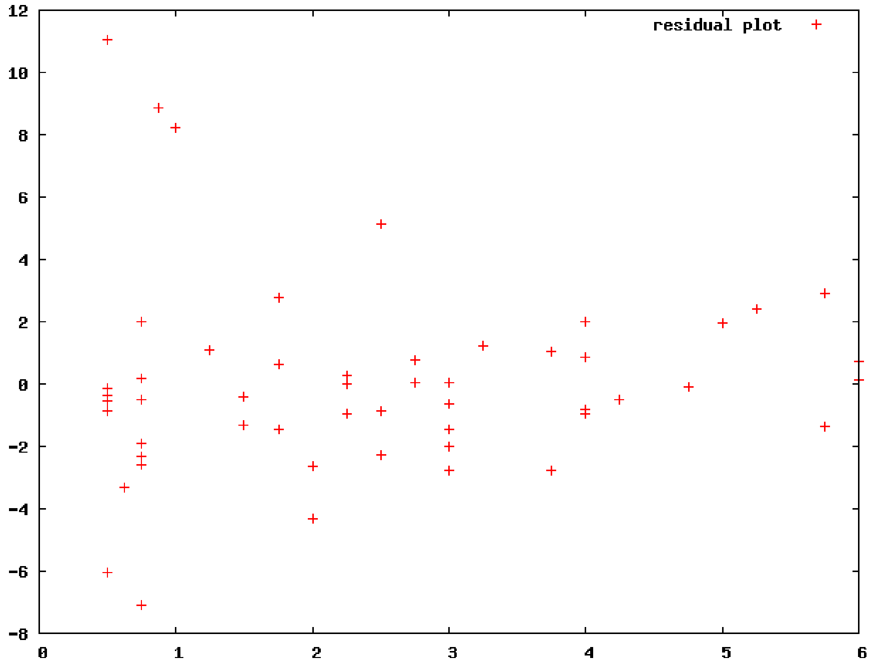

FIGURE 5. Gnuplot residual plot

```
file plt /Chwirut2res.plt/;
put plt;
putclose 'set term png'/
         'set output "Chwirut2res.png"'/
         'plot "Chwirut2res.dat" title "residual plot"'/;


execute '=wgnuplot.exe Chwirut2res.plt';
execute '=shellexecute Chwirut2res.png';
```

12. EXAMPLES

Most of these example come from the NIST nonlinear regression dataset [23].

12.1. **Sas.** This is a simple model that reproduces some results from a run using SAS PROC NLIN.

12.1.1. *Model Sas.gms.* [3]

```
$ontext

   Example of estimation of the Brody growth curve

   Reference:
        Miroslav Kaps, William R. Lamberson
        Biostatistics for Animal Science
        2004 CABI Publishing
```

[3] www.amsterdamoptimization.com/models/regression/Sas.gms

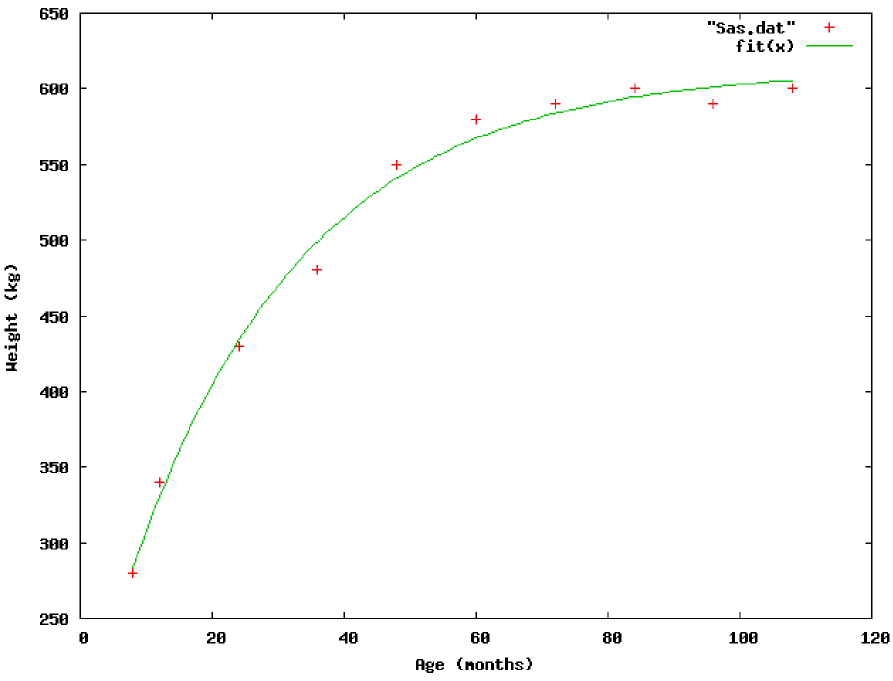

FIGURE 6. Brody growth curve

```
SAS Input:

DATA a;
INPUT age weight @@;
DATALINES;
   8  280     12  340     24  340     36  480     48  550
   60 580     72  590     84  600     96  590    108  600
;
PROC NLIN;
   PARMS A=600 weight0=280 k=0.05
   MODEL weight=A-(A-weight0)*exp(-k*(age-8));
   RUN;


SAS Output

                 Dependent Variable weight
                    Method: Gauss-Newton

                      Iterative Phase
                                                  Sum of
     Iter          A       weight0         k      Squares
        0      600.0         280.0    0.0500       2540.5
        1      610.2         285.8    0.0355       1388.7
        2      612.2         283.7    0.0381        966.9
        3      612.9         283.9    0.0379        965.9
        4      612.9         283.9    0.0380        965.9
        5      612.9         283.9    0.0380        965.9

           NOTE: Convergence criterion met.

                          Sum of        Mean                 Approx
Source             DF    Squares      Square     F Value     Pr > F
Regression          3    2663434      887811      446.69     <.0001
Residual            7      965.9       138.0
```

```
 Uncorrected Total 10     2664400
 Corrected Total    9      124240

                         Approx
 Parameter    Estimate   Std Error   Approximate 95% Confidence Limit
 A              612.9     9.2683      590.9       634.8
 weight0        283.9     9.4866      261.5       306.3
 k              0.0380    0.00383     0.0289      0.0470

             Approximate Correlation Matrix
                A            weight0             k

 A          1.0000000       0.2607907     -0.8276063
 weight0    0.2607907       1.0000000     -0.4940824
 k         -0.8276063      -0.4940824      1.0000000


$offtext


*-----------------------------------------------------------------------------
* data represents weight of an Angus cow at different ages
*-----------------------------------------------------------------------------

set i 'data points' /i1*i10/;

table data(i,*)
      weight    age
*       kg    months
  i1    280      8
  i2    340     12
  i3    430     24
  i4    480     36
  i5    550     48
  i6    580     60
  i7    590     72
  i8    600     84
  i9    590     96
  i10   600    108
;

parameter Weight(i), Age(i);
Weight(i) = data(i,'weight');
Age(i) = data(i,'age');


*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------

scalar Age0 'initial age (months)' /8/;

variables
    sse       'sum of squared errors'
    A         'Estimated asymptotic (mature) weight'
    Weight0   'Estimated initial weight at age0'
    k         'Estimated maturing weight index'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  Weight(i) =e= A - (A-Weight0) * exp[-k*(Age(i)-Age0)];

option nlp=nls;
model nlfit /obj,fit/;

*-----------------------------------------------------------------------------
* initial values
```

```
*-----------------------------------------------------------------------------

A.l = 600;
weight0.l = 280;
k.l = 0.05;

solve nlfit using nlp minimizing sse;
display sse.l,A.l,weight0.l,k.l;



*-----------------------------------------------------------------------------
* load and display the confidence intervals
*-----------------------------------------------------------------------------


sets
   alpha /'95%'/
   v /A,Weight0,k/
   interval /lo,up/
;

parameter fitresult(v,*);
fitresult('A',      'Estimate') = A.l;
fitresult('Weight0','Estimate') = Weight0.l;
fitresult('k',      'Estimate') = k.l;
fitresult('A',      'Std.Err.') = A.m;
fitresult('Weight0','Std.Err.') = Weight0.m;
fitresult('k',      'Std.Err.') = k.m;
display fitresult;


parameter
   confint(alpha,v,interval) "confidence intervals"
   confint95(v,interval)     "95% confidence interval"
   covar(v,v)                "Covariance matrix"
   corr(v,v)                 "Correlation matrix"
;
execute_load 'nls.gdx',confint,covar;
confint95(v,interval) = confint('95%',v,interval);
display confint95;

alias(v,vv);
corr(v,vv) = covar(v,vv)/(sqrt(covar(v,v))*sqrt(covar(vv,vv)));
option decimals=7;
display corr;
```

12.2. **Greene931.** This model is from [12] Section 9.3.1. The regression is performed on a non-linear consumption function:

$$C = \alpha + \beta Y^{\gamma} + \varepsilon \tag{34}$$

The initial values are constructed by setting $\gamma = 1$ and applying linear regression on

$$C = \alpha + \beta Y + \varepsilon \tag{35}$$

12.2.1. *Model Greene931.gms.* [4]

```
$ontext

   Section 9.3.1 from Greene.

   Nonlinear Least Squares Regression example

   Erwin Kalvelagen, nov 2007

   ==============================================================
```

[4] www.amsterdamoptimization.com/models/regression/Greene931.gms

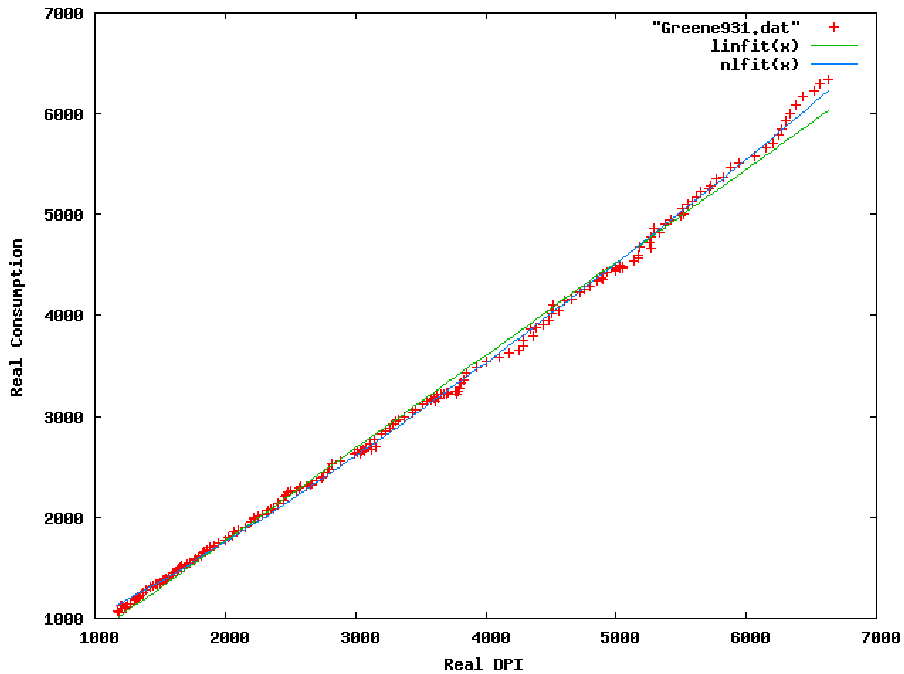

FIGURE 7. Consumption function: linear and non-linear fit

```
   Model:
        C = alpha + beta * Y ** gamma

   Starting values:
        gamma = 1
        alpha, beta from linear regression C = alpha + beta * Y

   Reference:
       Greene, Econometric Analysis, 5th Edition

$offtext

*---------------------------------------------------------------------------
* data
*---------------------------------------------------------------------------

set i /i1*i204/;

table data(i,*) 'Greene appendix table 5.1 (selected columns)'
         Year qtr realgdp  realcons realinvs realgovt realdpi
  i1     1950  1  1610.5   1058.9   198.1    361.0    1186.1
  i2     1950  2  1658.8   1075.9   220.4    366.4    1178.1
  i3     1950  3  1723.0   1131.0   239.7    359.6    1196.5
  i4     1950  4  1753.9   1097.6   271.8    382.5    1210.0
  i5     1951  1  1773.5   1122.8   242.9    421.9    1207.9
  i6     1951  2  1803.7   1091.4   249.2    480.1    1225.8
  i7     1951  3  1839.8   1103.9   230.1    534.2    1235.8
  i8     1951  4  1843.3   1110.5   210.6    563.7    1238.5
  i9     1952  1  1864.7   1113.6   215.6    584.8    1238.5
  i10    1952  2  1866.2   1135.1   197.7    604.4    1252.0
  i11    1952  3  1878.0   1140.4   207.8    610.5    1276.1
  i12    1952  4  1940.2   1180.5   223.3    620.8    1300.5
  i13    1953  1  1976.0   1194.9   227.5    641.2    1317.5
  i14    1953  2  1992.2   1202.5   228.5    655.9    1336.3
```

i15	1953	3	1979.5	1199.8	222.8	647.6	1330.2
i16	1953	4	1947.8	1191.8	205.0	645.4	1325.9
i17	1954	1	1938.1	1196.2	203.4	627.1	1330.3
i18	1954	2	1941.0	1211.3	203.0	606.1	1327.9
i19	1954	3	1962.0	1227.3	213.3	591.2	1344.2
i20	1954	4	2000.9	1252.6	223.3	587.4	1373.6
i21	1955	1	2058.1	1280.1	247.2	586.4	1392.7
i22	1955	2	2091.0	1304.3	262.8	579.9	1423.3
i23	1955	3	2118.9	1320.3	266.4	584.0	1451.1
i24	1955	4	2130.1	1336.7	272.0	571.3	1468.1
i25	1956	1	2121.0	1339.2	262.9	570.9	1480.9
i26	1956	2	2137.7	1343.7	260.0	582.6	1497.8
i27	1956	3	2135.3	1346.8	257.1	577.3	1504.1
i28	1956	4	2170.4	1365.3	254.4	592.5	1526.5
i29	1957	1	2182.7	1374.2	250.0	604.0	1527.5
i30	1957	2	2177.7	1376.5	249.9	600.6	1538.6
i31	1957	3	2198.9	1387.7	255.6	605.5	1548.7
i32	1957	4	2176.0	1388.8	234.1	616.6	1543.1
i33	1958	1	2117.4	1370.1	216.7	609.6	1524.7
i34	1958	2	2129.7	1380.9	211.3	625.0	1534.1
i35	1958	3	2177.5	1402.3	228.4	628.4	1568.1
i36	1958	4	2226.5	1418.8	249.6	641.5	1588.0
i37	1959	1	2273.0	1445.2	263.0	651.5	1599.5
i38	1959	2	2332.4	1468.2	286.2	663.9	1629.6
i39	1959	3	2331.4	1483.8	266.6	668.1	1627.0
i40	1959	4	2339.1	1485.6	275.6	662.2	1639.2
i41	1960	1	2391.0	1499.2	305.3	648.8	1657.7
i42	1960	2	2379.2	1518.1	274.0	657.4	1666.5
i43	1960	3	2383.6	1512.1	272.4	665.9	1667.7
i44	1960	4	2352.9	1513.5	239.5	673.1	1667.2
i45	1961	1	2366.5	1512.8	245.0	680.4	1680.6
i46	1961	2	2410.8	1535.2	263.3	687.2	1705.4
i47	1961	3	2450.4	1542.9	285.5	694.0	1729.4
i48	1961	4	2500.4	1574.2	290.2	711.1	1764.4
i49	1962	1	2544.0	1590.6	307.3	723.4	1777.9
i50	1962	2	2571.5	1609.9	304.5	731.7	1799.3
i51	1962	3	2596.8	1622.9	310.0	740.8	1811.4
i52	1962	4	2603.3	1645.9	299.5	744.2	1825.5
i53	1963	1	2634.1	1657.1	315.4	740.0	1838.9
i54	1963	2	2668.4	1673.0	320.8	744.3	1857.2
i55	1963	3	2719.6	1695.7	331.5	765.9	1879.2
i56	1963	4	2739.4	1710.0	335.2	759.2	1910.5
i57	1964	1	2800.5	1743.8	348.9	763.1	1947.6
i58	1964	2	2833.8	1775.0	347.5	772.9	1999.4
i59	1964	3	2872.0	1807.8	355.7	766.4	2027.8
i60	1964	4	2879.5	1812.8	358.3	766.1	2052.6
i61	1965	1	2950.1	1852.5	394.9	765.5	2071.8
i62	1965	2	2989.9	1873.2	394.6	781.3	2096.4
i63	1965	3	3050.7	1905.3	408.4	800.3	2155.3
i64	1965	4	3123.6	1959.3	410.1	817.2	2200.4
i65	1966	1	3201.1	1988.6	444.1	832.5	2219.3
i66	1966	2	3213.2	1994.0	436.5	857.8	2224.6
i67	1966	3	3233.6	2016.6	432.7	870.1	2254.0
i68	1966	4	3261.8	2025.1	435.8	888.0	2280.5
i69	1967	1	3291.8	2037.3	424.9	925.6	2312.6
i70	1967	2	3289.7	2064.6	405.0	921.3	2329.9
i71	1967	3	3313.5	2075.2	415.2	926.8	2351.4
i72	1967	4	3338.3	2087.9	423.6	934.8	2367.9
i73	1968	1	3406.2	2136.2	433.8	951.4	2409.5
i74	1968	2	3464.8	2169.6	451.8	956.0	2451.2
i75	1968	3	3489.2	2210.7	437.3	958.3	2457.9
i76	1968	4	3504.1	2220.4	442.2	960.5	2474.3
i77	1969	1	3558.3	2244.8	470.8	956.9	2477.5
i78	1969	2	3567.6	2258.8	467.1	956.0	2501.5
i79	1969	3	3588.3	2269.0	477.2	954.1	2550.2
i80	1969	4	3571.4	2286.5	452.6	943.1	2568.1
i81	1970	1	3566.5	2300.8	438.0	936.2	2581.9
i82	1970	2	3573.9	2312.0	439.4	927.3	2626.0
i83	1970	3	3605.2	2332.2	446.5	930.9	2661.1
i84	1970	4	3566.5	2324.9	421.0	929.9	2650.9
i85	1971	1	3666.1	2369.8	475.9	918.6	2703.5
i86	1971	2	3686.2	2391.4	490.2	915.2	2742.6

```
i87    1971  3  3714.5   2409.8   496.5    911.9    2752.9
i88    1971  4  3723.8   2449.8   480.6    909.4    2782.1
i89    1972  1  3796.9   2482.2   513.6    920.8    2797.6
i90    1972  2  3883.8   2527.5   544.9    921.9    2822.9
i91    1972  3  3922.3   2565.9   554.1    907.6    2883.6
i92    1972  4  3990.5   2626.3   559.4    909.1    2993.0
i93    1973  1  4092.3   2674.2   595.2    914.5    3031.9
i94    1973  2  4133.3   2671.4   618.2    911.5    3059.6
i95    1973  3  4117.0   2682.5   597.5    898.5    3079.3
i96    1973  4  4151.1   2675.6   615.3    908.4    3118.3
i97    1974  1  4119.3   2652.4   579.2    920.0    3072.1
i98    1974  2  4130.4   2662.0   577.3    927.8    3045.5
i99    1974  3  4084.5   2672.2   543.4    924.2    3053.3
i100   1974  4  4062.0   2628.4   547.0    927.4    3036.7
i101   1975  1  4010.0   2648.8   450.8    940.8    3015.0
i102   1975  2  4045.2   2695.4   436.4    938.3    3156.6
i103   1975  3  4115.4   2734.7   474.9    941.8    3114.9
i104   1975  4  4167.2   2764.6   486.8    949.1    3147.6
i105   1976  1  4266.1   2824.7   535.1    952.5    3201.9
i106   1976  2  4301.5   2850.9   559.8    943.3    3229.0
i107   1976  3  4321.9   2880.3   561.1    938.9    3259.7
i108   1976  4  4357.4   2919.6   565.9    938.6    3283.5
i109   1977  1  4410.5   2954.7   595.5    945.3    3305.4
i110   1977  2  4489.8   2970.5   635.0    955.1    3326.8
i111   1977  3  4570.6   2999.1   670.7    956.0    3376.5
i112   1977  4  4576.1   3044.0   656.4    954.5    3433.8
i113   1978  1  4588.9   3060.8   667.2    956.7    3466.3
i114   1978  2  4765.7   3127.0   709.7    982.1    3513.0
i115   1978  3  4811.7   3143.1   728.8    990.3    3548.1
i116   1978  4  4876.0   3167.8   746.3    999.6    3582.6
i117   1979  1  4888.3   3188.6   746.0    990.6    3620.7
i118   1979  2  4891.4   3184.3   745.7   1000.5    3607.1
i119   1979  3  4926.2   3213.9   732.1   1002.4    3628.8
i120   1979  4  4942.6   3225.7   717.8   1010.8    3657.8
i121   1980  1  4958.9   3222.4   711.7   1025.6    3678.5
i122   1980  2  4857.8   3149.2   647.4   1028.7    3612.2
i123   1980  3  4850.3   3181.2   599.8   1015.4    3637.6
i124   1980  4  4936.6   3219.4   662.2   1013.9    3703.8
i125   1981  1  5032.5   3233.1   726.3   1027.5    3713.5
i126   1981  2  4997.3   3235.5   693.4   1030.1    3696.6
i127   1981  3  5056.8   3250.5   733.9   1027.8    3777.0
i128   1981  4  4997.1   3225.0   708.8   1034.8    3777.2
i129   1982  1  4914.3   3244.3   634.8   1033.6    3769.4
i130   1982  2  4935.5   3253.4   631.6   1039.5    3791.4
i131   1982  3  4912.1   3274.6   623.5   1046.8    3799.4
i132   1982  4  4915.6   3329.6   571.1   1064.0    3806.4
i133   1983  1  4972.4   3360.1   590.7   1069.8    3831.2
i134   1983  2  5089.8   3430.1   650.7   1078.2    3857.8
i135   1983  3  5180.4   3484.7   691.4   1097.0    3928.6
i136   1983  4  5286.8   3542.2   762.2   1078.8    4010.2
i137   1984  1  5402.3   3579.7   845.0   1091.0    4103.0
i138   1984  2  5493.8   3628.3   873.2   1115.2    4182.4
i139   1984  3  5541.3   3653.5   890.7   1123.1    4258.8
i140   1984  4  5583.1   3700.9   876.9   1144.2    4286.1
i141   1985  1  5629.7   3756.8   848.9   1157.6    4287.6
i142   1985  2  5673.8   3791.5   862.8   1180.5    4368.7
i143   1985  3  5758.6   3860.9   854.1   1209.2    4346.6
i144   1985  4  5806.0   3874.2   887.8   1214.7    4388.3
i145   1986  1  5858.9   3907.9   886.2   1224.0    4444.5
i146   1986  2  5883.3   3950.4   868.3   1248.0    4489.3
i147   1986  3  5937.9   4019.7   838.0   1277.4    4507.9
i148   1986  4  5969.5   4046.8   838.2   1271.5    4504.5
i149   1987  1  6013.3   4049.7   863.4   1278.4    4556.9
i150   1987  2  6077.2   4101.5   863.9   1289.1    4512.7
i151   1987  3  6128.1   4147.0   860.5   1292.4    4600.7
i152   1987  4  6234.4   4155.3   929.3   1310.0    4659.6
i153   1988  1  6275.9   4228.0   884.6   1300.1    4724.1
i154   1988  2  6349.8   4256.8   902.5   1302.4    4758.9
i155   1988  3  6382.3   4291.6   907.5   1300.3    4801.9
i156   1988  4  6465.2   4341.4   916.7   1327.2    4851.4
i157   1989  1  6543.8   4357.1   952.7   1319.3    4903.5
i158   1989  2  6579.4   4374.8   941.1   1340.6    4891.0
```

```
  i159   1989  3  6610.6   4413.4   929.3   1353.5    4902.7
  i160   1989  4  6633.5   4429.4   922.9   1360.4    4928.8
  i161   1990  1  6716.3   4466.0   934.0   1381.2    5001.6
  i162   1990  2  6731.7   4478.8   933.0   1384.7    5026.6
  i163   1990  3  6719.4   4495.6   912.6   1384.8    5032.7
  i164   1990  4  6664.2   4457.7   849.6   1398.6    4995.8
  i165   1991  1  6631.4   4437.5   815.1   1404.7    4999.5
  i166   1991  2  6668.5   4469.9   808.8   1408.9    5033.3
  i167   1991  3  6684.9   4484.3   829.8   1403.0    5045.4
  i168   1991  4  6720.9   4474.8   864.2   1397.0    5053.8
  i169   1992  1  6783.3   4544.8   843.8   1407.6    5138.8
  i170   1992  2  6846.8   4566.7   901.8   1405.7    5172.5
  i171   1992  3  6899.7   4600.5   912.1   1413.1    5174.2
  i172   1992  4  6990.6   4665.9   941.6   1413.7    5271.5
  i173   1993  1  6988.7   4674.9   964.8   1396.4    5181.2
  i174   1993  2  7031.2   4721.5   967.0   1398.0    5258.6
  i175   1993  3  7062.0   4776.9   964.1   1398.4    5266.8
  i176   1993  4  7168.7   4822.3  1015.6   1402.2    5338.5
  i177   1994  1  7229.4   4866.6  1057.3   1388.0    5293.2
  i178   1994  2  7330.2   4907.9  1118.5   1390.4    5381.2
  i179   1994  3  7370.2   4944.5  1101.8   1417.5    5420.9
  i180   1994  4  7461.1   4993.6  1150.5   1404.5    5493.4
  i181   1995  1  7488.7   5011.6  1162.4   1407.3    5515.4
  i182   1995  2  7503.3   5059.6  1128.5   1414.0    5509.0
  i183   1995  3  7561.4   5099.2  1119.1   1410.8    5546.6
  i184   1995  4  7621.9   5132.1  1152.4   1393.5    5585.3
  i185   1996  1  7676.4   5174.3  1172.3   1404.8    5622.0
  i186   1996  2  7802.9   5229.5  1233.4   1430.4    5649.4
  i187   1996  3  7841.9   5254.3  1281.4   1422.0    5709.7
  i188   1996  4  7931.3   5291.9  1283.7   1430.6    5729.9
  i189   1997  1  8016.4   5350.7  1325.4   1434.6    5771.8
  i190   1997  2  8131.9   5375.7  1400.6   1457.0    5821.2
  i191   1997  3  8216.6   5462.1  1408.6   1464.8    5877.3
  i192   1997  4  8272.9   5507.1  1438.5   1465.3    5947.5
  i193   1998  1  8396.3   5576.3  1543.3   1456.1    6064.5
  i194   1998  2  8442.9   5660.2  1516.8   1482.6    6153.6
  i195   1998  3  8528.5   5713.7  1559.7   1489.9    6209.9
  i196   1998  4  8667.9   5784.7  1612.1   1504.8    6246.6
  i197   1999  1  8733.5   5854.0  1641.8   1512.3    6268.2
  i198   1999  2  8771.2   5936.1  1617.4   1516.8    6300.0
  i199   1999  3  8871.5   6000.0  1655.8   1533.2    6332.4
  i200   1999  4  9049.9   6083.6  1725.4   1564.8    6379.2
  i201   2000  1  9102.5   6171.7  1722.9   1560.4    6431.6
  i202   2000  2  9229.4   6226.3  1801.6   1577.2    6523.7
  i203   2000  3  9260.1   6292.1  1788.8   1570.0    6566.5
  i204   2000  4  9303.9   6341.1  1778.3   1582.8    6634.9

;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------

variables
   alpha  'parameter to estimate'
   beta   'parameter to estimate'
   gamma  'parameter to estimate'
   sse    'sum of squared errors'
;

equations
    obj        'dummy obj'
    linfit(i)  'linear fit'
    nlfit(i)   'non-linear fit'
;


obj..       sse =n= 0;
linfit(i).. data(i,'realcons') =e= alpha + beta*data(i,'realdpi');
nlfit(i)..  data(i,'realcons') =e= alpha + beta*data(i,'realdpi')**gamma;
```

```
model linear     /obj,linfit/;
model nonlinear  /obj,nlfit/;


*-----------------------------------------------------------------------------
* initial values for alpha and beta by OLS
*-----------------------------------------------------------------------------

option lp=ls;
solve linear using lp minimizing sse;

*-----------------------------------------------------------------------------
* nonlinear fit
*-----------------------------------------------------------------------------

gamma.l = 1;
option nlp=nls;
solve nonlinear using nlp minimizing sse;
```

12.3. **Misra1.** A set of nonlinear regression problems on the same data set from the NIST collection[23, 28]. The different functional forms that are fitted are:

$$y = \begin{cases} \beta_1(1-\exp(-\beta_2 x)) + \varepsilon \\ \beta_1(1-(1+\frac{\beta_2 * x}{2})^{-2}) + \varepsilon \\ \beta_1(1-\frac{1}{\sqrt{1+2\beta_2 x}}) + \varepsilon \\ \frac{\beta_1\beta_2 x}{1+\beta_2 x} + \varepsilon \end{cases} \tag{36}$$

12.3.1. *Misra1a.* First nonlinear regression model from the NIST data set. The model to be fitted is:

$$y = \beta_1\left[1-\exp(-\beta_2 x)\right] + \varepsilon \tag{37}$$

12.3.2. *Model Misra1a.gms.* [5]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------

Procedure:      Nonlinear Least Squares Regression

Description:    These data are the result of a NIST study regarding
                dental research in monomolecular adsorption.  The
                response variable is volume, and the predictor
                variable is pressure.

Reference:      Misra, D., NIST (1978).
                Dental Research Monomolecular Adsorption Study.

Data:           1 Response Variable  (y = volume)
                1 Predictor Variable (x = pressure)
                14 Observations
                Lower Level of Difficulty
                Observed Data

Model:          Exponential Class
                2 Parameters (b1 and b2)
```

[5] www.amsterdamoptimization.com/models/regression/Misra1a.gms

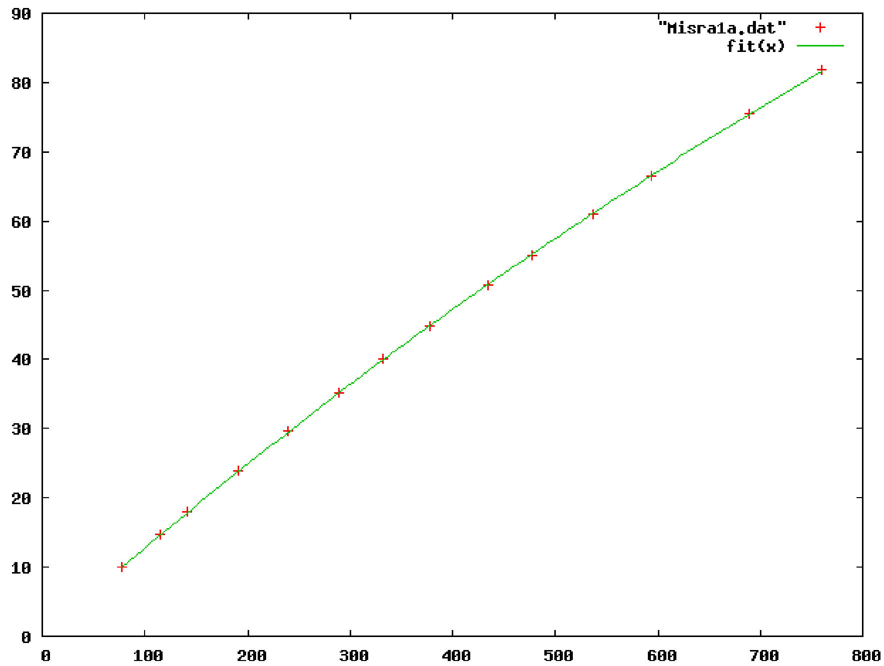

FIGURE 8. Scatter plot of model Misra1a

```
                y = b1*(1-exp[-b2*x])  +  e

           Starting values                  Certified Values

         Start 1      Start 2          Parameter     Standard Deviation
  b1 =   500          250           2.3894212918E+02  2.7070075241E+00
  b2 =     0.0001       0.0005      5.5015643181E-04  7.2668688436E-06

Residual Sum of Squares:                    1.2455138894E-01
Residual Standard Deviation:                1.0187876330E-01
Degrees of Freedom:                                12
Number of Observations:                            14


$offtext


*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i14/;

table data(i,*)
             y            x
  i1      10.07E0      77.6E0
  i2      14.73E0     114.9E0
  i3      17.94E0     141.1E0
  i4      23.93E0     190.8E0
  i5      29.61E0     239.9E0
  i6      35.18E0     289.0E0
  i7      40.02E0     332.8E0
  i8      44.82E0     378.4E0
  i9      50.76E0     434.8E0
```

```
  i10    55.05E0     477.3E0
  i11    61.01E0     536.8E0
  i12    66.40E0     593.1E0
  i13    75.47E0     689.1E0
  i14    81.78E0     760.0E0
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /2.3894212918E+02/
  cb2 'certified value for b2' /5.5015643181E-04/
  ce1 'certified std err for b1 ' / 2.7070075241E+00 /
  ce2 'certified std err for b2 ' / 7.2668688436E-06 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1*(1-exp[-b2*x(i)]);

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l = 500;
b2.l = 0.0001;

option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l = 250;
b2.l = 0.0005;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";
```

Here is the R output for comparison:

```
> library(NISTnls)
> data(Misra1a)
> fm1 <- nls(y ~ b1*(1-exp(-b2*x)), data = Misra1a, trace = TRUE,
+             start = c(b1 = 500, b2 = 0.0001) )
10780.19 :  5e+02 1e-04
10697.62 :  4.666633e+02 1.079252e-04
10640.08 :  4.112467e+02 1.234257e-04
10497.53 :  3.716416e+02 1.383456e-04
10374.65 :  3.127817e+02 1.669193e-04
10230.62 :  2.430935e+02 2.198045e-04
9280.133 :  1.868551e+02 3.105043e-04
5770.421 :  1.705528e+02 4.354661e-04
1242.317 :  1.979720e+02 5.328954e-04
1.137841 :  238.75381222   0.00055422
0.1245559 :  2.389272e+02 5.501891e-04
0.1245514 :  2.389421e+02 5.501565e-04
> summary(fm1)

Formula: y ~ b1 * (1 - exp(-b2 * x))

Parameters:
    Estimate Std. Error t value Pr(>|t|)
b1 2.389e+02  2.707e+00   88.27   <2e-16 ***
b2 5.502e-04  7.267e-06   75.71   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1019 on 12 degrees of freedom

Number of iterations to convergence: 11
Achieved convergence tolerance: 4.009e-06

>
```

12.3.3. *Misra1b.* Same data but different functional form as Misra1a. The model to be fitted is:

$$y = \beta_1 \left[1 - \left(1 + \frac{\beta_2 * x}{2} \right)^{-2} \right] + \varepsilon \tag{38}$$

12.3.4. *Model Misra1b.gms.* [6]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   These data are the result of a NIST study regarding
               dental research in monomolecular adsorption.  The
               response variable is volume, and the predictor
               variable is pressure.

Reference:     Misra, D., NIST (1978).
               Dental Research Monomolecular Adsorption Study.

Data:          1 Response  (y = volume)
               1 Predictor (x = pressure)
               14 Observations
               Lower Level of Difficulty
```

[6] www.amsterdamoptimization.com/models/regression/Misra1b.gms

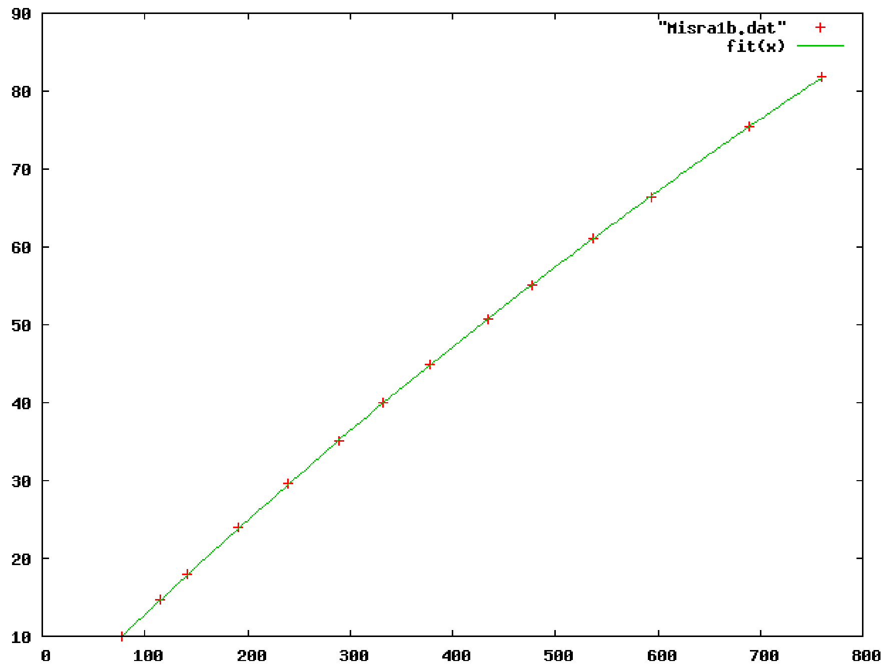

FIGURE 9. Scatter plot of model Misra1b

```
                Observed Data

Model:          Miscellaneous Class
                2 Parameters (b1 and b2)

                y = b1 * (1-(1+b2*x/2)**(-2))  +  e

           Starting values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   500         300           3.3799746163E+02  3.1643950207E+00
  b2 =     0.0001      0.0002      3.9039091287E-04  4.2547321834E-06

Residual Sum of Squares:                    7.5464681533E-02
Residual Standard Deviation:                7.9301471998E-02
Degrees of Freedom:                                12
Number of Observations:                            14


$offtext


*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i14/;

table data(i,*)
             y            x
  i1     10.07E0      77.6E0
  i2     14.73E0     114.9E0
  i3     17.94E0     141.1E0
  i4     23.93E0     190.8E0
  i5     29.61E0     239.9E0
```

```
  i6     35.18E0     289.0E0
  i7     40.02E0     332.8E0
  i8     44.82E0     378.4E0
  i9     50.76E0     434.8E0
  i10    55.05E0     477.3E0
  i11    61.01E0     536.8E0
  i12    66.40E0     593.1E0
  i13    75.47E0     689.1E0
  i14    81.78E0     760.0E0
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /3.3799746163E+02/
  cb2 'certified value for b2' /3.9039091287E-04/
  ce1 'certified std err for b1 ' / 3.1643950207E+00 /
  ce2 'certified std err for b2 ' / 4.2547321834E-06 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1 * (1-(1+b2*x(i)/2)**(-2));

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l = 500;
b2.l = 0.0001;

option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l = 300;
b2.l = 0.0002;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;
```

```
abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* third set of initial values
*-----------------------------------------------------------------------------

b1.l = 3.3799746163E+02;
b2.l = 3.9039091287E-04;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";
```

12.3.5. *Misra1c.* Same data but different functional form as Misra1a and Misra1b. The model to be fitted is:

$$y = \beta_1 \left[1 - \frac{1}{\sqrt{1 + 2\beta_2 x}} \right] + \varepsilon \quad (39)$$

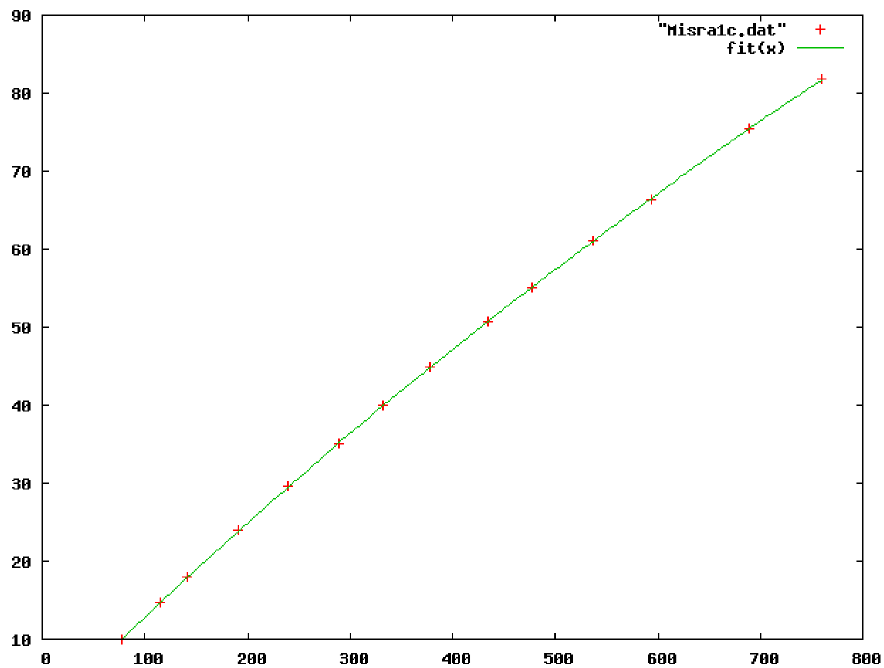

FIGURE 10. Scatter plot of model Misra1c

12.3.6. *Model Misra1c.gms.* [7]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007
```

[7] www.amsterdamoptimization.com/models/regression/Misra1c.gms

```
  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   These data are the result of a NIST study regarding
               dental research in monomolecular adsorption.  The
               response variable is volume, and the predictor
               variable is pressure.

Reference:     Misra, D., NIST (1978).
               Dental Research Monomolecular Adsorption.

Data:          1 Response  (y = volume)
               1 Predictor (x = pressure)
               14 Observations
               Average Level of Difficulty
               Observed Data

Model:         Miscellaneous Class
               2 Parameters (b1 and b2)

               y = b1 * (1-(1+2*b2*x)**(-.5))  +  e

          Starting values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   500         600           6.3642725809E+02  4.6638326572E+00
  b2 =     0.0001      0.0002      2.0813627256E-04  1.7728423155E-06

Residual Sum of Squares:                    4.0966836971E-02
Residual Standard Deviation:                5.8428615257E-02
Degrees of Freedom:                                12
Number of Observations:                            14

$offtext


*------------------------------------------------------------------------------
* data
*------------------------------------------------------------------------------

set i /i1*i14/;

table data(i,*)
            y           x
  i1     10.07E0      77.6E0
  i2     14.73E0     114.9E0
  i3     17.94E0     141.1E0
  i4     23.93E0     190.8E0
  i5     29.61E0     239.9E0
  i6     35.18E0     289.0E0
  i7     40.02E0     332.8E0
  i8     44.82E0     378.4E0
  i9     50.76E0     434.8E0
  i10    55.05E0     477.3E0
  i11    61.01E0     536.8E0
  i12    66.40E0     593.1E0
  i13    75.47E0     689.1E0
  i14    81.78E0     760.0E0
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
```

```
* certified values
*
scalars
  cb1 'certified value for b1' /6.3642725809E+02/
  cb2 'certified value for b2' /2.0813627256E-04/
  ce1 'certified std err for b1 ' / 4.6638326572E+00 /
  ce2 'certified std err for b2 ' / 1.7728423155E-06 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1*(1-(1+2*b2*x(i))**(-.5));

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l = 500;
b2.l = 0.0001;

option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l = 600;
b2.l = 0.0002;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";
```

12.3.7. *Misra1d.* Same data but different functional form as Misra1a, Misra1b and Misra1c. The model to be fitted is:

$$y = \frac{\beta_1 \beta_2 x}{1 + \beta_2 x} + \varepsilon \tag{40}$$

12.3.8. *Model Misra1d.gms.* [8]

```
$ontext

  Nonlinear Least Squares Regression example
```

[8] www.amsterdamoptimization.com/models/regression/Misra1d.gms

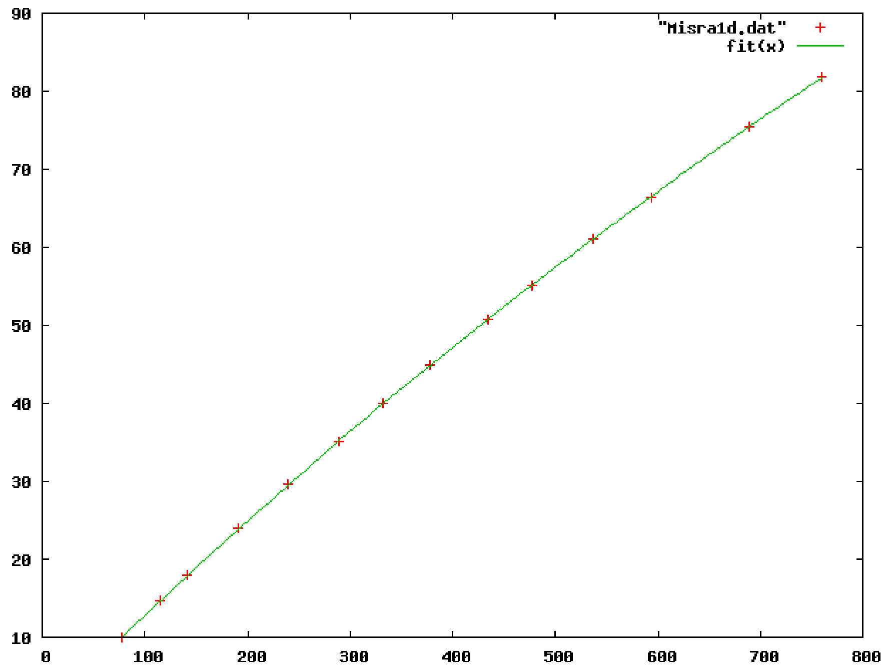

FIGURE 11. Scatter plot of model Misra1d

```
  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------

Procedure:     Nonlinear Least Squares Regression

Description:   These data are the result of a NIST study regarding
               dental research in monomolecular adsorption.  The
               response variable is volume, and the predictor
               variable is pressure.

Reference:     Misra, D., NIST (1978).
               Dental Research Monomolecular Adsorption Study.

Data:          1 Response  (y = volume)
               1 Predictor (x = pressure)
               14 Observations
               Average Level of Difficulty
               Observed Data

Model:         Miscellaneous Class
               2 Parameters (b1 and b2)

               y = b1*b2*x*((1+b2*x)**(-1))  +  e

          Starting values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   500         450           4.3736970754E+02  3.6489174345E+00
  b2 =     0.0001      0.0003      3.0227324449E-04  2.9334354479E-06

Residual Sum of Squares:                    5.6419295283E-02
```

```
Residual Standard Deviation:                6.8568272111E-02
Degrees of Freedom:                                12
Number of Observations:                            14

$offtext


*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i14/;

table data(i,*)
             y           x
  i1     10.07E0      77.6E0
  i2     14.73E0     114.9E0
  i3     17.94E0     141.1E0
  i4     23.93E0     190.8E0
  i5     29.61E0     239.9E0
  i6     35.18E0     289.0E0
  i7     40.02E0     332.8E0
  i8     44.82E0     378.4E0
  i9     50.76E0     434.8E0
  i10    55.05E0     477.3E0
  i11    61.01E0     536.8E0
  i12    66.40E0     593.1E0
  i13    75.47E0     689.1E0
  i14    81.78E0     760.0E0
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /4.3736970754E+02/
  cb2 'certified value for b2' /3.0227324449E-04/
  ce1 'certified std err for b1 ' / 3.6489174345E+00 /
  ce2 'certified std err for b2 ' / 2.9334354479E-06 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1*b2*x(i)*((1+b2*x(i))**(-1));

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l = 500;
```

```
b2.l = 0.0001;

option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l = 450;
b2.l = 0.0003;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";
```

12.4. **Chwirut.** These model from the NIST data set fits the equation:

$$y = \frac{\exp(-\beta_1 x)}{\beta_2 + \beta_3 x} + \varepsilon \tag{41}$$

The second model has fewer data points.

12.4.1. *Model Chwirut1.gms.* [9]

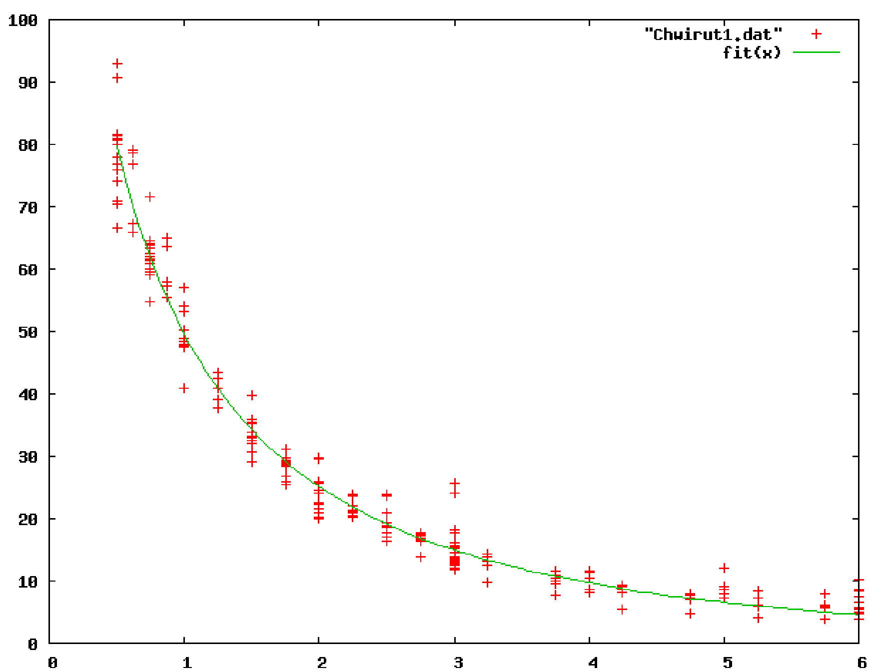

FIGURE 12. Scatter plot of model Chwirut1

[9] www.amsterdamoptimization.com/models/regression/Chwirut1.gms

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------

Procedure:     Nonlinear Least Squares Regression

Description:   These data are the result of a NIST study involving
               ultrasonic calibration.  The response variable is
               ultrasonic response, and the predictor variable is
               metal distance.

Reference:     Chwirut, D., NIST (197?).
               Ultrasonic Reference Block Study.

Data:          1 Response Variable  (y = ultrasonic response)
               1 Predictor Variable (x = metal distance)
               214 Observations
               Lower Level of Difficulty
               Observed Data

Model:         Exponential Class
               3 Parameters (b1 to b3)

               y = exp[-b1*x]/(b2+b3*x)  +  e

          Starting values                 Certified Values

        Start 1     Start 2          Parameter     Standard Deviation
  b1 =   0.1         0.15         1.9027818370E-01  2.1938557035E-02
  b2 =   0.01        0.008        6.1314004477E-03  3.4500025051E-04
  b3 =   0.02        0.010        1.0530908399E-02  7.9281847748E-04

Residual Sum of Squares:                    2.3844771393E+03
Residual Standard Deviation:                3.3616721320E+00
Degrees of Freedom:                               211
Number of Observations:                           214

$offtext

*-------------------------------------------------------------------------------
* data
*-------------------------------------------------------------------------------

set i /i1*i214/;

table data(i,*)
              y            x
  i1     92.9000E0     0.5000E0
  i2     78.7000E0     0.6250E0
  i3     64.2000E0     0.7500E0
  i4     64.9000E0     0.8750E0
  i5     57.1000E0     1.0000E0
  i6     43.3000E0     1.2500E0
  i7     31.1000E0     1.7500E0
  i8     23.6000E0     2.2500E0
  i9     31.0500E0     1.7500E0
  i10    23.7750E0     2.2500E0
  i11    17.7375E0     2.7500E0
  i12    13.8000E0     3.2500E0
  i13    11.5875E0     3.7500E0
  i14     9.4125E0     4.2500E0
  i15     7.7250E0     4.7500E0
  i16     7.3500E0     5.2500E0
  i17     8.0250E0     5.7500E0
```

```
 i18   90.6000E0    0.5000E0
 i19   76.9000E0    0.6250E0
 i20   71.6000E0    0.7500E0
 i21   63.6000E0    0.8750E0
 i22   54.0000E0    1.0000E0
 i23   39.2000E0    1.2500E0
 i24   29.3000E0    1.7500E0
 i25   21.4000E0    2.2500E0
 i26   29.1750E0    1.7500E0
 i27   22.1250E0    2.2500E0
 i28   17.5125E0    2.7500E0
 i29   14.2500E0    3.2500E0
 i30    9.4500E0    3.7500E0
 i31    9.1500E0    4.2500E0
 i32    7.9125E0    4.7500E0
 i33    8.4750E0    5.2500E0
 i34    6.1125E0    5.7500E0
 i35   80.0000E0    0.5000E0
 i36   79.0000E0    0.6250E0
 i37   63.8000E0    0.7500E0
 i38   57.2000E0    0.8750E0
 i39   53.2000E0    1.0000E0
 i40   42.5000E0    1.2500E0
 i41   26.8000E0    1.7500E0
 i42   20.4000E0    2.2500E0
 i43   26.8500E0    1.7500E0
 i44   21.0000E0    2.2500E0
 i45   16.4625E0    2.7500E0
 i46   12.5250E0    3.2500E0
 i47   10.5375E0    3.7500E0
 i48    8.5875E0    4.2500E0
 i49    7.1250E0    4.7500E0
 i50    6.1125E0    5.2500E0
 i51    5.9625E0    5.7500E0
 i52   74.1000E0    0.5000E0
 i53   67.3000E0    0.6250E0
 i54   60.8000E0    0.7500E0
 i55   55.5000E0    0.8750E0
 i56   50.3000E0    1.0000E0
 i57   41.0000E0    1.2500E0
 i58   29.4000E0    1.7500E0
 i59   20.4000E0    2.2500E0
 i60   29.3625E0    1.7500E0
 i61   21.1500E0    2.2500E0
 i62   16.7625E0    2.7500E0
 i63   13.2000E0    3.2500E0
 i64   10.8750E0    3.7500E0
 i65    8.1750E0    4.2500E0
 i66    7.3500E0    4.7500E0
 i67    5.9625E0    5.2500E0
 i68    5.6250E0    5.7500E0
 i69   81.5000E0     .5000E0
 i70   62.4000E0     .7500E0
 i71   32.5000E0    1.5000E0
 i72   12.4100E0    3.0000E0
 i73   13.1200E0    3.0000E0
 i74   15.5600E0    3.0000E0
 i75    5.6300E0    6.0000E0
 i76   78.0000E0     .5000E0
 i77   59.9000E0     .7500E0
 i78   33.2000E0    1.5000E0
 i79   13.8400E0    3.0000E0
 i80   12.7500E0    3.0000E0
 i81   14.6200E0    3.0000E0
 i82    3.9400E0    6.0000E0
 i83   76.8000E0     .5000E0
 i84   61.0000E0     .7500E0
 i85   32.9000E0    1.5000E0
 i86   13.8700E0    3.0000E0
 i87   11.8100E0    3.0000E0
 i88   13.3100E0    3.0000E0
 i89    5.4400E0    6.0000E0
```

```
i90    78.0000E0      .5000E0
i91    63.5000E0      .7500E0
i92    33.8000E0     1.5000E0
i93    12.5600E0     3.0000E0
i94     5.6300E0     6.0000E0
i95    12.7500E0     3.0000E0
i96    13.1200E0     3.0000E0
i97     5.4400E0     6.0000E0
i98    76.8000E0      .5000E0
i99    60.0000E0      .7500E0
i100   47.8000E0     1.0000E0
i101   32.0000E0     1.5000E0
i102   22.2000E0     2.0000E0
i103   22.5700E0     2.0000E0
i104   18.8200E0     2.5000E0
i105   13.9500E0     3.0000E0
i106   11.2500E0     4.0000E0
i107    9.0000E0     5.0000E0
i108    6.6700E0     6.0000E0
i109   75.8000E0      .5000E0
i110   62.0000E0      .7500E0
i111   48.8000E0     1.0000E0
i112   35.2000E0     1.5000E0
i113   20.0000E0     2.0000E0
i114   20.3200E0     2.0000E0
i115   19.3100E0     2.5000E0
i116   12.7500E0     3.0000E0
i117   10.4200E0     4.0000E0
i118    7.3100E0     5.0000E0
i119    7.4200E0     6.0000E0
i120   70.5000E0      .5000E0
i121   59.5000E0      .7500E0
i122   48.5000E0     1.0000E0
i123   35.8000E0     1.5000E0
i124   21.0000E0     2.0000E0
i125   21.6700E0     2.0000E0
i126   21.0000E0     2.5000E0
i127   15.6400E0     3.0000E0
i128    8.1700E0     4.0000E0
i129    8.5500E0     5.0000E0
i130   10.1200E0     6.0000E0
i131   78.0000E0      .5000E0
i132   66.0000E0      .6250E0
i133   62.0000E0      .7500E0
i134   58.0000E0      .8750E0
i135   47.7000E0     1.0000E0
i136   37.8000E0     1.2500E0
i137   20.2000E0     2.2500E0
i138   21.0700E0     2.2500E0
i139   13.8700E0     2.7500E0
i140    9.6700E0     3.2500E0
i141    7.7600E0     3.7500E0
i142    5.4400E0     4.2500E0
i143    4.8700E0     4.7500E0
i144    4.0100E0     5.2500E0
i145    3.7500E0     5.7500E0
i146   24.1900E0     3.0000E0
i147   25.7600E0     3.0000E0
i148   18.0700E0     3.0000E0
i149   11.8100E0     3.0000E0
i150   12.0700E0     3.0000E0
i151   16.1200E0     3.0000E0
i152   70.8000E0      .5000E0
i153   54.7000E0      .7500E0
i154   48.0000E0     1.0000E0
i155   39.8000E0     1.5000E0
i156   29.8000E0     2.0000E0
i157   23.7000E0     2.5000E0
i158   29.6200E0     2.0000E0
i159   23.8100E0     2.5000E0
i160   17.7000E0     3.0000E0
i161   11.5500E0     4.0000E0
```

```
  i162   12.0700E0     5.0000E0
  i163    8.7400E0     6.0000E0
  i164   80.7000E0      .5000E0
  i165   61.3000E0      .7500E0
  i166   47.5000E0     1.0000E0
  i167   29.0000E0     1.5000E0
  i168   24.0000E0     2.0000E0
  i169   17.7000E0     2.5000E0
  i170   24.5600E0     2.0000E0
  i171   18.6700E0     2.5000E0
  i172   16.2400E0     3.0000E0
  i173    8.7400E0     4.0000E0
  i174    7.8700E0     5.0000E0
  i175    8.5100E0     6.0000E0
  i176   66.7000E0      .5000E0
  i177   59.2000E0      .7500E0
  i178   40.8000E0     1.0000E0
  i179   30.7000E0     1.5000E0
  i180   25.7000E0     2.0000E0
  i181   16.3000E0     2.5000E0
  i182   25.9900E0     2.0000E0
  i183   16.9500E0     2.5000E0
  i184   13.3500E0     3.0000E0
  i185    8.6200E0     4.0000E0
  i186    7.2000E0     5.0000E0
  i187    6.6400E0     6.0000E0
  i188   13.6900E0     3.0000E0
  i189   81.0000E0      .5000E0
  i190   64.5000E0      .7500E0
  i191   35.5000E0     1.5000E0
  i192   13.3100E0     3.0000E0
  i193    4.8700E0     6.0000E0
  i194   12.9400E0     3.0000E0
  i195    5.0600E0     6.0000E0
  i196   15.1900E0     3.0000E0
  i197   14.6200E0     3.0000E0
  i198   15.6400E0     3.0000E0
  i199   25.5000E0     1.7500E0
  i200   25.9500E0     1.7500E0
  i201   81.7000E0      .5000E0
  i202   61.6000E0      .7500E0
  i203   29.8000E0     1.7500E0
  i204   29.8100E0     1.7500E0
  i205   17.1700E0     2.7500E0
  i206   10.3900E0     3.7500E0
  i207   28.4000E0     1.7500E0
  i208   28.6900E0     1.7500E0
  i209   81.3000E0      .5000E0
  i210   60.9000E0      .7500E0
  i211   16.6500E0     2.7500E0
  i212   10.0500E0     3.7500E0
  i213   28.9000E0     1.7500E0
  i214   28.9500E0     1.7500E0
;


*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' / 1.9027818370E-01 /
  cb2 'certified value for b2' / 6.1314004477E-03 /
  cb3 'certified value for b3' / 1.0530908399E-02 /
  ce1 'certified std err for b1 ' / 2.1938557035E-02 /
  ce2 'certified std err for b2 ' / 3.4500025051E-04 /
```

```
  ce3 'certified std err for b3 ' / 7.9281847748E-04 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= exp(-b1*x(i))/(b2+b3*x(i));

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l = 0.1;
b2.l = 0.01;
b3.l = 0.02;

option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l = 0.15;
b2.l = 0.008;
b3.l = 0.010;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";
```

12.4.2. *Model Chwirut2.gms.* [10]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------


Procedure:     Nonlinear Least Squares Regression
```

[10] www.amsterdamoptimization.com/models/regression/Chwirut2.gms

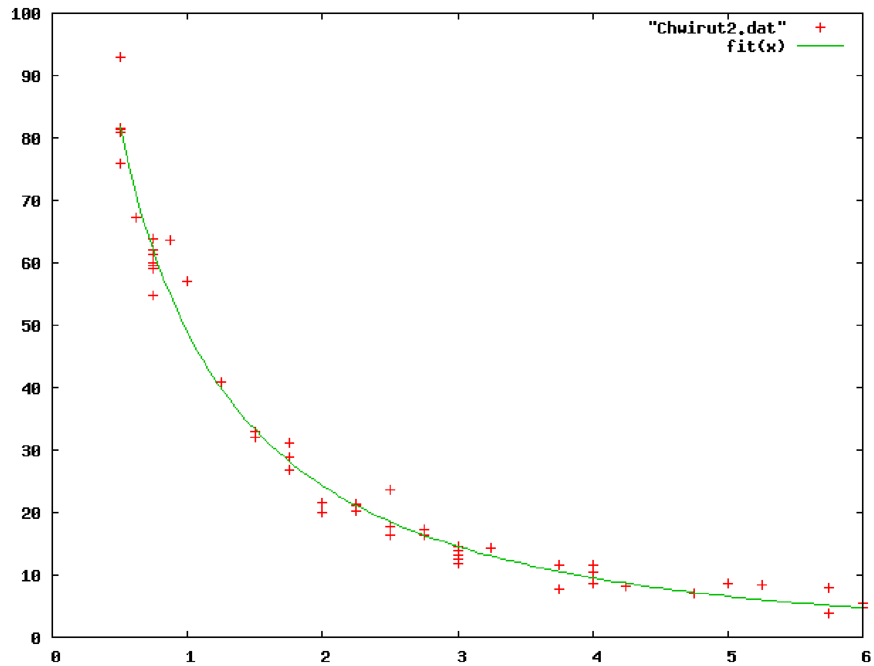

FIGURE 13. Scatter plot of model Chwirut2

```
Description:   These data are the result of a NIST study involving
               ultrasonic calibration.  The response variable is
               ultrasonic response, and the predictor variable is
               metal distance.

Reference:     Chwirut, D., NIST (197?).
               Ultrasonic Reference Block Study.

Data:          1 Response   (y = ultrasonic response)
               1 Predictor  (x = metal distance)
               54 Observations
               Lower Level of Difficulty
               Observed Data

Model:         Exponential Class
               3 Parameters (b1 to b3)

               y = exp(-b1*x)/(b2+b3*x)   +  e

          Starting values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   0.1         0.15          1.6657666537E-01  3.8303286810E-02
  b2 =   0.01        0.008         5.1653291286E-03  6.6621605126E-04
  b3 =   0.02        0.010         1.2150007096E-02  1.5304234767E-03

Residual Sum of Squares:                    5.1304802941E+02
Residual Standard Deviation:                3.1717133040E+00
Degrees of Freedom:                                51
Number of Observations:                            54

$offtext

*-----------------------------------------------------------------------------
* data
```

```
*-----------------------------------------------------------------------------

set i /i1*i54/;

table data(i,*)
              y          x
  i1     92.9000E0     0.500E0
  i2     57.1000E0     1.000E0
  i3     31.0500E0     1.750E0
  i4     11.5875E0     3.750E0
  i5      8.0250E0     5.750E0
  i6     63.6000E0     0.875E0
  i7     21.4000E0     2.250E0
  i8     14.2500E0     3.250E0
  i9      8.4750E0     5.250E0
  i10    63.8000E0     0.750E0
  i11    26.8000E0     1.750E0
  i12    16.4625E0     2.750E0
  i13     7.1250E0     4.750E0
  i14    67.3000E0     0.625E0
  i15    41.0000E0     1.250E0
  i16    21.1500E0     2.250E0
  i17     8.1750E0     4.250E0
  i18    81.5000E0      .500E0
  i19    13.1200E0     3.000E0
  i20    59.9000E0      .750E0
  i21    14.6200E0     3.000E0
  i22    32.9000E0     1.500E0
  i23     5.4400E0     6.000E0
  i24    12.5600E0     3.000E0
  i25     5.4400E0     6.000E0
  i26    32.0000E0     1.500E0
  i27    13.9500E0     3.000E0
  i28    75.8000E0      .500E0
  i29    20.0000E0     2.000E0
  i30    10.4200E0     4.000E0
  i31    59.5000E0      .750E0
  i32    21.6700E0     2.000E0
  i33     8.5500E0     5.000E0
  i34    62.0000E0      .750E0
  i35    20.2000E0     2.250E0
  i36     7.7600E0     3.750E0
  i37     3.7500E0     5.750E0
  i38    11.8100E0     3.000E0
  i39    54.7000E0      .750E0
  i40    23.7000E0     2.500E0
  i41    11.5500E0     4.000E0
  i42    61.3000E0      .750E0
  i43    17.7000E0     2.500E0
  i44     8.7400E0     4.000E0
  i45    59.2000E0      .750E0
  i46    16.3000E0     2.500E0
  i47     8.6200E0     4.000E0
  i48    81.0000E0      .500E0
  i49     4.8700E0     6.000E0
  i50    14.6200E0     3.000E0
  i51    81.7000E0      .500E0
  i52    17.1700E0     2.750E0
  i53    81.3000E0      .500E0
  i54    28.9000E0     1.750E0
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
```

```
scalars
  cb1 'certified value for b1' / 1.6657666537E-01 /
  cb2 'certified value for b2' / 5.1653291286E-03 /
  cb3 'certified value for b3' / 1.2150007096E-02 /
  ce1 'certified std err for b1 ' / 3.8303286810E-02 /
  ce2 'certified std err for b2 ' / 6.6621605126E-04 /
  ce3 'certified std err for b3 ' / 1.5304234767E-03 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= exp(-b1*x(i))/(b2+b3*x(i));

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l = 0.1;
b2.l = 0.01;
b3.l = 0.02;

option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l = 0.15;
b2.l = 0.008;
b3.l = 0.010;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";
```

12.5. **Lanczos.** NIST[23] model with 6 coefficients:

$$y = \sum_{j=1}^{3} a_j \exp(-b_j x) + \varepsilon \tag{42}$$

The models are taken from an example discussed in [24]. The data were generated to different accuracy using

$$f(x) = 0.0951 \exp(-x) + 0.8607 \exp(-3 * x) + 1.5576 \exp(-5 * x)$$

Model `Lanczos1.gms` uses 14-digits of accuracy, model `Lanczos2.gms` 6 digits and `Lanczos3.gms` uses 5 digits.

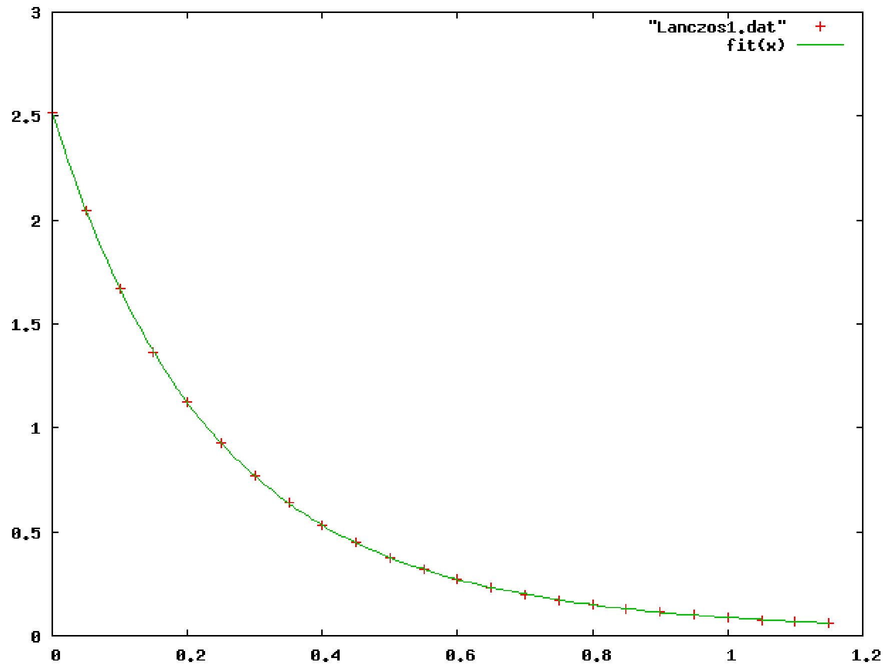

FIGURE 14. Scatter plot of model Lanczos1

12.5.1. *Model Lanczos1.gms.* [11]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   These data are taken from an example discussed in
               Lanczos (1956).  The data were generated to 14-digits
               of accuracy using
               f(x) = 0.0951*exp(-x) + 0.8607*exp(-3*x)
                                      + 1.5576*exp(-5*x).

Reference:     Lanczos, C. (1956).
               Applied Analysis.
               Englewood Cliffs, NJ:  Prentice Hall, pp. 272-280.

Data:          1 Response  (y)
               1 Predictor (x)
               24 Observations
               Average Level of Difficulty
```

[11] `www.amsterdamoptimization.com/models/regression/Lanczos1.gms`

```
               Generated Data

Model:         Exponential Class
               6 Parameters (b1 to b6)

               y = b1*exp(-b2*x) + b3*exp(-b4*x) + b5*exp(-b6*x)  +  e

          Starting values                 Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   1.2         0.5           9.5100000027E-02  5.3347304234E-11
  b2 =   0.3         0.7           1.0000000001E+00  2.7473038179E-10
  b3 =   5.6         3.6           8.6070000013E-01  1.3576062225E-10
  b4 =   5.5         4.2           3.0000000002E+00  3.3308253069E-10
  b5 =   6.5         4             1.5575999998E+00  1.8815731448E-10
  b6 =   7.6         6.3           5.0000000001E+00  1.1057500538E-10

Residual Sum of Squares:                    1.4307867721E-25
Residual Standard Deviation:                8.9156129349E-14
Degrees of Freedom:                                18
Number of Observations:                            24

$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i24/;

table data(i,*)

               y                  x
  i1    2.513400000000E+00  0.000000000000E+00
  i2    2.044333373291E+00  5.000000000000E-02
  i3    1.668404436564E+00  1.000000000000E-01
  i4    1.366418021208E+00  1.500000000000E-01
  i5    1.123232487372E+00  2.000000000000E-01
  i6    9.268897180037E-01  2.500000000000E-01
  i7    7.679338563728E-01  3.000000000000E-01
  i8    6.388775523106E-01  3.500000000000E-01
  i9    5.337835317402E-01  4.000000000000E-01
  i10   4.479363617347E-01  4.500000000000E-01
  i11   3.775847884350E-01  5.000000000000E-01
  i12   3.197393199326E-01  5.500000000000E-01
  i13   2.720130773746E-01  6.000000000000E-01
  i14   2.324965529032E-01  6.500000000000E-01
  i15   1.996589546065E-01  7.000000000000E-01
  i16   1.722704126914E-01  7.500000000000E-01
  i17   1.493405660168E-01  8.000000000000E-01
  i18   1.300700206922E-01  8.500000000000E-01
  i19   1.138119324644E-01  9.000000000000E-01
  i20   1.000415587559E-01  9.500000000000E-01
  i21   8.833209084540E-02  1.000000000000E+00
  i22   7.833544019350E-02  1.050000000000E+00
  i23   6.976693743449E-02  1.100000000000E+00
  i24   6.239312536719E-02  1.150000000000E+00

;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

set j /j1*j3/;

*
* certified values
*
```

```
parameters
  ca(j) 'certified value for a' /j1 9.5100000027E-02
                               j2 8.6070000013E-01
                               j3 1.5575999998E+00  /
  cb(j) 'certified value for b' /j1 1.0000000001E+00
                               j2 3.0000000002E+00
                               j3 5.0000000001E+00 /
  cea(j) 'certified std err for a ' /j1 5.3347304234E-11
                                   j2 1.3576062225E-10
                                   j3 1.8815731448E-10 /
  ceb(j) 'certified std err for b ' /j1 2.7473038179E-10
                                   j2 3.3308253069E-10
                                   j3 1.1057500538E-10 /
;

set st 'start value' /st1*st3/;

*
* starting values
*
table start(*,j,st)
                       st1                 st2                st3
  a.j1          1.2000000000E+00    5.0000000000E-01   9.5100000027E-02
  a.j2          5.6000000000E+00    3.6000000000E+00   8.6070000013E-01
  a.j3          6.5000000000E+00    4.0000000000E+00   1.5575999998E+00
  b.j1          3.0000000000E-01    7.0000000000E-01   1.0000000001E+00
  b.j2          5.5000000000E+00    4.2000000000E+00   3.0000000002E+00
  b.j3          7.6000000000E+00    6.3000000000E+00   5.0000000001E+00
;


*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    a(j)         'coefficient to estimate'
    b(j)         'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= sum(j, a(j)*exp[-b(j)*x(i)] );

option nlp=nls;
model nlfit /obj,fit/;

*-----------------------------------------------------------------------------
* solve with different starting points
*-----------------------------------------------------------------------------

loop(st,

   a.l(j) = start("a",j,st);
   b.l(j) = start("b",j,st);

   solve nlfit minimizing sse using nlp;

   abort$(sum(j,abs[a.l(j)-ca(j)]+abs[b.l(j)-cb(j)])>0.0001) "Accuracy problem";
   abort$(sum(j,abs[a.m(j)-cea(j)]+abs[b.m(j)-ceb(j)])>0.0001) "Accuracy problem";

);
```

12.5.2. *Model Lanczos2.gms.* [12]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   These data are taken from an example discussed in
               Lanczos (1956).  The data were generated to 6-digits
               of accuracy using
               f(x) = 0.0951*exp(-x) + 0.8607*exp(-3*x)
                                      + 1.5576*exp(-5*x).

Reference:     Lanczos, C. (1956).
               Applied Analysis.
               Englewood Cliffs, NJ:  Prentice Hall, pp. 272-280.

Data:          1 Response  (y)
               1 Predictor (x)
               24 Observations
               Average Level of Difficulty
               Generated Data

Model:         Exponential Class
               6 Parameters (b1 to b6)

               y = b1*exp(-b2*x) + b3*exp(-b4*x) + b5*exp(-b6*x)  +  e

          Starting values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   1.2         0.5           9.6251029939E-02  6.6770575477E-04
  b2 =   0.3         0.7           1.0057332849E+00  3.3989646176E-03
  b3 =   5.6         3.6           8.6424689056E-01  1.7185846685E-03
  b4 =   5.5         4.2           3.0078283915E+00  4.1707005856E-03
  b5 =   6.5         4             1.5529016879E+00  2.3744381417E-03
  b6 =   7.6         6.3           5.0028798100E+00  1.3958787284E-03

Residual Sum of Squares:                    2.2299428125E-11
Residual Standard Deviation:                1.1130395851E-06
Degrees of Freedom:                                18
Number of Observations:                            24

$offtext

*------------------------------------------------------------------------------
* data
*------------------------------------------------------------------------------

set i /i1*i24/;

table data(i,*)

              y            x
  i1    2.51340E+00  0.00000E+00
  i2    2.04433E+00  5.00000E-02
  i3    1.66840E+00  1.00000E-01
  i4    1.36642E+00  1.50000E-01
  i5    1.12323E+00  2.00000E-01
  i6    9.26890E-01  2.50000E-01
  i7    7.67934E-01  3.00000E-01
  i8    6.38878E-01  3.50000E-01
  i9    5.33784E-01  4.00000E-01
```

[12] www.amsterdamoptimization.com/models/regression/Lanczos2.gms

```
  i10   4.47936E-01  4.50000E-01
  i11   3.77585E-01  5.00000E-01
  i12   3.19739E-01  5.50000E-01
  i13   2.72013E-01  6.00000E-01
  i14   2.32497E-01  6.50000E-01
  i15   1.99659E-01  7.00000E-01
  i16   1.72270E-01  7.50000E-01
  i17   1.49341E-01  8.00000E-01
  i18   1.30070E-01  8.50000E-01
  i19   1.13812E-01  9.00000E-01
  i20   1.00042E-01  9.50000E-01
  i21   8.83321E-02  1.00000E+00
  i22   7.83354E-02  1.05000E+00
  i23   6.97669E-02  1.10000E+00
  i24   6.23931E-02  1.15000E+00

;


*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

set j /j1*j3/;

*
* certified values
*
parameters
  ca(j) 'certified value for a' /j1 9.6251029939E-02
                                 j2 8.6424689056E-01
                                 j3 1.5529016879E+00  /
  cb(j) 'certified value for b' /j1 1.0057332849E+00
                                 j2 3.0078283915E+00
                                 j3 5.0028798100E+00 /
  cea(j) 'certified std err for a ' /j1 6.6770575477E-04
                                     j2 1.7185846685E-03
                                     j3 2.3744381417E-03 /
  ceb(j) 'certified std err for b ' /j1 3.3989646176E-03
                                     j2 4.1707005856E-03
                                     j3 1.3958787284E-03 /
;

set st 'start value' /st1*st3/;

*
* starting values
*
table start(*,j,st)
                  st1                 st2                 st3
  a.j1       1.2000000000E+00   5.0000000000E-01    9.6251029939E-02
  a.j2       5.6000000000E+00   3.6000000000E+00    8.6424689056E-01
  a.j3       6.5000000000E+00   4.0000000000E+00    1.5529016879E+00
  b.j1       3.0000000000E-01   7.0000000000E-01    1.0057332849E+00
  b.j2       5.5000000000E+00   4.2000000000E+00    3.0078283915E+00
  b.j3       7.6000000000E+00   6.3000000000E+00    5.0028798100E+00
;



*------------------------------------------------------------------------------
* statistical model
*------------------------------------------------------------------------------


variables
    sse         'sum of squared errors'
    a(j)        'coefficient to estimate'
```

```
    b(j)         'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= sum(j, a(j)*exp[-b(j)*x(i)] );

option nlp=nls;
model nlfit /obj,fit/;

*------------------------------------------------------------------------------
* solve with different starting points
*------------------------------------------------------------------------------

loop(st,

   a.l(j) = start("a",j,st);
   b.l(j) = start("b",j,st);

   solve nlfit minimizing sse using nlp;

   abort$(sum(j,abs[a.l(j)-ca(j)]+abs[b.l(j)-cb(j)])>0.0001) "Accuracy problem";
   abort$(sum(j,abs[a.m(j)-cea(j)]+abs[b.m(j)-ceb(j)])>0.0001) "Accuracy problem";

);
```

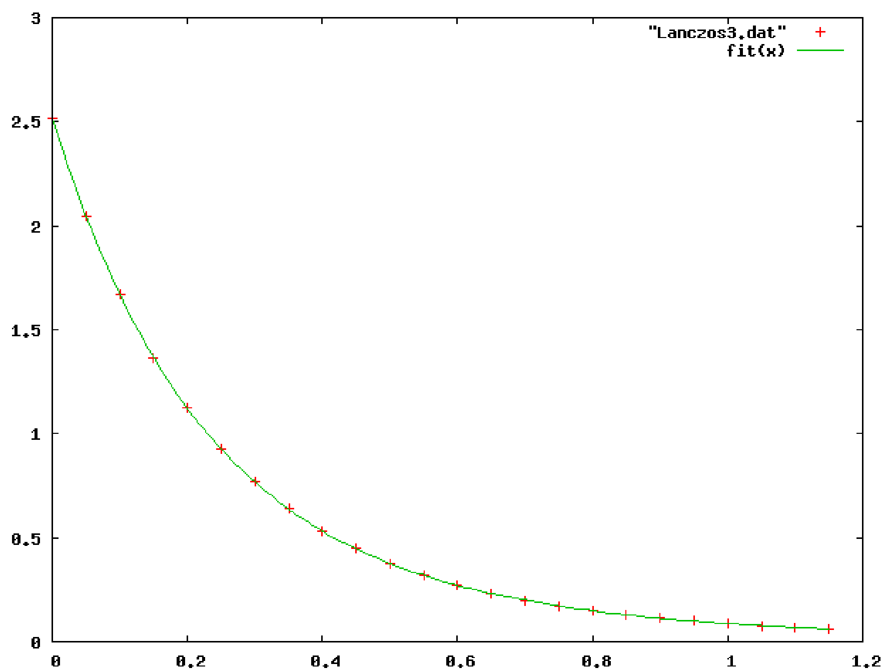

FIGURE 15. Scatter plot of model Lanczos3

12.5.3. *Model Lanczos3.gms.* [13]

[13]www.amsterdamoptimization.com/models/regression/Lanczos3.gms

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------

Procedure:     Nonlinear Least Squares Regression

Description:   These data are taken from an example discussed in
               Lanczos (1956).  The data were generated to 5-digits
               of accuracy using
               f(x) = 0.0951*exp(-x) + 0.8607*exp(-3*x)
                                     + 1.5576*exp(-5*x).


Reference:     Lanczos, C. (1956).
               Applied Analysis.
               Englewood Cliffs, NJ:  Prentice Hall, pp. 272-280.




Data:          1 Response  (y)
               1 Predictor (x)
               24 Observations
               Lower Level of Difficulty
               Generated Data

Model:         Exponential Class
               6 Parameters (b1 to b6)

               y = b1*exp(-b2*x) + b3*exp(-b4*x) + b5*exp(-b6*x)  +  e



          Starting values                 Certified Values

        Start 1     Start 2          Parameter     Standard Deviation
  b1 =   1.2         0.5          8.6816414977E-02  1.7197908859E-02
  b2 =   0.3         0.7          9.5498101505E-01  9.7041624475E-02
  b3 =   5.6         3.6          8.4400777463E-01  4.1488663282E-02
  b4 =   5.5         4.2          2.9515951832E+00  1.0766312506E-01
  b5 =   6.5         4            1.5825685901E+00  5.8371576281E-02
  b6 =   7.6         6.3          4.9863565084E+00  3.4436403035E-02

Residual Sum of Squares:                   1.6117193594E-08
Residual Standard Deviation:               2.9923229172E-05
Degrees of Freedom:                               18
Number of Observations:                           24

$offtext

*------------------------------------------------------------------------------
* data
*------------------------------------------------------------------------------

set i /i1*i24/;

table data(i,*)

               y          x
  i1      2.5134E+00  0.00000E+00
  i2      2.0443E+00  5.00000E-02
  i3      1.6684E+00  1.00000E-01
  i4      1.3664E+00  1.50000E-01
  i5      1.1232E+00  2.00000E-01
```

```
  i6      0.9269E+00  2.50000E-01
  i7      0.7679E+00  3.00000E-01
  i8      0.6389E+00  3.50000E-01
  i9      0.5338E+00  4.00000E-01
  i10     0.4479E+00  4.50000E-01
  i11     0.3776E+00  5.00000E-01
  i12     0.3197E+00  5.50000E-01
  i13     0.2720E+00  6.00000E-01
  i14     0.2325E+00  6.50000E-01
  i15     0.1997E+00  7.00000E-01
  i16     0.1723E+00  7.50000E-01
  i17     0.1493E+00  8.00000E-01
  i18     0.1301E+00  8.50000E-01
  i19     0.1138E+00  9.00000E-01
  i20     0.1000E+00  9.50000E-01
  i21     0.0883E+00  1.00000E+00
  i22     0.0783E+00  1.05000E+00
  i23     0.0698E+00  1.10000E+00
  i24     0.0624E+00  1.15000E+00

;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

set j /j1*j3/;

*
* certified values
*
parameters
  ca(j) 'certified value for a' /j1 8.6816414977E-02
                                 j2 8.4400777463E-01
                                 j3 1.5825685901E+00  /
  cb(j) 'certified value for b' /j1 9.5498101505E-01
                                 j2 2.9515951832E+00
                                 j3 4.9863565084E+00 /
  cea(j) 'certified std err for a ' /j1 1.7197908859E-02
                                     j2 4.1488663282E-02
                                     j3 5.8371576281E-02 /
  ceb(j) 'certified std err for b ' /j1 9.7041624475E-02
                                     j2 1.0766312506E-01
                                     j3 3.4436403035E-02 /
;

set st 'start value' /st1*st3/;

*
* starting values
*
table start(*,j,st)
                       st1               st2               st3
  a.j1    1.2000000000E+00  5.0000000000E-01  8.6816414977E-02
  a.j2    5.6000000000E+00  3.6000000000E+00  8.4400777463E-01
  a.j3    6.5000000000E+00  4.0000000000E+00  1.5825685901E+00
  b.j1    3.0000000000E-01  7.0000000000E-01  9.5498101505E-01
  b.j2    5.5000000000E+00  4.2000000000E+00  2.9515951832E+00
  b.j3    7.6000000000E+00  6.3000000000E+00  4.9863565084E+00
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    a(j)         'coefficient to estimate'
```

```
    b(j)          'coefficient to estimate'
;

equations
    fit(i)      'the non-linear model'
    obj         'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= sum(j, a(j)*exp[-b(j)*x(i)] );

option nlp=nls;
model nlfit /obj,fit/;

*----------------------------------------------------------------------------
* solve with different starting points
*----------------------------------------------------------------------------

loop(st,

   a.l(j) = start("a",j,st);
   b.l(j) = start("b",j,st);

   solve nlfit minimizing sse using nlp;

   abort$(sum(j,abs[a.l(j)-ca(j)]+abs[b.l(j)-cb(j)])>0.0001) "Accuracy problem";
   abort$(sum(j,abs[a.m(j)-cea(j)]+abs[b.m(j)-ceb(j)])>0.0001) "Accuracy problem";

);
```

12.6. **Gauss.** NIST[23] model with 8 coefficients:

$$y = \beta_1 \exp(-\beta_2 x) + \beta_3 \exp\left[\frac{-(x-\beta_4)^2}{\beta_5^2}\right] + \beta_6 \exp\left[\frac{-(x-\beta_7)^2}{\beta_8^2}\right] + \varepsilon \tag{43}$$

The data are two Gaussians on a decaying exponential baseline plus normally distributed zero-mean noise with variance = 6.25. The differences between models `Gauss1.gms`, `Gauss2.gms` and `Gauss3.gms` are related to how separated the two Guassians are. The last one is strongly blended and the first one is well-separated.

12.6.1. *Model Gauss1.gms.* [14]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   The data are two well-separated Gaussians on a
               decaying exponential baseline plus normally
               distributed zero-mean noise with variance = 6.25.

Reference:     Rust, B., NIST (1996).


Data:          1 Response  (y)
               1 Predictor (x)
               250 Observations
               Lower Level of Difficulty
               Generated Data
```

[14] `www.amsterdamoptimization.com/models/regression/Gauss1.gms`

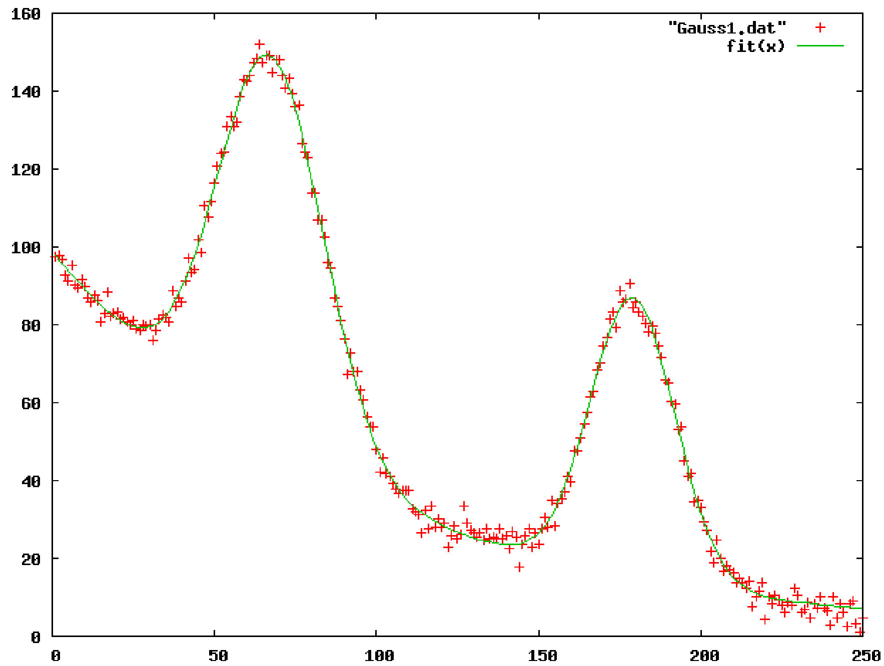

FIGURE 16. Scatter plot of model Gauss1

```
Model:          Exponential Class
                8 Parameters (b1 to b8)

                y = b1*exp( -b2*x ) + b3*exp( -(x-b4)**2 / b5**2 )
                                    + b6*exp( -(x-b7)**2 / b8**2 ) + e


           Starting values                  Certified Values

         Start 1      Start 2          Parameter     Standard Deviation
  b1 =     97.0         94.0        9.8778210871E+01  5.7527312730E-01
  b2 =      0.009        0.0105     1.0497276517E-02  1.1406289017E-04
  b3 =    100.0         99.0        1.0048990633E+02  5.8831775752E-01
  b4 =     65.0         63.0        6.7481111276E+01  1.0460593412E-01
  b5 =     20.0         25.0        2.3129773360E+01  1.7439951146E-01
  b6 =     70.0         71.0        7.1994503004E+01  6.2622793913E-01
  b7 =    178.0        180.0        1.7899805021E+02  1.2436988217E-01
  b8 =     16.5         20.0        1.8389389025E+01  2.0134312832E-01

Residual Sum of Squares:                    1.3158222432E+03
Residual Standard Deviation:                2.3317980180E+00
Degrees of Freedom:                               242
Number of Observations:                           250


$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i250/;

table data(i,*)
```

```
              y          x
i1     97.62227   1.000000
i2     97.80724   2.000000
i3     96.62247   3.000000
i4     92.59022   4.000000
i5     91.23869   5.000000
i6     95.32704   6.000000
i7     90.35040   7.000000
i8     89.46235   8.000000
i9     91.72520   9.000000
i10    89.86916  10.000000
i11    86.88076   11.00000
i12    85.94360   12.00000
i13    87.60686   13.00000
i14    86.25839   14.00000
i15    80.74976   15.00000
i16    83.03551   16.00000
i17    88.25837   17.00000
i18    82.01316   18.00000
i19    82.74098   19.00000
i20    83.30034   20.00000
i21    81.27850   21.00000
i22    81.85506   22.00000
i23    80.75195   23.00000
i24    80.09573   24.00000
i25    81.07633   25.00000
i26    78.81542   26.00000
i27    78.38596   27.00000
i28    79.93386   28.00000
i29    79.48474   29.00000
i30    79.95942   30.00000
i31    76.10691   31.00000
i32    78.39830   32.00000
i33    81.43060   33.00000
i34    82.48867   34.00000
i35    81.65462   35.00000
i36    80.84323   36.00000
i37    88.68663   37.00000
i38    84.74438   38.00000
i39    86.83934   39.00000
i40    85.97739   40.00000
i41    91.28509   41.00000
i42    97.22411   42.00000
i43    93.51733   43.00000
i44    94.10159   44.00000
i45   101.91760   45.00000
i46    98.43134   46.00000
i47    110.4214   47.00000
i48    107.6628   48.00000
i49    111.7288   49.00000
i50    116.5115   50.00000
i51    120.7609   51.00000
i52    123.9553   52.00000
i53    124.2437   53.00000
i54    130.7996   54.00000
i55    133.2960   55.00000
i56    130.7788   56.00000
i57    132.0565   57.00000
i58    138.6584   58.00000
i59    142.9252   59.00000
i60    142.7215   60.00000
i61    144.1249   61.00000
i62    147.4377   62.00000
i63    148.2647   63.00000
i64    152.0519   64.00000
i65    147.3863   65.00000
i66    149.2074   66.00000
i67    148.9537   67.00000
i68    144.5876   68.00000
i69    148.1226   69.00000
i70    148.0144   70.00000
i71    143.8893   71.00000
```

```
i72   140.9088    72.00000
i73   143.4434    73.00000
i74   139.3938    74.00000
i75   135.9878    75.00000
i76   136.3927    76.00000
i77   126.7262    77.00000
i78   124.4487    78.00000
i79   122.8647    79.00000
i80   113.8557    80.00000
i81   113.7037    81.00000
i82   106.8407    82.00000
i83   107.0034    83.00000
i84  102.46290    84.00000
i85   96.09296    85.00000
i86   94.57555    86.00000
i87   86.98824    87.00000
i88   84.90154    88.00000
i89   81.18023    89.00000
i90   76.40117    90.00000
i91   67.09200    91.00000
i92   72.67155    92.00000
i93   68.10848    93.00000
i94   67.99088    94.00000
i95   63.34094    95.00000
i96   60.55253    96.00000
i97   56.18687    97.00000
i98   53.64482    98.00000
i99   53.70307    99.00000
i100  48.07893   100.00000
i101  42.21258   101.00000
i102  45.65181   102.00000
i103  41.69728   103.00000
i104  41.24946   104.00000
i105  39.21349   105.00000
i106  37.71696    106.0000
i107  36.68395    107.0000
i108  37.30393    108.0000
i109  37.43277    109.0000
i110  37.45012    110.0000
i111  32.64648    111.0000
i112  31.84347    112.0000
i113  31.39951    113.0000
i114  26.68912    114.0000
i115  32.25323    115.0000
i116  27.61008    116.0000
i117  33.58649    117.0000
i118  28.10714    118.0000
i119  30.26428    119.0000
i120  28.01648    120.0000
i121  29.11021    121.0000
i122  23.02099    122.0000
i123  25.65091    123.0000
i124  28.50295    124.0000
i125  25.23701    125.0000
i126  26.13828    126.0000
i127  33.53260    127.0000
i128  29.25195    128.0000
i129  27.09847    129.0000
i130  26.52999    130.0000
i131  25.52401    131.0000
i132  26.69218    132.0000
i133  24.55269    133.0000
i134  27.71763    134.0000
i135  25.20297    135.0000
i136  25.61483    136.0000
i137  25.06893    137.0000
i138  27.63930    138.0000
i139  24.94851    139.0000
i140  25.86806    140.0000
i141  22.48183    141.0000
i142  26.90045    142.0000
i143  25.39919    143.0000
```

```
i144  17.90614    144.0000
i145  23.76039    145.0000
i146  25.89689    146.0000
i147  27.64231    147.0000
i148  22.86101    148.0000
i149  26.47003    149.0000
i150  23.72888    150.0000
i151  27.54334    151.0000
i152  30.52683    152.0000
i153  28.07261    153.0000
i154  34.92815    154.0000
i155  28.29194    155.0000
i156  34.19161    156.0000
i157  35.41207    157.0000
i158  37.09336    158.0000
i159  40.98330    159.0000
i160  39.53923    160.0000
i161  47.80123    161.0000
i162  47.46305    162.0000
i163  51.04166    163.0000
i164  54.58065    164.0000
i165  57.53001    165.0000
i166  61.42089    166.0000
i167  62.79032    167.0000
i168  68.51455    168.0000
i169  70.23053    169.0000
i170  74.42776    170.0000
i171  76.59911    171.0000
i172  81.62053    172.0000
i173  83.42208    173.0000
i174  79.17451    174.0000
i175  88.56985    175.0000
i176  85.66525    176.0000
i177  86.55502    177.0000
i178  90.65907    178.0000
i179  84.27290    179.0000
i180  85.72220    180.0000
i181  83.10702    181.0000
i182  82.16884    182.0000
i183  80.42568    183.0000
i184  78.15692    184.0000
i185  79.79691    185.0000
i186  77.84378    186.0000
i187  74.50327    187.0000
i188  71.57289    188.0000
i189  65.88031    189.0000
i190  65.01385    190.0000
i191  60.19582    191.0000
i192  59.66726    192.0000
i193  52.95478    193.0000
i194  53.87792    194.0000
i195  44.91274    195.0000
i196  41.09909    196.0000
i197  41.68018    197.0000
i198  34.53379    198.0000
i199  34.86419    199.0000
i200  33.14787    200.0000
i201  29.58864    201.0000
i202  27.29462    202.0000
i203  21.91439    203.0000
i204  19.08159    204.0000
i205  24.90290    205.0000
i206  19.82341    206.0000
i207  16.75551    207.0000
i208  18.24558    208.0000
i209  17.23549    209.0000
i210  16.34934    210.0000
i211  13.71285    211.0000
i212  14.75676    212.0000
i213  13.97169    213.0000
i214  12.42867    214.0000
i215  14.35519    215.0000
```

```
  i216  7.703309    216.0000
  i217 10.234410    217.0000
  i218  11.78315    218.0000
  i219  13.87768    219.0000
  i220  4.535700    220.0000
  i221 10.059280    221.0000
  i222  8.424824    222.0000
  i223 10.533120    223.0000
  i224  9.602255    224.0000
  i225  7.877514    225.0000
  i226  6.258121    226.0000
  i227  8.899865    227.0000
  i228  7.877754    228.0000
  i229  12.51191    229.0000
  i230  10.66205    230.0000
  i231  6.035400    231.0000
  i232  6.790655    232.0000
  i233  8.783535    233.0000
  i234  4.600288    234.0000
  i235  8.400915    235.0000
  i236  7.216561    236.0000
  i237 10.017410    237.0000
  i238  7.331278    238.0000
  i239  6.527863    239.0000
  i240  2.842001    240.0000
  i241 10.325070    241.0000
  i242  4.790995    242.0000
  i243  8.377101    243.0000
  i244  6.264445    244.0000
  i245  2.706213    245.0000
  i246  8.362329    246.0000
  i247  8.983658    247.0000
  i248  3.362571    248.0000
  i249  1.182746    249.0000
  i250  4.875359    250.0000
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /9.8778210871E+01/
  cb2 'certified value for b2' /1.0497276517E-02/
  cb3 'certified value for b3' /1.0048990633E+02/
  cb4 'certified value for b4' /6.7481111276E+01/
  cb5 'certified value for b5' /2.3129773360E+01/
  cb6 'certified value for b6' /7.1994503004E+01/
  cb7 'certified value for b7' /1.7899805021E+02/
  cb8 'certified value for b8' /1.8389389025E+01/
  ce1 'certified std err for b1 ' /5.7527312730E-01/
  ce2 'certified std err for b2 ' /1.1406289017E-04/
  ce3 'certified std err for b3 ' /5.8831775752E-01/
  ce4 'certified std err for b4 ' /1.0460593412E-01/
  ce5 'certified std err for b5 ' /1.7439951146E-01/
  ce6 'certified std err for b6 ' /6.2622793913E-01/
  ce7 'certified std err for b7 ' /1.2436988217E-01/
  ce8 'certified std err for b8 ' /2.0134312832E-01/
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
```

```
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
    b4           'coefficient to estimate'
    b5           'coefficient to estimate'
    b6           'coefficient to estimate'
    b7           'coefficient to estimate'
    b8           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1*exp(-b2*x(i))
                   +b3*exp(-sqr(x(i)-b4)/sqr(b5))
                   +b6*exp(-sqr(x(i)-b7)/sqr(b8));

option nlp=nls;
model nlfit /obj,fit/;


*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l=9.7000000000E+01;
b2.l=9.0000000000E-03;
b3.l=1.0000000000E+02;
b4.l=6.5000000000E+01;
b5.l=2.0000000000E+01;
b6.l=7.0000000000E+01;
b7.l=1.7800000000E+02;
b8.l=1.6500000000E+01;

solve nlfit minimizing sse using nlp;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)+abs(b4.l-cb4)
       +abs(b5.l-cb5)+abs(b6.l-cb6)+abs(b7.l-cb7)+abs(b8.l-cb8))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)+abs(b4.m-ce4)
       +abs(b5.m-ce5)+abs(b6.m-ce6)+abs(b7.m-ce7)+abs(b8.m-ce8))>0.0001) "Accuracy problem";


*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l=9.4000000000E+01;
b2.l=1.0500000000E-02;
b3.l=9.9000000000E+01;
b4.l=6.3000000000E+01;
b5.l=2.5000000000E+01;
b6.l=7.1000000000E+01;
b7.l=1.8000000000E+02;
b8.l=2.0000000000E+01;

solve nlfit minimizing sse using nlp;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)+abs(b4.l-cb4)
       +abs(b5.l-cb5)+abs(b6.l-cb6)+abs(b7.l-cb7)+abs(b8.l-cb8))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)+abs(b4.m-ce4)
       +abs(b5.m-ce5)+abs(b6.m-ce6)+abs(b7.m-ce7)+abs(b8.m-ce8))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* third set of initial values
*-----------------------------------------------------------------------------

b1.l=9.8778210871E+01;
b2.l=1.0497276517E-02;
```

```
b3.l=1.0048990633E+02;
b4.l=6.7481111276E+01;
b5.l=2.3129773360E+01;
b6.l=7.1994503004E+01;
b7.l=1.7899805021E+02;
b8.l=1.8389389025E+01;

solve nlfit minimizing sse using nlp;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)+abs(b4.l-cb4)
       +abs(b5.l-cb5)+abs(b6.l-cb6)+abs(b7.l-cb7)+abs(b8.l-cb8))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)+abs(b4.m-ce4)
       +abs(b5.m-ce5)+abs(b6.m-ce6)+abs(b7.m-ce7)+abs(b8.m-ce8))>0.0001) "Accuracy problem";
```

12.6.2. *Model Gauss2.gms.* [15]

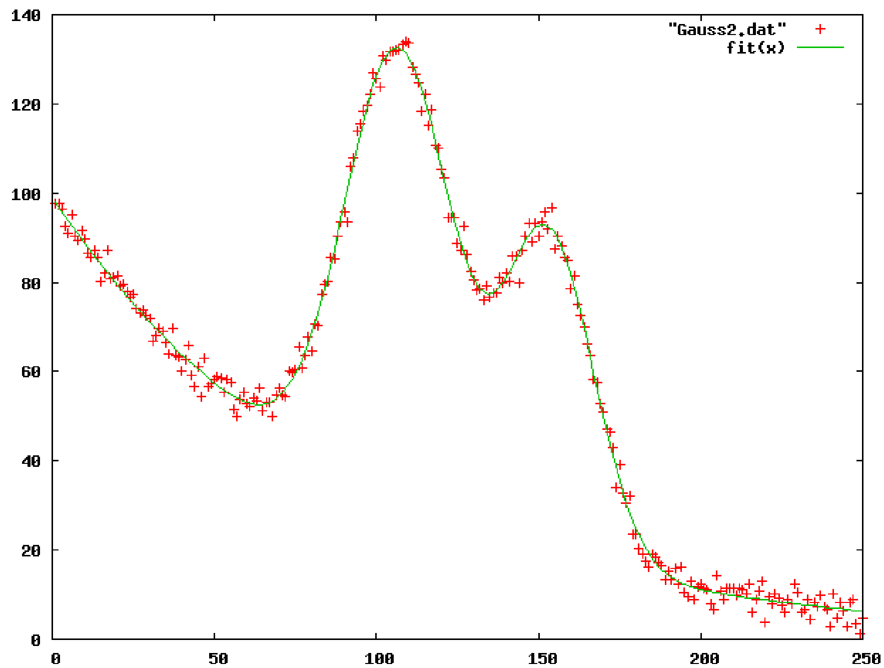

FIGURE 17. Scatter plot of model Gauss2

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   The data are two slightly-blended Gaussians on a
               decaying exponential baseline plus normally
               distributed zero-mean noise with variance = 6.25.
```

[15] www.amsterdamoptimization.com/models/regression/Gauss2.gms

```
Reference:     Rust, B., NIST (1996).

Data:          1 Response  (y)
               1 Predictor (x)
               250 Observations
               Lower Level of Difficulty
               Generated Data

Model:         Exponential Class
               8 Parameters (b1 to b8)

               y = b1*exp( -b2*x ) + b3*exp( -(x-b4)**2 / b5**2 )
                                   + b6*exp( -(x-b7)**2 / b8**2 ) + e


          Starting values                 Certified Values

        Start 1     Start 2          Parameter     Standard Deviation
  b1 =    96.0        98.0        9.9018328406E+01  5.3748766879E-01
  b2 =     0.009       0.0105     1.0994945399E-02  1.3335306766E-04
  b3 =   103.0       103.0        1.0188022528E+02  5.9217315772E-01
  b4 =   106.0       105.0        1.0703095519E+02  1.5006798316E-01
  b5 =    18.0        20.0        2.3578584029E+01  2.2695595067E-01
  b6 =    72.0        73.0        7.2045589471E+01  6.1721965884E-01
  b7 =   151.0       150.0        1.5327010194E+02  1.9466674341E-01
  b8 =    18.0        20.0        1.9525972636E+01  2.6416549393E-01

Residual Sum of Squares:                    1.2475282092E+03
Residual Standard Deviation:                2.2704790782E+00
Degrees of Freedom:                               242
Number of Observations:                           250


$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i250/;

table data(i,*)
              y            x
  i1     97.58776    1.000000
  i2     97.76344    2.000000
  i3     96.56705    3.000000
  i4     92.52037    4.000000
  i5     91.15097    5.000000
  i6     95.21728    6.000000
  i7     90.21355    7.000000
  i8     89.29235    8.000000
  i9     91.51479    9.000000
  i10    89.60966   10.000000
  i11    86.56187    11.00000
  i12    85.55316    12.00000
  i13    87.13054    13.00000
  i14    85.67940    14.00000
  i15    80.04851    15.00000
  i16    82.18925    16.00000
  i17    87.24081    17.00000
  i18    80.79407    18.00000
  i19    81.28570    19.00000
  i20    81.56940    20.00000
  i21    79.22715    21.00000
  i22    79.43275    22.00000
  i23    77.90195    23.00000
  i24    76.75468    24.00000
  i25    77.17377    25.00000
  i26    74.27348    26.00000
  i27    73.11900    27.00000
  i28    73.84826    28.00000
```

```
i29    72.47870    29.00000
i30    71.92292    30.00000
i31    66.92176    31.00000
i32    67.93835    32.00000
i33    69.56207    33.00000
i34    69.07066    34.00000
i35    66.53983    35.00000
i36    63.87883    36.00000
i37    69.71537    37.00000
i38    63.60588    38.00000
i39    63.37154    39.00000
i40    60.01835    40.00000
i41    62.67481    41.00000
i42    65.80666    42.00000
i43    59.14304    43.00000
i44    56.62951    44.00000
i45    61.21785    45.00000
i46    54.38790    46.00000
i47    62.93443    47.00000
i48    56.65144    48.00000
i49    57.13362    49.00000
i50    58.29689    50.00000
i51    58.91744    51.00000
i52    58.50172    52.00000
i53    55.22885    53.00000
i54    58.30375    54.00000
i55    57.43237    55.00000
i56    51.69407    56.00000
i57    49.93132    57.00000
i58    53.70760    58.00000
i59    55.39712    59.00000
i60    52.89709    60.00000
i61    52.31649    61.00000
i62    53.98720    62.00000
i63    53.54158    63.00000
i64    56.45046    64.00000
i65    51.32276    65.00000
i66    53.11676    66.00000
i67    53.28631    67.00000
i68    49.80555    68.00000
i69    54.69564    69.00000
i70    56.41627    70.00000
i71    54.59362    71.00000
i72    54.38520    72.00000
i73    60.15354    73.00000
i74    59.78773    74.00000
i75    60.49995    75.00000
i76    65.43885    76.00000
i77    60.70001    77.00000
i78    63.71865    78.00000
i79    67.77139    79.00000
i80    64.70934    80.00000
i81    70.78193    81.00000
i82    70.38651    82.00000
i83    77.22359    83.00000
i84    79.52665    84.00000
i85    80.13077    85.00000
i86    85.67823    86.00000
i87    85.20647    87.00000
i88    90.24548    88.00000
i89    93.61953    89.00000
i90    95.86509    90.00000
i91    93.46992    91.00000
i92    105.8137    92.00000
i93    107.8269    93.00000
i94    114.0607    94.00000
i95    115.5019    95.00000
i96    118.5110    96.00000
i97    119.6177    97.00000
i98    122.1940    98.00000
i99    126.9903    99.00000
i100   125.7005   100.00000
```

```
i101   123.7447   101.00000
i102   130.6543   102.00000
i103   129.7168   103.00000
i104   131.8240   104.00000
i105   131.8759   105.00000
i106   131.9994    106.0000
i107   132.1221    107.0000
i108   133.4414    108.0000
i109   133.8252    109.0000
i110   133.6695    110.0000
i111   128.2851    111.0000
i112   126.5182    112.0000
i113   124.7550    113.0000
i114   118.4016    114.0000
i115   122.0334    115.0000
i116   115.2059    116.0000
i117   118.7856    117.0000
i118   110.7387    118.0000
i119   110.2003    119.0000
i120  105.17290    120.0000
i121  103.44720    121.0000
i122   94.54280    122.0000
i123   94.40526    123.0000
i124   94.57964    124.0000
i125   88.76605    125.0000
i126   87.28747    126.0000
i127   92.50443    127.0000
i128   86.27997    128.0000
i129   82.44307    129.0000
i130   80.47367    130.0000
i131   78.36608    131.0000
i132   78.74307    132.0000
i133   76.12786    133.0000
i134   79.13108    134.0000
i135   76.76062    135.0000
i136   77.60769    136.0000
i137   77.76633    137.0000
i138   81.28220    138.0000
i139   79.74307    139.0000
i140   81.97964    140.0000
i141   80.02952    141.0000
i142   85.95232    142.0000
i143   85.96838    143.0000
i144   79.94789    144.0000
i145   87.17023    145.0000
i146   90.50992    146.0000
i147   93.23373    147.0000
i148   89.14803    148.0000
i149   93.11492    149.0000
i150   90.34337    150.0000
i151   93.69421    151.0000
i152   95.74256    152.0000
i153   91.85105    153.0000
i154   96.74503    154.0000
i155   87.60996    155.0000
i156   90.47012    156.0000
i157   88.11690    157.0000
i158   85.70673    158.0000
i159   85.01361    159.0000
i160   78.53040    160.0000
i161   81.34148    161.0000
i162   75.19295    162.0000
i163   72.66115    163.0000
i164   69.85504    164.0000
i165   66.29476    165.0000
i166   63.58502    166.0000
i167   58.33847    167.0000
i168   57.50766    168.0000
i169   52.80498    169.0000
i170   50.79319    170.0000
i171   47.03490    171.0000
i172   46.47090    172.0000
```

```
i173   43.09016    173.0000
i174   34.11531    174.0000
i175   39.28235    175.0000
i176   32.68386    176.0000
i177   30.44056    177.0000
i178   31.98932    178.0000
i179   23.63330    179.0000
i180   23.69643    180.0000
i181   20.26812    181.0000
i182   19.07074    182.0000
i183   17.59544    183.0000
i184   16.08785    184.0000
i185   18.94267    185.0000
i186   18.61354    186.0000
i187   17.25800    187.0000
i188   16.62285    188.0000
i189   13.48367    189.0000
i190   15.37647    190.0000
i191   13.47208    191.0000
i192   15.96188    192.0000
i193   12.32547    193.0000
i194   16.33880    194.0000
i195  10.438330    195.0000
i196   9.628715    196.0000
i197   13.12268    197.0000
i198   8.772417    198.0000
i199   11.76143    199.0000
i200   12.55020    200.0000
i201   11.33108    201.0000
i202   11.20493    202.0000
i203   7.816916    203.0000
i204   6.800675    204.0000
i205   14.26581    205.0000
i206   10.66285    206.0000
i207   8.911574    207.0000
i208   11.56733    208.0000
i209   11.58207    209.0000
i210   11.59071    210.0000
i211   9.730134    211.0000
i212   11.44237    212.0000
i213   11.22912    213.0000
i214  10.172130    214.0000
i215   12.50905    215.0000
i216   6.201493    216.0000
i217   9.019605    217.0000
i218   10.80607    218.0000
i219   13.09625    219.0000
i220   3.914271    220.0000
i221   9.567886    221.0000
i222   8.038448    222.0000
i223  10.231040    223.0000
i224   9.367410    224.0000
i225   7.695971    225.0000
i226   6.118575    226.0000
i227   8.793207    227.0000
i228   7.796692    228.0000
i229   12.45065    229.0000
i230   10.61601    230.0000
i231   6.001003    231.0000
i232   6.765098    232.0000
i233   8.764653    233.0000
i234   4.586418    234.0000
i235   8.390783    235.0000
i236   7.209202    236.0000
i237  10.012090    237.0000
i238   7.327461    238.0000
i239   6.525136    239.0000
i240   2.840065    240.0000
i241  10.323710    241.0000
i242   4.790035    242.0000
i243   8.376431    243.0000
i244   6.263980    244.0000
```

```
  i245   2.705892    245.0000
  i246   8.362109    246.0000
  i247   8.983507    247.0000
  i248   3.362469    248.0000
  i249   1.182678    249.0000
  i250   4.875312    250.0000
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /9.9018328406E+01/
  cb2 'certified value for b2' /1.0994945399E-02/
  cb3 'certified value for b3' /1.0188022528E+02/
  cb4 'certified value for b4' /1.0703095519E+02/
  cb5 'certified value for b5' /2.3578584029E+01/
  cb6 'certified value for b6' /7.2045589471E+01/
  cb7 'certified value for b7' /1.5327010194E+02/
  cb8 'certified value for b8' /1.9525972636E+01/
  ce1 'certified std err for b1 ' /5.3748766879E-01/
  ce2 'certified std err for b2 ' /1.3335306766E-04/
  ce3 'certified std err for b3 ' /5.9217315772E-01/
  ce4 'certified std err for b4 ' /1.5006798316E-01/
  ce5 'certified std err for b5 ' /2.2695595067E-01/
  ce6 'certified std err for b6 ' /6.1721965884E-01/
  ce7 'certified std err for b7 ' /1.9466674341E-01/
  ce8 'certified std err for b8 ' /2.6416549393E-01/
;


*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
    b4           'coefficient to estimate'
    b5           'coefficient to estimate'
    b6           'coefficient to estimate'
    b7           'coefficient to estimate'
    b8           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1*exp(-b2*x(i))
                   +b3*exp(-sqr(x(i)-b4)/sqr(b5))
                   +b6*exp(-sqr(x(i)-b7)/sqr(b8));

option nlp=nls;
model nlfit /obj,fit/;


*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------
```

```
b1.l=9.6000000000E+01;
b2.l=9.0000000000E-03;
b3.l=1.0300000000E+02;
b4.l=1.0600000000E+02;
b5.l=1.8000000000E+01;
b6.l=7.2000000000E+01;
b7.l=1.5100000000E+02;
b8.l=1.8000000000E+01;

solve nlfit minimizing sse using nlp;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)+abs(b4.l-cb4)
       +abs(b5.l-cb5)+abs(b6.l-cb6)+abs(b7.l-cb7)+abs(b8.l-cb8))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)+abs(b4.m-ce4)
       +abs(b5.m-ce5)+abs(b6.m-ce6)+abs(b7.m-ce7)+abs(b8.m-ce8))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l=9.8000000000E+01;
b2.l=1.0500000000E-02;
b3.l=1.0300000000E+02;
b4.l=1.0500000000E+02;
b5.l=2.0000000000E+01;
b6.l=7.3000000000E+01;
b7.l=1.5000000000E+02;
b8.l=2.0000000000E+01;

solve nlfit minimizing sse using nlp;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)+abs(b4.l-cb4)
       +abs(b5.l-cb5)+abs(b6.l-cb6)+abs(b7.l-cb7)+abs(b8.l-cb8))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)+abs(b4.m-ce4)
       +abs(b5.m-ce5)+abs(b6.m-ce6)+abs(b7.m-ce7)+abs(b8.m-ce8))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* third set of initial values
*-----------------------------------------------------------------------------

b1.l=9.9018328406E+01;
b2.l=1.0994945399E-02;
b3.l=1.0188022528E+02;
b4.l=1.0703095519E+02;
b5.l=2.3578584029E+01;
b6.l=7.2045589471E+01;
b7.l=1.5327010194E+02;
b8.l=1.9525972636E+01;

solve nlfit minimizing sse using nlp;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)+abs(b4.l-cb4)
       +abs(b5.l-cb5)+abs(b6.l-cb6)+abs(b7.l-cb7)+abs(b8.l-cb8))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)+abs(b4.m-ce4)
       +abs(b5.m-ce5)+abs(b6.m-ce6)+abs(b7.m-ce7)+abs(b8.m-ce8))>0.0001) "Accuracy problem";
```

12.6.3. *Model Gauss3.gms.* [16]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

----------------------------------------------------------------------------
```

[16] www.amsterdamoptimization.com/models/regression/Gauss3.gms

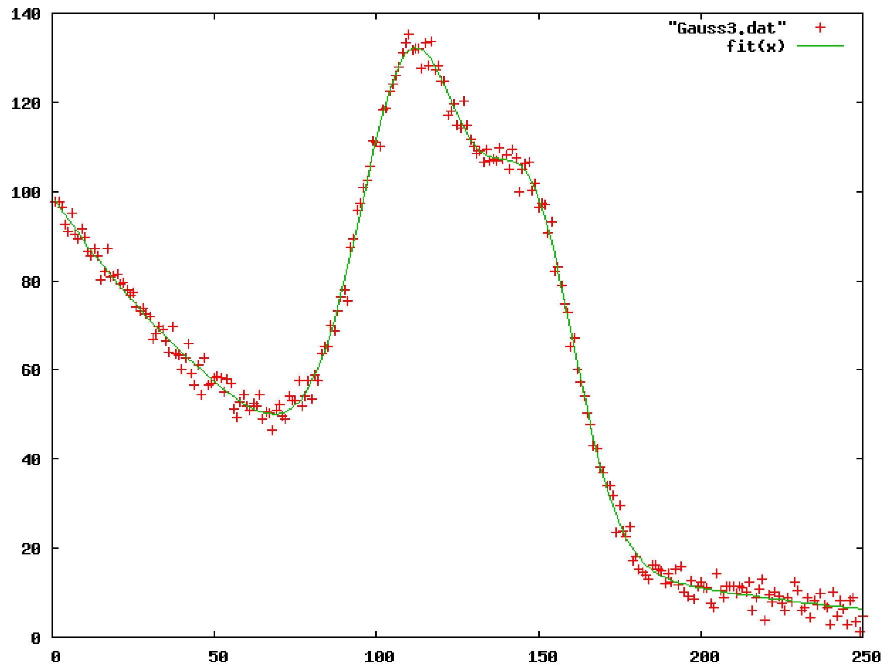

FIGURE 18. Scatter plot of model Gauss3

```
Procedure:      Nonlinear Least Squares Regression

Description:    The data are two strongly-blended Gaussians on a
                decaying exponential baseline plus normally
                distributed zero-mean noise with variance = 6.25.

Reference:      Rust, B., NIST (1996).

Data:           1 Response  (y)
                1 Predictor (x)
                250 Observations
                Average Level of Difficulty
                Generated Data

Model:          Exponential Class
                8 Parameters (b1 to b8)

                y = b1*exp( -b2*x ) + b3*exp( -(x-b4)**2 / b5**2 )
                                    + b6*exp( -(x-b7)**2 / b8**2 ) + e

          Starting values                  Certified Values

        Start 1     Start 2          Parameter     Standard Deviation
  b1 =     94.9        96.0        9.8940368970E+01  5.3005192833E-01
  b2 =      0.009       0.0096     1.0945879335E-02  1.2554058911E-04
  b3 =     90.1        80.0        1.0069553078E+02  8.1256587317E-01
  b4 =    113.0       110.0        1.1163619459E+02  3.5317859757E-01
  b5 =     20.0        25.0        2.3300500029E+01  3.6584783023E-01
  b6 =     73.8        74.0        7.3705031418E+01  1.2091239082E+00
  b7 =    140.0       139.0        1.4776164251E+02  4.0488183351E-01
  b8 =     20.0        25.0        1.9668221230E+01  3.7806634336E-01

Residual Sum of Squares:                    1.2444846360E+03
Residual Standard Deviation:                2.2677077625E+00
Degrees of Freedom:                               242
```

```
Number of Observations:                          250

$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i250/;

table data(i,*)
              y           x
  i1     97.58776    1.000000
  i2     97.76344    2.000000
  i3     96.56705    3.000000
  i4     92.52037    4.000000
  i5     91.15097    5.000000
  i6     95.21728    6.000000
  i7     90.21355    7.000000
  i8     89.29235    8.000000
  i9     91.51479    9.000000
  i10    89.60965   10.000000
  i11    86.56187    11.00000
  i12    85.55315    12.00000
  i13    87.13053    13.00000
  i14    85.67938    14.00000
  i15    80.04849    15.00000
  i16    82.18922    16.00000
  i17    87.24078    17.00000
  i18    80.79401    18.00000
  i19    81.28564    19.00000
  i20    81.56932    20.00000
  i21    79.22703    21.00000
  i22    79.43259    22.00000
  i23    77.90174    23.00000
  i24    76.75438    24.00000
  i25    77.17338    25.00000
  i26    74.27296    26.00000
  i27    73.11830    27.00000
  i28    73.84732    28.00000
  i29    72.47746    29.00000
  i30    71.92128    30.00000
  i31    66.91962    31.00000
  i32    67.93554    32.00000
  i33    69.55841    33.00000
  i34    69.06592    34.00000
  i35    66.53371    35.00000
  i36    63.87094    36.00000
  i37    69.70526    37.00000
  i38    63.59295    38.00000
  i39    63.35509    39.00000
  i40    59.99747    40.00000
  i41    62.64843    41.00000
  i42    65.77345    42.00000
  i43    59.10141    43.00000
  i44    56.57750    44.00000
  i45    61.15313    45.00000
  i46    54.30767    46.00000
  i47    62.83535    47.00000
  i48    56.52957    48.00000
  i49    56.98427    49.00000
  i50    58.11459    50.00000
  i51    58.69576    51.00000
  i52    58.23322    52.00000
  i53    54.90490    53.00000
  i54    57.91442    54.00000
  i55    56.96629    55.00000
  i56    51.13831    56.00000
  i57    49.27123    57.00000
  i58    52.92668    58.00000
  i59    54.47693    59.00000
  i60    51.81710    60.00000
```

```
i61    51.05401   61.00000
i62    52.51731   62.00000
i63    51.83710   63.00000
i64    54.48196   64.00000
i65    49.05859   65.00000
i66    50.52315   66.00000
i67    50.32755   67.00000
i68    46.44419   68.00000
i69    50.89281   69.00000
i70    52.13203   70.00000
i71    49.78741   71.00000
i72    49.01637   72.00000
i73    54.18198   73.00000
i74    53.17456   74.00000
i75    53.20827   75.00000
i76    57.43459   76.00000
i77    51.95282   77.00000
i78    54.20282   78.00000
i79    57.46687   79.00000
i80    53.60268   80.00000
i81    58.86728   81.00000
i82    57.66652   82.00000
i83    63.71034   83.00000
i84    65.24244   84.00000
i85    65.10878   85.00000
i86    69.96313   86.00000
i87    68.85475   87.00000
i88    73.32574   88.00000
i89    76.21241   89.00000
i90    78.06311   90.00000
i91    75.37701   91.00000
i92    87.54449   92.00000
i93    89.50588   93.00000
i94    95.82098   94.00000
i95    97.48390   95.00000
i96   100.86070   96.00000
i97   102.48510   97.00000
i98    105.7311   98.00000
i99    111.3489   99.00000
i100   111.0305  100.00000
i101   110.1920  101.00000
i102   118.3581  102.00000
i103   118.8086  103.00000
i104   122.4249  104.00000
i105   124.0953  105.00000
i106   125.9337   106.0000
i107   127.8533   107.0000
i108   131.0361   108.0000
i109   133.3343   109.0000
i110   135.1278   110.0000
i111   131.7113   111.0000
i112   131.9151   112.0000
i113   132.1107   113.0000
i114   127.6898   114.0000
i115   133.2148   115.0000
i116   128.2296   116.0000
i117   133.5902   117.0000
i118   127.2539   118.0000
i119   128.3482   119.0000
i120   124.8694   120.0000
i121   124.6031   121.0000
i122   117.0648   122.0000
i123   118.1966   123.0000
i124   119.5408   124.0000
i125   114.7946   125.0000
i126   114.2780   126.0000
i127   120.3484   127.0000
i128   114.8647   128.0000
i129   111.6514   129.0000
i130   110.1826   130.0000
i131   108.4461   131.0000
i132   109.0571   132.0000
```

i133	106.5308	133.0000
i134	109.4691	134.0000
i135	106.8709	135.0000
i136	107.3192	136.0000
i137	106.9000	137.0000
i138	109.6526	138.0000
i139	107.1602	139.0000
i140	108.2509	140.0000
i141	104.96310	141.0000
i142	109.3601	142.0000
i143	107.6696	143.0000
i144	99.77286	144.0000
i145	104.96440	145.0000
i146	106.1376	146.0000
i147	106.5816	147.0000
i148	100.12860	148.0000
i149	101.66910	149.0000
i150	96.44254	150.0000
i151	97.34169	151.0000
i152	96.97412	152.0000
i153	90.73460	153.0000
i154	93.37949	154.0000
i155	82.12331	155.0000
i156	83.01657	156.0000
i157	78.87360	157.0000
i158	74.86971	158.0000
i159	72.79341	159.0000
i160	65.14744	160.0000
i161	67.02127	161.0000
i162	60.16136	162.0000
i163	57.13996	163.0000
i164	54.05769	164.0000
i165	50.42265	165.0000
i166	47.82430	166.0000
i167	42.85748	167.0000
i168	42.45495	168.0000
i169	38.30808	169.0000
i170	36.95794	170.0000
i171	33.94543	171.0000
i172	34.19017	172.0000
i173	31.66097	173.0000
i174	23.56172	174.0000
i175	29.61143	175.0000
i176	23.88765	176.0000
i177	22.49812	177.0000
i178	24.86901	178.0000
i179	17.29481	179.0000
i180	18.09291	180.0000
i181	15.34813	181.0000
i182	14.77997	182.0000
i183	13.87832	183.0000
i184	12.88891	184.0000
i185	16.20763	185.0000
i186	16.29024	186.0000
i187	15.29712	187.0000
i188	14.97839	188.0000
i189	12.11330	189.0000
i190	14.24168	190.0000
i191	12.53824	191.0000
i192	15.19818	192.0000
i193	11.70478	193.0000
i194	15.83745	194.0000
i195	10.035850	195.0000
i196	9.307574	196.0000
i197	12.86800	197.0000
i198	8.571671	198.0000
i199	11.60415	199.0000
i200	12.42772	200.0000
i201	11.23627	201.0000
i202	11.13198	202.0000
i203	7.761117	203.0000
i204	6.758250	204.0000

```
  i205   14.23375    205.0000
  i206   10.63876    206.0000
  i207   8.893581    207.0000
  i208   11.55398    208.0000
  i209   11.57221    209.0000
  i210   11.58347    210.0000
  i211   9.724857    211.0000
  i212   11.43854    212.0000
  i213   11.22636    213.0000
  i214  10.170150    214.0000
  i215   12.50765    215.0000
  i216   6.200494    216.0000
  i217   9.018902    217.0000
  i218   10.80557    218.0000
  i219   13.09591    219.0000
  i220   3.914033    220.0000
  i221   9.567723    221.0000
  i222   8.038338    222.0000
  i223  10.230960    223.0000
  i224   9.367358    224.0000
  i225   7.695937    225.0000
  i226   6.118552    226.0000
  i227   8.793192    227.0000
  i228   7.796682    228.0000
  i229   12.45064    229.0000
  i230   10.61601    230.0000
  i231   6.001000    231.0000
  i232   6.765096    232.0000
  i233   8.764652    233.0000
  i234   4.586417    234.0000
  i235   8.390782    235.0000
  i236   7.209201    236.0000
  i237  10.012090    237.0000
  i238   7.327461    238.0000
  i239   6.525136    239.0000
  i240   2.840065    240.0000
  i241  10.323710    241.0000
  i242   4.790035    242.0000
  i243   8.376431    243.0000
  i244   6.263980    244.0000
  i245   2.705892    245.0000
  i246   8.362109    246.0000
  i247   8.983507    247.0000
  i248   3.362469    248.0000
  i249   1.182678    249.0000
  i250   4.875312    250.0000
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /9.8940368970E+01/
  cb2 'certified value for b2' /1.0945879335E-02/
  cb3 'certified value for b3' /1.0069553078E+02/
  cb4 'certified value for b4' /1.1163619459E+02/
  cb5 'certified value for b5' /2.3300500029E+01/
  cb6 'certified value for b6' /7.3705031418E+01/
  cb7 'certified value for b7' /1.4776164251E+02/
  cb8 'certified value for b8' /1.9668221230E+01/
  ce1 'certified std err for b1 ' /5.3005192833E-01/
  ce2 'certified std err for b2 ' /1.2554058911E-04/
  ce3 'certified std err for b3 ' /8.1256587317E-01/
  ce4 'certified std err for b4 ' /3.5317859757E-01/
  ce5 'certified std err for b5 ' /3.6584783023E-01/
```

```
  ce6 'certified std err for b6 ' /1.2091239082E+00/
  ce7 'certified std err for b7 ' /4.0488183351E-01/
  ce8 'certified std err for b8 ' /3.7806634336E-01/
;



*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
    b4           'coefficient to estimate'
    b5           'coefficient to estimate'
    b6           'coefficient to estimate'
    b7           'coefficient to estimate'
    b8           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1*exp(-b2*x(i))
                   +b3*exp(-sqr(x(i)-b4)/sqr(b5))
                   +b6*exp(-sqr(x(i)-b7)/sqr(b8));

option nlp=nls;
model nlfit /obj,fit/;


*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l=   94.9;
b2.l=    0.009;
b3.l=   90.1;
b4.l=  113.0;
b5.l=   20.0;
b6.l=   73.8;
b7.l=  140.0;
b8.l=   20.0;

solve nlfit minimizing sse using nlp;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)+abs(b4.l-cb4)
       +abs(b5.l-cb5)+abs(b6.l-cb6)+abs(b7.l-cb7)+abs(b8.l-cb8))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)+abs(b4.m-ce4)
       +abs(b5.m-ce5)+abs(b6.m-ce6)+abs(b7.m-ce7)+abs(b8.m-ce8))>0.0001) "Accuracy problem";


*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l=   96.0;
b2.l=    0.0096;
b3.l=   80.0;
b4.l=  110.0;
b5.l=   25.0;
b6.l=   74.0;
b7.l=  139.0;
b8.l=   25.0;
```

```
solve nlfit minimizing sse using nlp;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)+abs(b4.l-cb4)
       +abs(b5.l-cb5)+abs(b6.l-cb6)+abs(b7.l-cb7)+abs(b8.l-cb8))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)+abs(b4.m-ce4)
       +abs(b5.m-ce5)+abs(b6.m-ce6)+abs(b7.m-ce7)+abs(b8.m-ce8))>0.0001) "Accuracy problem";
```

12.7. **DanWood.** NIST[23] model of the form

$$y = \beta_1 x_2^{\beta} + \varepsilon \tag{44}$$

These data and model are described in [6], and originally published in [9]. The response variable is energy radiated from a carbon filament lamp per cm^2 per second, and the predictor variable is the absolute temperature of the filament in 1000 degrees Kelvin.

Note that this model can be linearized if we make the error structure multiplicative. In that case we can write:

$$\ln y = \ln \beta_1 + \beta_2 \ln x + \varepsilon \tag{45}$$

The second version performs the linear regression on this model.

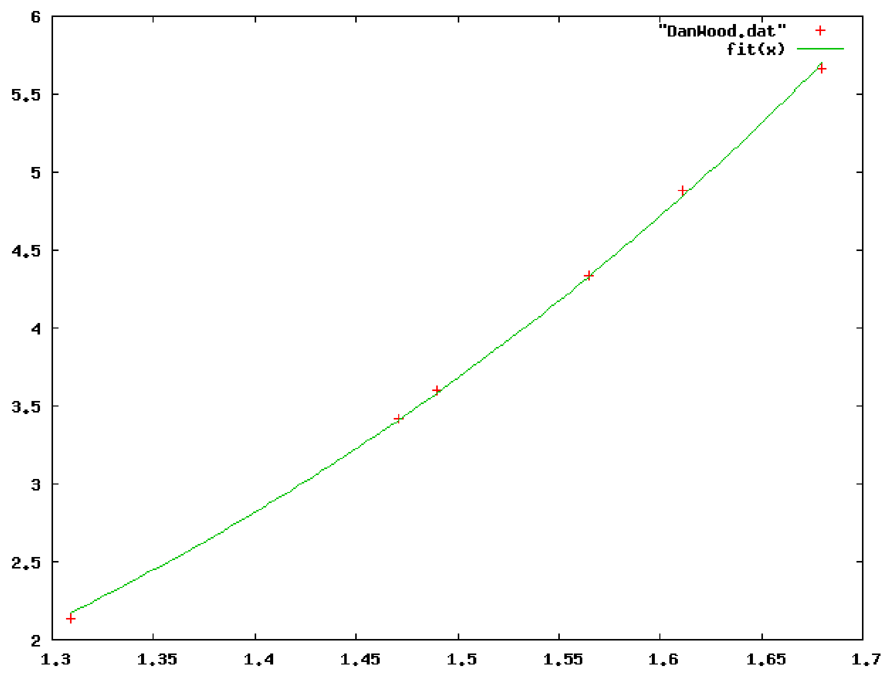

FIGURE 19. Scatter plot of model DanWood

12.7.1. *Model DanWood.gms.* [17]

[17] www.amsterdamoptimization.com/models/regression/DanWood.gms

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------

Procedure:     Nonlinear Least Squares Regression

Description:   These data and model are described in Daniel and Wood
               (1980), and originally published in E.S.Keeping,
               "Introduction to Statistical Inference," Van Nostrand
               Company, Princeton, NJ, 1962, p. 354.  The response
               variable is energy radieted from a carbon filament
               lamp per cm**2 per second, and the predictor variable
               is the absolute temperature of the filament in 1000
               degrees Kelvin.

Reference:     Daniel, C. and F. S. Wood (1980).
               Fitting Equations to Data, Second Edition.
               New York, NY:  John Wiley and Sons, pp. 428-431.

Data:          1 Response Variable  (y = energy)
               1 Predictor Variable (x = temperature)
               6 Observations
               Lower Level of Difficulty
               Observed Data

Model:         Miscellaneous Class
               2 Parameters (b1 and b2)

               y  = b1*x**b2  +  e

          Starting values                 Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   1           0.7           7.6886226176E-01  1.8281973860E-02
  b2 =   5           4             3.8604055871E+00  5.1726610913E-02

Residual Sum of Squares:                    4.3173084083E-03
Residual Standard Deviation:                3.2853114039E-02
Degrees of Freedom:                                4
Number of Observations:                            6

$offtext


*------------------------------------------------------------------------------
* data
*------------------------------------------------------------------------------

set i /i1*i6/;

table data(i,*)
             y           x
  i1   2.138E0     1.309E0
  i2   3.421E0     1.471E0
  i3   3.597E0     1.490E0
  i4   4.340E0     1.565E0
  i5   4.882E0     1.611E0
  i6   5.660E0     1.680E0
;

*
* extract data
*
parameter x(i),y(i);
```

```
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /7.6886226176E-01/
  cb2 'certified value for b2' /3.8604055871E+00/
  ce1 'certified std err for b1 ' / 1.8281973860E-02 /
  ce2 'certified std err for b2 ' / 5.1726610913E-02 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1*x(i)**b2;

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l = 1;
b2.l = 5;

option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l = 0.7;
b2.l = 4;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* third set of initial values
*-----------------------------------------------------------------------------

b1.l = 7.6886226176E-01;
b2.l = 3.8604055871E+00;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";
```

12.7.2. *Model DanWood2.gms.* [18]

```
$ontext

  Nonlinear Least Squares Regression example

  We solve this an OLS by taking logs.

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

  NLS model:  y  = b1*x**b2

  LS model:  log(y) = log(b1) + b2*log(x)

$offtext


*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i6/;

table data(i,*)
             y           x
  i1   2.138E0     1.309E0
  i2   3.421E0     1.471E0
  i3   3.597E0     1.490E0
  i4   4.340E0     1.565E0
  i5   4.882E0     1.611E0
  i6   5.660E0     1.680E0
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    logb1        'coefficient to estimate'
    b2           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  log(y(i)) =e= logb1 + b2*log(x(i));


option lp=ls;
model lfit /obj,fit/;
solve lfit minimizing sse using lp;
display "Solver results:",sse.l,logb1.l,b2.l;

scalar b1;
b1 = exp(logb1.l);
```

[18] www.amsterdamoptimization.com/models/regression/DanWood2.gms

```
display "Model results:",b1,b2.l;
```

12.8. **Kirby2.** NIST[23] model with

$$y = \frac{\beta_1 + \beta_2 x + \beta_3 x^2}{1 + \beta_4 x + \beta_5 x^2} + \varepsilon \tag{46}$$

These data are the result of a NIST study involving scanning electron microscope line with standards[21].

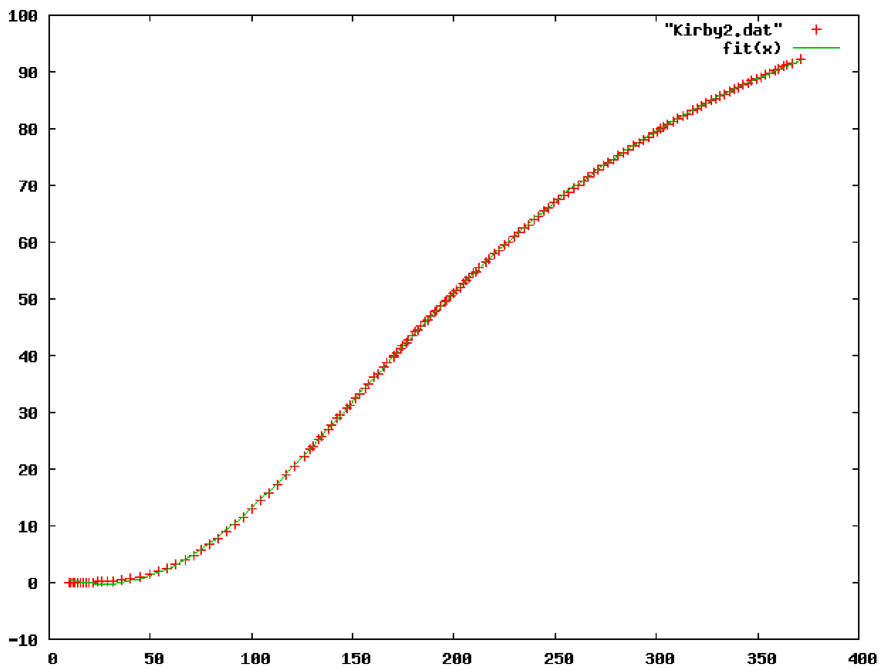

FIGURE 20. Scatter plot of model Kirby2

12.8.1. *Model Kirby2.gms.* [19]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   These data are the result of a NIST study involving
```

[19] www.amsterdamoptimization.com/models/regression/Kirby2.gms

```
               scanning electron microscope line with standards.

Reference:     Kirby, R., NIST (197?).
               Scanning electron microscope line width standards.

Data:          1 Response  (y)
               1 Predictor (x)
               151 Observations
               Average Level of Difficulty
               Observed Data

Model:         Rational Class (quadratic/quadratic)
               5 Parameters (b1 to b5)

               y = (b1 + b2*x + b3*x**2) /
                   (1 + b4*x + b5*x**2)  +  e

          Starting values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =    2           1.5          1.6745063063E+00  8.7989634338E-02
  b2 =   -0.1        -0.15        -1.3927397867E-01  4.1182041386E-03
  b3 =    0.003       0.0025       2.5961181191E-03  4.1856520458E-05
  b4 =   -0.001      -0.0015      -1.7241811870E-03  5.8931897355E-05
  b5 =    0.00001     0.00002      2.1664802578E-05  2.0129761919E-07

Residual Sum of Squares:                    3.9050739624E+00
Residual Standard Deviation:                1.6354535131E-01
Degrees of Freedom:                               146
Number of Observations:                           151

$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i151/;

table data(i,*)

              y            x
  i1       0.0082E0      9.65E0
  i2       0.0112E0     10.74E0
  i3       0.0149E0     11.81E0
  i4       0.0198E0     12.88E0
  i5       0.0248E0     14.06E0
  i6       0.0324E0     15.28E0
  i7       0.0420E0     16.63E0
  i8       0.0549E0     18.19E0
  i9       0.0719E0     19.88E0
  i10      0.0963E0     21.84E0
  i11      0.1291E0     24.00E0
  i12      0.1710E0     26.25E0
  i13      0.2314E0     28.86E0
  i14      0.3227E0     31.85E0
  i15      0.4809E0     35.79E0
  i16      0.7084E0     40.18E0
  i17      1.0220E0     44.74E0
  i18      1.4580E0     49.53E0
  i19      1.9520E0     53.94E0
  i20      2.5410E0     58.29E0
  i21      3.2230E0     62.63E0
  i22      3.9990E0     67.03E0
  i23      4.8520E0     71.25E0
  i24      5.7320E0     75.22E0
  i25      6.7270E0     79.33E0
  i26      7.8350E0     83.56E0
  i27      9.0250E0     87.75E0
  i28     10.2670E0     91.93E0
  i29     11.5780E0     96.10E0
  i30     12.9440E0    100.28E0
```

```
i31    14.3770E0    104.46E0
i32    15.8560E0    108.66E0
i33    17.3310E0    112.71E0
i34    18.8850E0    116.88E0
i35    20.5750E0    121.33E0
i36    22.3200E0    125.79E0
i37    22.3030E0    125.79E0
i38    23.4600E0    128.74E0
i39    24.0600E0    130.27E0
i40    25.2720E0    133.33E0
i41    25.8530E0    134.79E0
i42    27.1100E0    137.93E0
i43    27.6580E0    139.33E0
i44    28.9240E0    142.46E0
i45    29.5110E0    143.90E0
i46    30.7100E0    146.91E0
i47    31.3500E0    148.51E0
i48    32.5200E0    151.41E0
i49    33.2300E0    153.17E0
i50    34.3300E0    155.97E0
i51    35.0600E0    157.76E0
i52    36.1700E0    160.56E0
i53    36.8400E0    162.30E0
i54    38.0100E0    165.21E0
i55    38.6700E0    166.90E0
i56    39.8700E0    169.92E0
i57    40.0300E0    170.32E0
i58    40.5000E0    171.54E0
i59    41.3700E0    173.79E0
i60    41.6700E0    174.57E0
i61    42.3100E0    176.25E0
i62    42.7300E0    177.34E0
i63    43.4600E0    179.19E0
i64    44.1400E0    181.02E0
i65    44.5500E0    182.08E0
i66    45.2200E0    183.88E0
i67    45.9200E0    185.75E0
i68    46.3000E0    186.80E0
i69    47.0000E0    188.63E0
i70    47.6800E0    190.45E0
i71    48.0600E0    191.48E0
i72    48.7400E0    193.35E0
i73    49.4100E0    195.22E0
i74    49.7600E0    196.23E0
i75    50.4300E0    198.05E0
i76    51.1100E0    199.97E0
i77    51.5000E0    201.06E0
i78    52.1200E0    202.83E0
i79    52.7600E0    204.69E0
i80    53.1800E0    205.86E0
i81    53.7800E0    207.58E0
i82    54.4600E0    209.50E0
i83    54.8300E0    210.65E0
i84    55.4000E0    212.33E0
i85    56.4300E0    215.43E0
i86    57.0300E0    217.16E0
i87    58.0000E0    220.21E0
i88    58.6100E0    221.98E0
i89    59.5800E0    225.06E0
i90    60.1100E0    226.79E0
i91    61.1000E0    229.92E0
i92    61.6500E0    231.69E0
i93    62.5900E0    234.77E0
i94    63.1200E0    236.60E0
i95    64.0300E0    239.63E0
i96    64.6200E0    241.50E0
i97    65.4900E0    244.48E0
i98    66.0300E0    246.40E0
i99    66.8900E0    249.35E0
i100   67.4200E0    251.32E0
i101   68.2300E0    254.22E0
i102   68.7700E0    256.24E0
```

```
  i103    69.5900E0    259.11E0
  i104    70.1100E0    261.18E0
  i105    70.8600E0    264.02E0
  i106    71.4300E0    266.13E0
  i107    72.1600E0    268.94E0
  i108    72.7000E0    271.09E0
  i109    73.4000E0    273.87E0
  i110    73.9300E0    276.08E0
  i111    74.6000E0    278.83E0
  i112    75.1600E0    281.08E0
  i113    75.8200E0    283.81E0
  i114    76.3400E0    286.11E0
  i115    76.9800E0    288.81E0
  i116    77.4800E0    291.08E0
  i117    78.0800E0    293.75E0
  i118    78.6000E0    295.99E0
  i119    79.1700E0    298.64E0
  i120    79.6200E0    300.84E0
  i121    79.8800E0    302.02E0
  i122    80.1900E0    303.48E0
  i123    80.6600E0    305.65E0
  i124    81.2200E0    308.27E0
  i125    81.6600E0    310.41E0
  i126    82.1600E0    313.01E0
  i127    82.5900E0    315.12E0
  i128    83.1400E0    317.71E0
  i129    83.5000E0    319.79E0
  i130    84.0000E0    322.36E0
  i131    84.4000E0    324.42E0
  i132    84.8900E0    326.98E0
  i133    85.2600E0    329.01E0
  i134    85.7400E0    331.56E0
  i135    86.0700E0    333.56E0
  i136    86.5400E0    336.10E0
  i137    86.8900E0    338.08E0
  i138    87.3200E0    340.60E0
  i139    87.6500E0    342.57E0
  i140    88.1000E0    345.08E0
  i141    88.4300E0    347.02E0
  i142    88.8300E0    349.52E0
  i143    89.1200E0    351.44E0
  i144    89.5400E0    353.93E0
  i145    89.8500E0    355.83E0
  i146    90.2500E0    358.32E0
  i147    90.5500E0    360.20E0
  i148    90.9300E0    362.67E0
  i149    91.2000E0    364.53E0
  i150    91.5500E0    367.00E0
  i151    92.2000E0    371.30E0

;


*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /  1.6745063063E+00 /
  cb2 'certified value for b2' / -1.3927397867E-01 /
  cb3 'certified value for b3' /  2.5961181191E-03 /
  cb4 'certified value for b4' / -1.7241811870E-03 /
  cb5 'certified value for b5' /  2.1664802578E-05 /
  ce1 'certified std err for b1 ' / 8.7989634338E-02 /
  ce2 'certified std err for b2 ' / 4.1182041386E-03 /
  ce3 'certified std err for b3 ' / 4.1856520458E-05 /
```

```
  ce4 'certified std err for b4 ' / 5.8931897355E-05 /
  ce5 'certified std err for b5 ' / 2.0129761919E-07 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
    b4           'coefficient to estimate'
    b5           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= (b1 + b2*x(i) + b3*x(i)**2) /
                   (1 + b4*x(i) + b5*x(i)**2);

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l =  2;
b2.l = -0.1;
b3.l =  0.003;
b4.l = -0.001;
b5.l =  0.00001;


option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l,b5.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4)+abs(b5.l-cb5))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4)+abs(b5.m-ce5))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =    1.5;
b2.l =   -0.15;
b3.l =    0.0025;
b4.l =   -0.0015;
b5.l =    0.00002;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l,b5.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4)+abs(b5.l-cb5))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4)+abs(b5.m-ce5))>0.0001) "Accuracy problem";
```

12.9. **Hahn1.** NIST[23] model with

$$y = \frac{\beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3}{1 + \beta_5 x + \beta_6 x^2 + \beta_7 x^3} + \varepsilon \tag{47}$$

These data are the result of a NIST study involving the thermal expansion of copper. The response variable is the coefficient of thermal expansion, and the predictor variable is temperature in degrees kelvin[14].

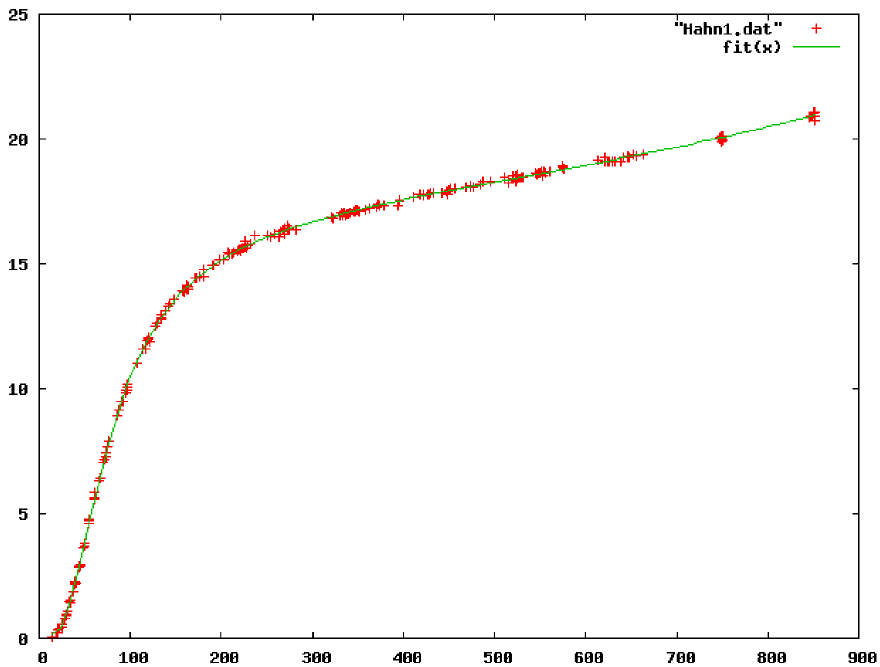

FIGURE 21. Scatter plot of model Hahn1

12.9.1. *Model Hahn1.gms.* [20]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   These data are the result of a NIST study involving
               the thermal expansion of copper.  The response
               variable is the coefficient of thermal expansion, and
               the predictor variable is temperature in degrees
               kelvin.

Reference:     Hahn, T., NIST (197?).
               Copper Thermal Expansion Study.

Data:          1 Response  (y = coefficient of thermal expansion)
               1 Predictor (x = temperature, degrees kelvin)
               236 Observations
               Average Level of Difficulty
```

[20] www.amsterdamoptimization.com/models/regression/Hahn1.gms

```
               Observed Data

Model:         Rational Class (cubic/cubic)
               7 Parameters (b1 to b7)

               y = (b1+b2*x+b3*x**2+b4*x**3) /
                   (1+b5*x+b6*x**2+b7*x**3)  +  e

          Starting values                 Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   10           1           1.0776351733E+00  1.7070154742E-01
  b2 =   -1          -0.1        -1.2269296921E-01  1.2000289189E-02
  b3 =    0.05        0.005       4.0863750610E-03  2.2508314937E-04
  b4 =   -0.00001    -0.000001   -1.4262662514E-06  2.7578037666E-07
  b5 =   -0.05       -0.005      -5.7609940901E-03  2.4712888219E-04
  b6 =    0.001       0.0001      2.4053735503E-04  1.0449373768E-05
  b7 =   -0.000001   -0.0000001  -1.2314450199E-07  1.3027335327E-08

Residual Sum of Squares:                    1.5324382854E+00
Residual Standard Deviation:                8.1803852243E-02
Degrees of Freedom:                               229
Number of Observations:                           236

$offtext

*------------------------------------------------------------------------------
* data
*------------------------------------------------------------------------------

set i /i1*i236/;

table data(i,*)
             y               x
  i1        .591E0         24.41E0
  i2       1.547E0         34.82E0
  i3       2.902E0         44.09E0
  i4       2.894E0         45.07E0
  i5       4.703E0         54.98E0
  i6       6.307E0         65.51E0
  i7       7.03E0          70.53E0
  i8       7.898E0         75.70E0
  i9       9.470E0         89.57E0
  i10      9.484E0         91.14E0
  i11     10.072E0         96.40E0
  i12     10.163E0         97.19E0
  i13     11.615E0        114.26E0
  i14     12.005E0        120.25E0
  i15     12.478E0        127.08E0
  i16     12.982E0        133.55E0
  i17     12.970E0        133.61E0
  i18     13.926E0        158.67E0
  i19     14.452E0        172.74E0
  i20     14.404E0        171.31E0
  i21     15.190E0        202.14E0
  i22     15.550E0        220.55E0
  i23     15.528E0        221.05E0
  i24     15.499E0        221.39E0
  i25     16.131E0        250.99E0
  i26     16.438E0        268.99E0
  i27     16.387E0        271.80E0
  i28     16.549E0        271.97E0
  i29     16.872E0        321.31E0
  i30     16.830E0        321.69E0
  i31     16.926E0        330.14E0
  i32     16.907E0        333.03E0
  i33     16.966E0        333.47E0
  i34     17.060E0        340.77E0
  i35     17.122E0        345.65E0
  i36     17.311E0        373.11E0
  i37     17.355E0        373.79E0
  i38     17.668E0        411.82E0
```

```
  i39     17.767E0        419.51E0
  i40     17.803E0        421.59E0
  i41     17.765E0        422.02E0
  i42     17.768E0        422.47E0
  i43     17.736E0        422.61E0
  i44     17.858E0        441.75E0
  i45     17.877E0        447.41E0
  i46     17.912E0        448.7E0
  i47     18.046E0        472.89E0
  i48     18.085E0        476.69E0
  i49     18.291E0        522.47E0
  i50     18.357E0        522.62E0
  i51     18.426E0        524.43E0
  i52     18.584E0        546.75E0
  i53     18.610E0        549.53E0
  i54     18.870E0        575.29E0
  i55     18.795E0        576.00E0
  i56     19.111E0        625.55E0
  i57       .367E0         20.15E0
  i58       .796E0         28.78E0
  i59      0.892E0         29.57E0
  i60      1.903E0         37.41E0
  i61      2.150E0         39.12E0
  i62      3.697E0         50.24E0
  i63      5.870E0         61.38E0
  i64      6.421E0         66.25E0
  i65      7.422E0         73.42E0
  i66      9.944E0         95.52E0
  i67     11.023E0        107.32E0
  i68     11.87E0         122.04E0
  i69     12.786E0        134.03E0
  i70     14.067E0        163.19E0
  i71     13.974E0        163.48E0
  i72     14.462E0        175.70E0
  i73     14.464E0        179.86E0
  i74     15.381E0        211.27E0
  i75     15.483E0        217.78E0
  i76     15.59E0         219.14E0
  i77     16.075E0        262.52E0
  i78     16.347E0        268.01E0
  i79     16.181E0        268.62E0
  i80     16.915E0        336.25E0
  i81     17.003E0        337.23E0
  i82     16.978E0        339.33E0
  i83     17.756E0        427.38E0
  i84     17.808E0        428.58E0
  i85     17.868E0        432.68E0
  i86     18.481E0        528.99E0
  i87     18.486E0        531.08E0
  i88     19.090E0        628.34E0
  i89     16.062E0        253.24E0
  i90     16.337E0        273.13E0
  i91     16.345E0        273.66E0
  i92     16.388E0        282.10E0
  i93     17.159E0        346.62E0
  i94     17.116E0        347.19E0
  i95     17.164E0        348.78E0
  i96     17.123E0        351.18E0
  i97     17.979E0        450.10E0
  i98     17.974E0        450.35E0
  i99     18.007E0        451.92E0
  i100    17.993E0        455.56E0
  i101    18.523E0        552.22E0
  i102    18.669E0        553.56E0
  i103    18.617E0        555.74E0
  i104    19.371E0        652.59E0
  i105    19.330E0        656.20E0
  i106     0.080E0         14.13E0
  i107     0.248E0         20.41E0
  i108     1.089E0         31.30E0
  i109     1.418E0         33.84E0
  i110     2.278E0         39.70E0
```

```
i111     3.624E0        48.83E0
i112     4.574E0        54.50E0
i113     5.556E0        60.41E0
i114     7.267E0        72.77E0
i115     7.695E0        75.25E0
i116     9.136E0        86.84E0
i117     9.959E0        94.88E0
i118     9.957E0        96.40E0
i119    11.600E0       117.37E0
i120    13.138E0       139.08E0
i121    13.564E0       147.73E0
i122    13.871E0       158.63E0
i123    13.994E0       161.84E0
i124    14.947E0       192.11E0
i125    15.473E0       206.76E0
i126    15.379E0       209.07E0
i127    15.455E0       213.32E0
i128    15.908E0       226.44E0
i129    16.114E0       237.12E0
i130    17.071E0       330.90E0
i131    17.135E0       358.72E0
i132    17.282E0       370.77E0
i133    17.368E0       372.72E0
i134    17.483E0       396.24E0
i135    17.764E0       416.59E0
i136    18.185E0       484.02E0
i137    18.271E0       495.47E0
i138    18.236E0       514.78E0
i139    18.237E0       515.65E0
i140    18.523E0       519.47E0
i141    18.627E0       544.47E0
i142    18.665E0       560.11E0
i143    19.086E0       620.77E0
i144     0.214E0        18.97E0
i145     0.943E0        28.93E0
i146     1.429E0        33.91E0
i147     2.241E0        40.03E0
i148     2.951E0        44.66E0
i149     3.782E0        49.87E0
i150     4.757E0        55.16E0
i151     5.602E0        60.90E0
i152     7.169E0        72.08E0
i153     8.920E0        85.15E0
i154    10.055E0        97.06E0
i155    12.035E0       119.63E0
i156    12.861E0       133.27E0
i157    13.436E0       143.84E0
i158    14.167E0       161.91E0
i159    14.755E0       180.67E0
i160    15.168E0       198.44E0
i161    15.651E0       226.86E0
i162    15.746E0       229.65E0
i163    16.216E0       258.27E0
i164    16.445E0       273.77E0
i165    16.965E0       339.15E0
i166    17.121E0       350.13E0
i167    17.206E0       362.75E0
i168    17.250E0       371.03E0
i169    17.339E0       393.32E0
i170    17.793E0       448.53E0
i171    18.123E0       473.78E0
i172    18.49E0        511.12E0
i173    18.566E0       524.70E0
i174    18.645E0       548.75E0
i175    18.706E0       551.64E0
i176    18.924E0       574.02E0
i177    19.1E0         623.86E0
i178     0.375E0        21.46E0
i179     0.471E0        24.33E0
i180     1.504E0        33.43E0
i181     2.204E0        39.22E0
i182     2.813E0        44.18E0
```

```
  i183    4.765E0        55.02E0
  i184    9.835E0        94.33E0
  i185   10.040E0        96.44E0
  i186   11.946E0       118.82E0
  i187   12.596E0       128.48E0
  i188   13.303E0       141.94E0
  i189   13.922E0       156.92E0
  i190   14.440E0       171.65E0
  i191   14.951E0       190.00E0
  i192   15.627E0       223.26E0
  i193   15.639E0       223.88E0
  i194   15.814E0       231.50E0
  i195   16.315E0       265.05E0
  i196   16.334E0       269.44E0
  i197   16.430E0       271.78E0
  i198   16.423E0       273.46E0
  i199   17.024E0       334.61E0
  i200   17.009E0       339.79E0
  i201   17.165E0       349.52E0
  i202   17.134E0       358.18E0
  i203   17.349E0       377.98E0
  i204   17.576E0       394.77E0
  i205   17.848E0       429.66E0
  i206   18.090E0       468.22E0
  i207   18.276E0       487.27E0
  i208   18.404E0       519.54E0
  i209   18.519E0       523.03E0
  i210   19.133E0       612.99E0
  i211   19.074E0       638.59E0
  i212   19.239E0       641.36E0
  i213   19.280E0       622.05E0
  i214   19.101E0       631.50E0
  i215   19.398E0       663.97E0
  i216   19.252E0       646.9E0
  i217   19.89E0        748.29E0
  i218   20.007E0       749.21E0
  i219   19.929E0       750.14E0
  i220   19.268E0       647.04E0
  i221   19.324E0       646.89E0
  i222   20.049E0       746.9E0
  i223   20.107E0       748.43E0
  i224   20.062E0       747.35E0
  i225   20.065E0       749.27E0
  i226   19.286E0       647.61E0
  i227   19.972E0       747.78E0
  i228   20.088E0       750.51E0
  i229   20.743E0       851.37E0
  i230   20.83E0        845.97E0
  i231   20.935E0       847.54E0
  i232   21.035E0       849.93E0
  i233   20.93E0        851.61E0
  i234   21.074E0       849.75E0
  i235   21.085E0       850.98E0
  i236   20.935E0       848.23E0

;


*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /  1.0776351733E+00 /
  cb2 'certified value for b2' / -1.2269296921E-01 /
  cb3 'certified value for b3' /  4.0863750610E-03 /
```

```
  cb4 'certified value for b4' / -1.4262662514E-06 /
  cb5 'certified value for b5' / -5.7609940901E-03 /
  cb6 'certified value for b6' /  2.4053735503E-04 /
  cb7 'certified value for b7' / -1.2314450199E-07 /
  ce1 'certified std err for b1 ' / 1.7070154742E-01 /
  ce2 'certified std err for b2 ' / 1.2000289189E-02 /
  ce3 'certified std err for b3 ' / 2.2508314937E-04 /
  ce4 'certified std err for b4 ' / 2.7578037666E-07 /
  ce5 'certified std err for b5 ' / 2.4712888219E-04 /
  ce6 'certified std err for b6 ' / 1.0449373768E-05 /
  ce7 'certified std err for b7 ' / 1.3027335327E-08 /
;


*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
    b4           'coefficient to estimate'
    b5           'coefficient to estimate'
    b6           'coefficient to estimate'
    b7           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= (b1 + b2*x(i) + b3*x(i)**2 + b4*x(i)**3) /
                   (1 + b5*x(i) + b6*x(i)**2 + b7*x(i)**3);

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l =  10;
b2.l =  -1;
b3.l =   0.05;
b4.l =  -0.00001;
b5.l =  -0.05;
b6.l =   0.001;
b7.l =  -0.000001;

option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l,b5.l,b6.l,b7.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4)+abs(b5.l-cb5)+abs(b6.l-cb6)
       +abs(b7.l-cb7))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4)+abs(b5.m-ce5)+abs(b6.m-ce6)
       +abs(b7.m-ce7))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =    1;
b2.l =   -0.1;
b3.l =    0.005;
b4.l =   -0.000001;
b5.l =   -0.005;
```

```
b6.l =    0.0001;
b7.l =   -0.0000001;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l,b5.l,b6.l,b7.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4)+abs(b5.l-cb5)+abs(b6.l-cb6)
       +abs(b7.l-cb7))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4)+abs(b5.m-ce5)+abs(b6.m-ce6)
       +abs(b7.m-ce7))>0.0001) "Accuracy problem";
```

12.10. **Nelson.** Multiple regression problem with:

$$\log(y) = \beta_1 - \beta_2 x_1 * \exp(-\beta_3 x_2) + \varepsilon \tag{48}$$

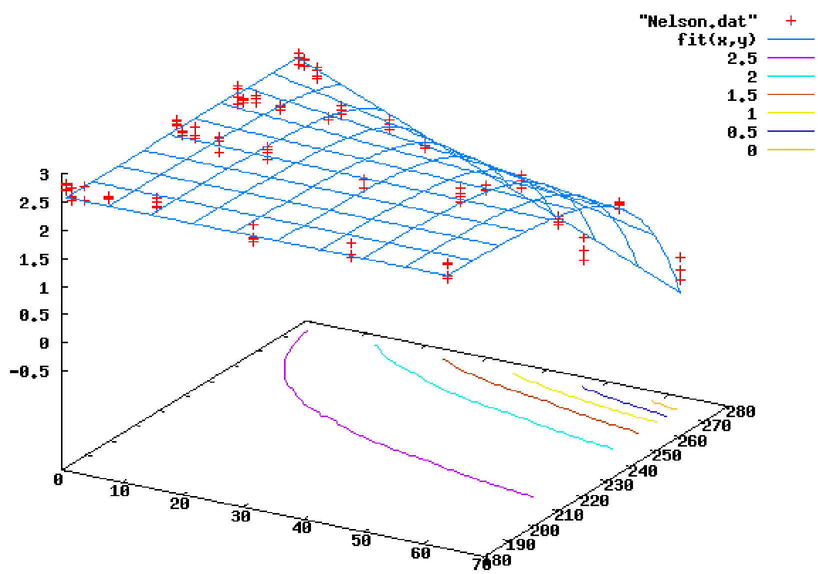

FIGURE 22. Plot of model Nelson

These data are the result of a study involving the analysis of performance degradation data from accelerated tests, published in[30]. The response variable is dialectric breakdown strength in kilo-volts, and the predictor variables are time in weeks and temperature in degrees Celcius.

12.10.1. *Model Nelson.gms.* [21]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007
```

[21] www.amsterdamoptimization.com/models/regression/Nelson.gms

```
  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   These data are the result of a study involving
               the analysis of performance degradation data from
               accelerated tests, published in IEEE Transactions
               on Reliability.  The response variable is dialectric
               breakdown strength in kilo-volts, and the predictor
               variables are time in weeks and temperature in degrees
               Celcius.

Reference:     Nelson, W. (1981).
               Analysis of Performance-Degradation Data.
               IEEE Transactions on Reliability.
               Vol. 2, R-30, No. 2, pp. 149-155.

Data:          1 Response   ( y = dialectric breakdown strength)
               2 Predictors (x1 = time; x2 = temperature)
               128 Observations
               Average Level of Difficulty
               Observed Data

Model:         Exponential Class
               3 Parameters (b1 to b3)

               log[y] = b1 - b2*x1 * exp[-b3*x2]  +  e

          Starting values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =    2           2.5          2.5906836021E+00  1.9149996413E-02
  b2 =    0.0001      0.000000005  5.6177717026E-09  6.1124096540E-09
  b3 =   -0.01       -0.05        -5.7701013174E-02  3.9572366543E-03

Residual Sum of Squares:                    3.7976833176E+00
Residual Standard Deviation:                1.7430280130E-01
Degrees of Freedom:                               125
Number of Observations:                           128

$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i128/;

table data(i,*)

            y              x1            x2
  i1      15.00E0         1E0         180E0
  i2      17.00E0         1E0         180E0
  i3      15.50E0         1E0         180E0
  i4      16.50E0         1E0         180E0
  i5      15.50E0         1E0         225E0
  i6      15.00E0         1E0         225E0
  i7      16.00E0         1E0         225E0
  i8      14.50E0         1E0         225E0
  i9      15.00E0         1E0         250E0
  i10     14.50E0         1E0         250E0
  i11     12.50E0         1E0         250E0
  i12     11.00E0         1E0         250E0
  i13     14.00E0         1E0         275E0
  i14     13.00E0         1E0         275E0
  i15     14.00E0         1E0         275E0
  i16     11.50E0         1E0         275E0
  i17     14.00E0         2E0         180E0
  i18     16.00E0         2E0         180E0
```

```
  i19    13.00E0        2E0         180E0
  i20    13.50E0        2E0         180E0
  i21    13.00E0        2E0         225E0
  i22    13.50E0        2E0         225E0
  i23    12.50E0        2E0         225E0
  i24    12.50E0        2E0         225E0
  i25    12.50E0        2E0         250E0
  i26    12.00E0        2E0         250E0
  i27    11.50E0        2E0         250E0
  i28    12.00E0        2E0         250E0
  i29    13.00E0        2E0         275E0
  i30    11.50E0        2E0         275E0
  i31    13.00E0        2E0         275E0
  i32    12.50E0        2E0         275E0
  i33    13.50E0        4E0         180E0
  i34    17.50E0        4E0         180E0
  i35    17.50E0        4E0         180E0
  i36    13.50E0        4E0         180E0
  i37    12.50E0        4E0         225E0
  i38    12.50E0        4E0         225E0
  i39    15.00E0        4E0         225E0
  i40    13.00E0        4E0         225E0
  i41    12.00E0        4E0         250E0
  i42    13.00E0        4E0         250E0
  i43    12.00E0        4E0         250E0
  i44    13.50E0        4E0         250E0
  i45    10.00E0        4E0         275E0
  i46    11.50E0        4E0         275E0
  i47    11.00E0        4E0         275E0
  i48     9.50E0        4E0         275E0
  i49    15.00E0        8E0         180E0
  i50    15.00E0        8E0         180E0
  i51    15.50E0        8E0         180E0
  i52    16.00E0        8E0         180E0
  i53    13.00E0        8E0         225E0
  i54    10.50E0        8E0         225E0
  i55    13.50E0        8E0         225E0
  i56    14.00E0        8E0         225E0
  i57    12.50E0        8E0         250E0
  i58    12.00E0        8E0         250E0
  i59    11.50E0        8E0         250E0
  i60    11.50E0        8E0         250E0
  i61     6.50E0        8E0         275E0
  i62     5.50E0        8E0         275E0
  i63     6.00E0        8E0         275E0
  i64     6.00E0        8E0         275E0
  i65    18.50E0       16E0         180E0
  i66    17.00E0       16E0         180E0
  i67    15.30E0       16E0         180E0
  i68    16.00E0       16E0         180E0
  i69    13.00E0       16E0         225E0
  i70    14.00E0       16E0         225E0
  i71    12.50E0       16E0         225E0
  i72    11.00E0       16E0         225E0
  i73    12.00E0       16E0         250E0
  i74    12.00E0       16E0         250E0
  i75    11.50E0       16E0         250E0
  i76    12.00E0       16E0         250E0
  i77     6.00E0       16E0         275E0
  i78     6.00E0       16E0         275E0
  i79     5.00E0       16E0         275E0
  i80     5.50E0       16E0         275E0
  i81    12.50E0       32E0         180E0
  i82    13.00E0       32E0         180E0
  i83    16.00E0       32E0         180E0
  i84    12.00E0       32E0         180E0
  i85    11.00E0       32E0         225E0
  i86     9.50E0       32E0         225E0
  i87    11.00E0       32E0         225E0
  i88    11.00E0       32E0         225E0
  i89    11.00E0       32E0         250E0
  i90    10.00E0       32E0         250E0
```

```
  i91     10.50E0       32E0         250E0
  i92     10.50E0       32E0         250E0
  i93      2.70E0       32E0         275E0
  i94      2.70E0       32E0         275E0
  i95      2.50E0       32E0         275E0
  i96      2.40E0       32E0         275E0
  i97     13.00E0       48E0         180E0
  i98     13.50E0       48E0         180E0
  i99     16.50E0       48E0         180E0
  i100    13.60E0       48E0         180E0
  i101    11.50E0       48E0         225E0
  i102    10.50E0       48E0         225E0
  i103    13.50E0       48E0         225E0
  i104    12.00E0       48E0         225E0
  i105     7.00E0       48E0         250E0
  i106     6.90E0       48E0         250E0
  i107     8.80E0       48E0         250E0
  i108     7.90E0       48E0         250E0
  i109     1.20E0       48E0         275E0
  i110     1.50E0       48E0         275E0
  i111     1.00E0       48E0         275E0
  i112     1.50E0       48E0         275E0
  i113    13.00E0       64E0         180E0
  i114    12.50E0       64E0         180E0
  i115    16.50E0       64E0         180E0
  i116    16.00E0       64E0         180E0
  i117    11.00E0       64E0         225E0
  i118    11.50E0       64E0         225E0
  i119    10.50E0       64E0         225E0
  i120    10.00E0       64E0         225E0
  i121     7.27E0       64E0         250E0
  i122     7.50E0       64E0         250E0
  i123     6.70E0       64E0         250E0
  i124     7.60E0       64E0         250E0
  i125     1.50E0       64E0         275E0
  i126     1.00E0       64E0         275E0
  i127     1.20E0       64E0         275E0
  i128     1.20E0       64E0         275E0

;


*
* extract data
*
parameter x1(i),x2(i),y(i);
x1(i) = data(i,'x1');
x2(i) = data(i,'x2');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /  2.5906836021E+00 /
  cb2 'certified value for b2' /  5.6177717026E-09 /
  cb3 'certified value for b3' / -5.7701013174E-02 /
  ce1 'certified std err for b1 ' / 1.9149996413E-02 /
  ce2 'certified std err for b2 ' / 6.1124096540E-09 /
  ce3 'certified std err for b3 ' / 3.9572366543E-03 /
;


*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse         'sum of squared errors'
    b1          'coefficient to estimate'
    b2          'coefficient to estimate'
```

```
    b3           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  log(y(i)) =e= b1 - b2*x1(i) * exp[-b3*x2(i)];

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l =    2;
b2.l =    0.0001;
b3.l =   -0.01;

option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =   2.5;
b2.l =   0.000000005;
b3.l =  -0.05;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";
```

12.11. **MGH.** The MGH set of models from the NIST[23] collection are used in testing unconstrained solvers[29]. Some of them are quite challenging.

12.11.1. *MGH09.* The model[22]:

$$y = \frac{\beta_1(x^2 + \beta_2 x)}{(x^2 + \beta_3 x + \beta_4)} \tag{49}$$

This problem was found to be difficult for some very good algorithms. There is a local minimum at $(+\infty, -14.07..., -\infty, -\infty)$ with final sum of squares .00102734...[29].

12.11.2. *Model MGH09.gms.* [22]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------

Procedure:     Nonlinear Least Squares Regression

```

[22] www.amsterdamoptimization.com/models/regression/MGH09.gms

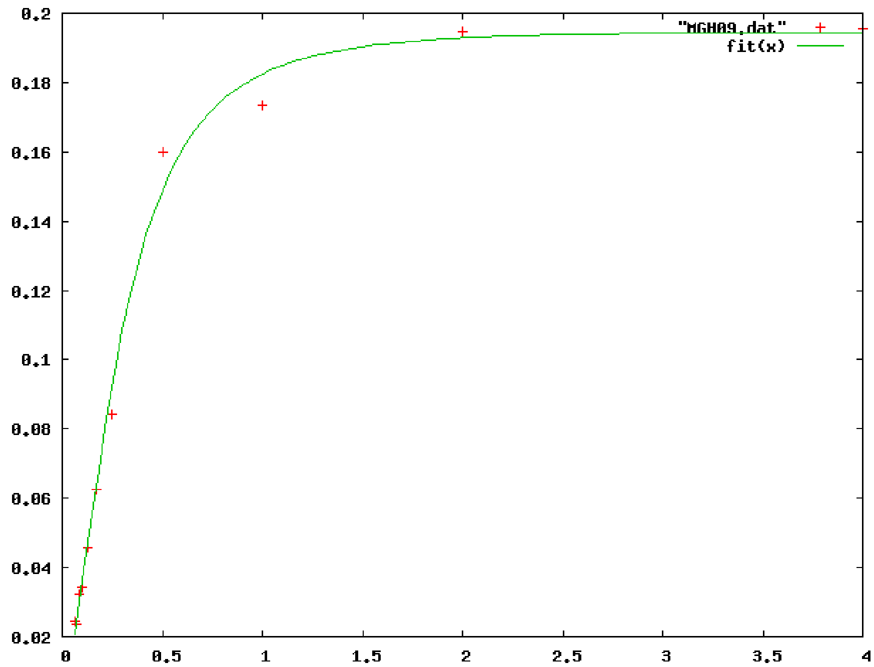

FIGURE 23. Scatter plot of model MGH09

```
Description:   This problem was found to be difficult for some very
               good algorithms.  There is a local minimum at (+inf,
               -14.07..., -inf, -inf) with final sum of squares
               0.00102734....

               See More, J. J., Garbow, B. S., and Hillstrom, K. E.
               (1981).  Testing unconstrained optimization software.
               ACM Transactions on Mathematical Software. 7(1):
               pp. 17-41.

Reference:     Kowalik, J.S., and M. R. Osborne, (1978).
               Methods for Unconstrained Optimization Problems.
               New York, NY:  Elsevier North-Holland.

Data:          1 Response  (y)
               1 Predictor (x)
               11 Observations
               Higher Level of Difficulty
               Generated Data

Model:         Rational Class (linear/quadratic)
               4 Parameters (b1 to b4)

               y = b1*(x**2+x*b2) / (x**2+x*b3+b4)  +  e

          Starting values                  Certified Values

        Start 1     Start 2          Parameter     Standard Deviation
  b1 =   25          0.25         1.9280693458E-01  1.1435312227E-02
  b2 =   39          0.39         1.9128232873E-01  1.9633220911E-01
  b3 =   41.5        0.415        1.2305650693E-01  8.0842031232E-02
  b4 =   39          0.39         1.3606233068E-01  9.0025542308E-02

Residual Sum of Squares:                    3.0750560385E-04
Residual Standard Deviation:                6.6279236551E-03
```

```
Degrees of Freedom:                                    7
Number of Observations:                               11

$offtext


*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i11/;

table data(i,*)
             y               x
  i1    1.957000E-01    4.000000E+00
  i2    1.947000E-01    2.000000E+00
  i3    1.735000E-01    1.000000E+00
  i4    1.600000E-01    5.000000E-01
  i5    8.440000E-02    2.500000E-01
  i6    6.270000E-02    1.670000E-01
  i7    4.560000E-02    1.250000E-01
  i8    3.420000E-02    1.000000E-01
  i9    3.230000E-02    8.330000E-02
  i10   2.350000E-02    7.140000E-02
  i11   2.460000E-02    6.250000E-02
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /1.9280693458E-01/
  cb2 'certified value for b2' /1.9128232873E-01/
  cb3 'certified value for b3' /1.2305650693E-01/
  cb4 'certified value for b4' /1.3606233068E-01/
  ce1 'certified std err for b1 ' / 1.1435312227E-02 /
  ce2 'certified std err for b2 ' / 1.9633220911E-01 /
  ce3 'certified std err for b3 ' / 8.0842031232E-02 /
  ce4 'certified std err for b4 ' / 9.0025542308E-02 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
    b4           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1*(sqr(x(i))+x(i)*b2) / (sqr(x(i))+x(i)*b3+b4);

option nlp=nls;
model nlfit /obj,fit/;
```

```
*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l =  25;
b2.l =  39;
b3.l =  41.5;
b4.l =  39;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)+abs(b4.l-cb4))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)+abs(b4.m-ce4))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =  0.25;
b2.l =  0.39;
b3.l =  0.415;
b4.l =  0.39;


solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)+abs(b4.l-cb4))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)+abs(b4.m-ce4))>0.0001) "Accuracy problem";
```

12.11.3. *MGH10.* The model is:

$$y = \beta_1 \exp\left[\frac{\beta_2}{x+\beta_3}\right] + \varepsilon \tag{50}$$

This problem was found to be difficult for some very good algorithms[29, 27].

12.11.4. *Model MGH10.gms.* [23]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   This problem was found to be difficult for some very
               good algorithms.

               See More, J. J., Garbow, B. S., and Hillstrom, K. E.
               (1981).  Testing unconstrained optimization software.
               ACM Transactions on Mathematical Software. 7(1):
               pp. 17-41.

Reference:     Meyer, R. R. (1970).
               Theoretical and computational aspects of nonlinear
               regression.  In Nonlinear Programming, Rosen,
               Mangasarian and Ritter (Eds).
               New York, NY: Academic Press, pp. 465-486.

Data:          1 Response  (y)
               1 Predictor (x)
               16 Observations
```

[23] www.amsterdamoptimization.com/models/regression/MGH10.gms

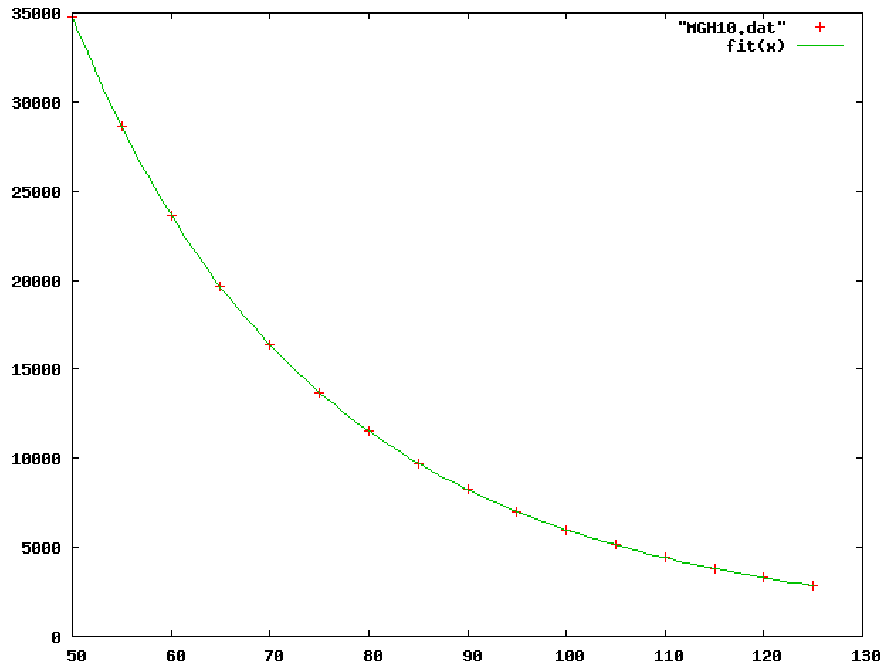

FIGURE 24. Scatter plot of model MGH10

```
               Higher Level of Difficulty
               Generated Data

Model:         Exponential Class
               3 Parameters (b1 to b3)

               y = b1 * exp[b2/(x+b3)]  +  e

          Starting values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =        2          0.02       5.6096364710E-03  1.5687892471E-04
  b2 =    400000       4000         6.1813463463E+03  2.3309021107E+01
  b3 =     25000        250         3.4522363462E+02  7.8486103508E-01

Residual Sum of Squares:                     8.7945855171E+01
Residual Standard Deviation:                 2.6009740065E+00
Degrees of Freedom:                                 13
Number of Observations:                             16

$offtext


*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i16/;

table data(i,*)
              y               x
   i1    3.478000E+04    5.000000E+01
   i2    2.861000E+04    5.500000E+01
   i3    2.365000E+04    6.000000E+01
   i4    1.963000E+04    6.500000E+01
```

```
   i5    1.637000E+04    7.000000E+01
   i6    1.372000E+04    7.500000E+01
   i7    1.154000E+04    8.000000E+01
   i8    9.744000E+03    8.500000E+01
   i9    8.261000E+03    9.000000E+01
   i10   7.030000E+03    9.500000E+01
   i11   6.005000E+03    1.000000E+02
   i12   5.147000E+03    1.050000E+02
   i13   4.427000E+03    1.100000E+02
   i14   3.820000E+03    1.150000E+02
   i15   3.307000E+03    1.200000E+02
   i16   2.872000E+03    1.250000E+02
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' / 5.6096364710E-03 /
  cb2 'certified value for b2' / 6.1813463463E+03 /
  cb3 'certified value for b3' / 3.4522363462E+02 /
  ce1 'certified std err for b1 ' / 1.5687892471E-04 /
  ce2 'certified std err for b2 ' / 2.3309021107E+01 /
  ce3 'certified std err for b3 ' / 7.8486103508E-01 /
;


*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model, LS format'
    obj        'objective, LS format'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1 * exp[b2/(x(i)+b3)] ;

model nlfit /obj,fit/;

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l =       2;
b2.l =  400000;
b3.l =   25000;

* The starting point is bad.
* We need to extend the max number of function evaluation calls.

$onecho > nls.opt
* This option is needed in MGH10.gms
mxfcal 10000
$offecho
```

```
option nlp=nls;
nlfit.optfile=1;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";


*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =     0.02;
b2.l =  4000;
b3.l =   250;

option nlp=nls;
nlfit.optfile=1;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";
```

12.11.5. *MGH17.* This problem was found to be difficult for some very good algorithms. See [29, 31]. It also causes some problems with NL2SOL. The first set of initial values will cause termination with *False Convergence*. The second starting point is closer and yields good results.

To find the optimal fit for the first starting point anyway, we throw the model formulated as an optimization model into CONOPT. Then the solution CONOPT finds is fed into NLS. This sounds more difficult than it is: GAMS will automatically do this when we solve the model twice by repeating the solve statement.

The model is:

$$y = \beta_1 + \beta_2 \exp(-\beta_4 x) + \beta_3 \exp(-\beta_5 x) + \varepsilon \tag{51}$$

12.11.6. *Model MGH17.gms.* [24]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   This problem was found to be difficult for some very
               good algorithms.

               See More, J. J., Garbow, B. S., and Hillstrom, K. E.
               (1981).  Testing unconstrained optimization software.
               ACM Transactions on Mathematical Software. 7(1):
               pp. 17-41.

Reference:     Osborne, M. R. (1972).
               Some aspects of nonlinear least squares
               calculations.  In Numerical Methods for Nonlinear
               Optimization, Lootsma (Ed).
               New York, NY:  Academic Press, pp. 171-189.
```

[24] www.amsterdamoptimization.com/models/regression/MGH17.gms

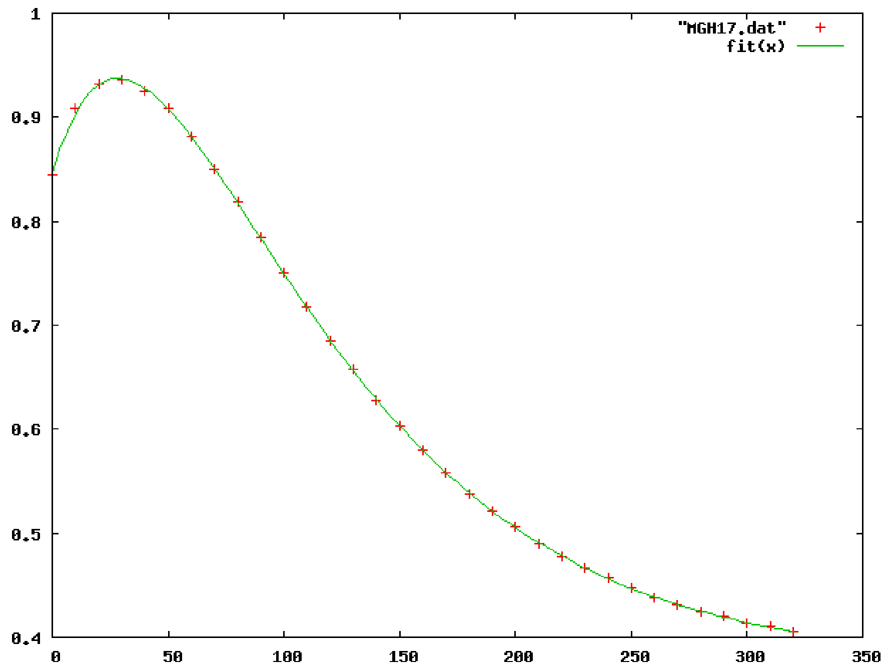

FIGURE 25. Scatter plot of model MGH17

```
Data:          1 Response  (y)
               1 Predictor (x)
               33 Observations
               Average Level of Difficulty
               Generated Data

Model:         Exponential Class
               5 Parameters (b1 to b5)

               y = b1 + b2*exp[-x*b4] + b3*exp[-x*b5]  +  e

          Starting values                  Certified Values

        Start 1     Start 2          Parameter     Standard Deviation
  b1 =     50         0.5          3.7541005211E-01  2.0723153551E-03
  b2 =    150         1.5          1.9358469127E+00  2.2031669222E-01
  b3 =   -100        -1           -1.4646871366E+00  2.2175707739E-01
  b4 =      1          0.01        1.2867534640E-02  4.4861358114E-04
  b5 =      2          0.02        2.2122699662E-02  8.9471996575E-04

Residual Sum of Squares:                    5.4648946975E-05
Residual Standard Deviation:                1.3970497866E-03
Degrees of Freedom:                                28
Number of Observations:                            33


$offtext


*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i33/;
```

```
table data(i,*)
              y               x
   i1    8.440000E-01    0.000000E+00
   i2    9.080000E-01    1.000000E+01
   i3    9.320000E-01    2.000000E+01
   i4    9.360000E-01    3.000000E+01
   i5    9.250000E-01    4.000000E+01
   i6    9.080000E-01    5.000000E+01
   i7    8.810000E-01    6.000000E+01
   i8    8.500000E-01    7.000000E+01
   i9    8.180000E-01    8.000000E+01
   i10   7.840000E-01    9.000000E+01
   i11   7.510000E-01    1.000000E+02
   i12   7.180000E-01    1.100000E+02
   i13   6.850000E-01    1.200000E+02
   i14   6.580000E-01    1.300000E+02
   i15   6.280000E-01    1.400000E+02
   i16   6.030000E-01    1.500000E+02
   i17   5.800000E-01    1.600000E+02
   i18   5.580000E-01    1.700000E+02
   i19   5.380000E-01    1.800000E+02
   i20   5.220000E-01    1.900000E+02
   i21   5.060000E-01    2.000000E+02
   i22   4.900000E-01    2.100000E+02
   i23   4.780000E-01    2.200000E+02
   i24   4.670000E-01    2.300000E+02
   i25   4.570000E-01    2.400000E+02
   i26   4.480000E-01    2.500000E+02
   i27   4.380000E-01    2.600000E+02
   i28   4.310000E-01    2.700000E+02
   i29   4.240000E-01    2.800000E+02
   i30   4.200000E-01    2.900000E+02
   i31   4.140000E-01    3.000000E+02
   i32   4.110000E-01    3.100000E+02
   i33   4.060000E-01    3.200000E+02
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /  3.7541005211E-01 /
  cb2 'certified value for b2' /  1.9358469127E+00 /
  cb3 'certified value for b3' / -1.4646871366E+00 /
  cb4 'certified value for b4' /  1.2867534640E-02 /
  cb5 'certified value for b5' /  2.2122699662E-02 /
  ce1 'certified std err for b1 ' / 2.0723153551E-03 /
  ce2 'certified std err for b2 ' / 2.2031669222E-01 /
  ce3 'certified std err for b3 ' / 2.2175707739E-01 /
  ce4 'certified std err for b4 ' / 4.4861358114E-04 /
  ce5 'certified std err for b5 ' / 8.9471996575E-04 /
;


*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
```

```
    b4          'coefficient to estimate'
    b5          'coefficient to estimate'
    r(i)        'residuals'
;

equations
    fit1(i)     'the non-linear model, LS format'
    obj1        'objective, LS format'
    fit2(i)     'the non-linear model, NLP format'
    obj2        'objective, NLP format'
;


obj1..     sse =n= 0;
fit1(i)..  y(i) =e= b1 + b2*exp[-x(i)*b4] + b3*exp[-x(i)*b5];

obj2..     sse =e= sum(i, sqr(r(i)));
fit2(i)..  y(i) =e= b1 + b2*exp[-x(i)*b4] + b3*exp[-x(i)*b5] + r(i);

model nlfit1 /obj1,fit1/;
model nlfit2 /obj2,fit2/;

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------


b1.l =    50;
b2.l =   150;
b3.l =  -100;
b4.l =     1;
b5.l =     2;


* The starting point is bad.
* NL2SOL will fail with "False Convergence".
* We solve this by first using CONOPT to find an
* optimal solution which we then pass on to NL2SOL.

option nlp=conopt;
solve nlfit2 minimizing sse using nlp;
option nlp=nls;
solve nlfit2 minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l,b5.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4)+abs(b5.l-cb5))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4)+abs(b5.m-ce5))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =  0.5;
b2.l =  1.5;
b3.l = -1;
b4.l =   0.01;
b5.l =   0.02;

solve nlfit1 minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l,b5.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4)+abs(b5.l-cb5))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4)+abs(b5.m-ce5))>0.0001) "Accuracy problem";
```

12.12. **Roszman1.** This model from [23, 34] has the form:

$$y = \beta_1 - \beta_2 x - \frac{\arctan\left[\frac{\beta_3}{x-\beta_4}\right]}{\pi} + \varepsilon \tag{52}$$

The data are the result of a NIST study involving quantum defects in iodine atoms. The response variable is the number of quantum defects, and the predictor variable is the excited energy state. The argument to the ARCTAN function is in radians.

12.12.1. *Model Roszman1.gms.* [25]

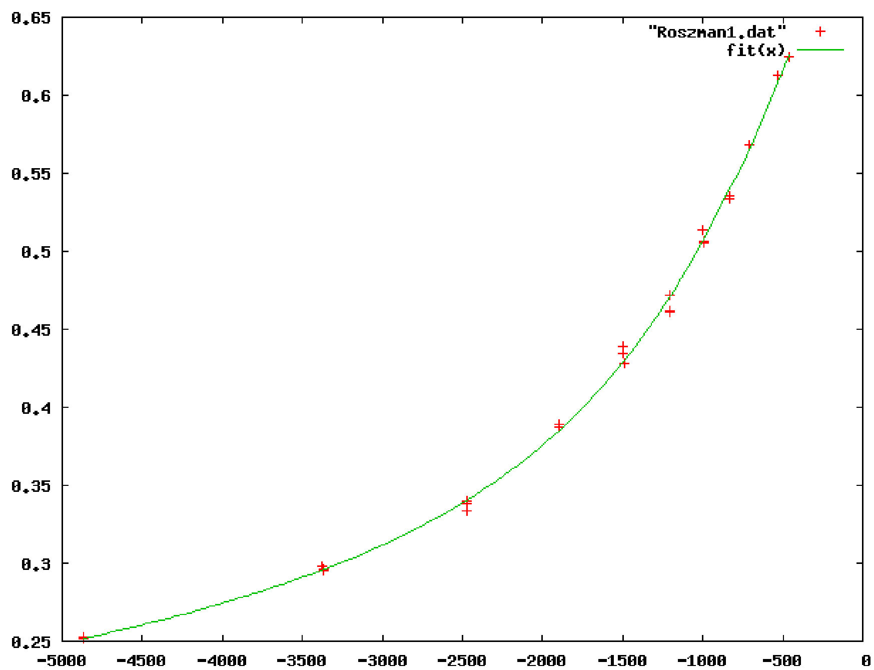

FIGURE 26. Scatter plot of model Roszman1

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   These data are the result of a NIST study involving
               quantum defects in iodine atoms.  The response
               variable is the number of quantum defects, and the
               predictor variable is the excited energy state.
               The argument to the ARCTAN function is in radians.
```

[25] www.amsterdamoptimization.com/models/regression/Roszman1.gms

```
Reference:     Roszman, L., NIST (19??).
               Quantum Defects for Sulfur I Atom.

Data:          1 Response  (y = quantum defect)
               1 Predictor (x = excited state energy)
               25 Observations
               Average Level of Difficulty
               Observed Data

Model:         Miscellaneous Class
               4 Parameters (b1 to b4)

               pi = 3.1415926535897932384626433832795E0
               y =  b1 - b2*x - arctan[b3/(x-b4)]/pi  +  e


          Starting Values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =      0.1         0.2         2.0196866396E-01  1.9172666023E-02
  b2 =     -0.00001    -0.000005   -6.1953516256E-06  3.2058931691E-06
  b3 =   1000        1200           1.2044556708E+03  7.4050983057E+01
  b4 =   -100        -150          -1.8134269537E+02  4.9573513849E+01

Residual Sum of Squares:                    4.9484847331E-04
Residual Standard Deviation:                4.8542984060E-03
Degrees of Freedom:                                 21
Number of Observations:                             25
$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i25/;

table data(i,*)
             y           x
  i1     0.252429    -4868.68
  i2     0.252141    -4868.09
  i3     0.251809    -4867.41
  i4     0.297989    -3375.19
  i5     0.296257    -3373.14
  i6     0.295319    -3372.03
  i7     0.339603    -2473.74
  i8     0.337731    -2472.35
  i9     0.333820    -2469.45
  i10    0.389510    -1894.65
  i11    0.386998    -1893.40
  i12    0.438864    -1497.24
  i13    0.434887    -1495.85
  i14    0.427893    -1493.41
  i15    0.471568    -1208.68
  i16    0.461699    -1206.18
  i17    0.461144    -1206.04
  i18    0.513532     -997.92
  i19    0.506641     -996.61
  i20    0.505062     -996.31
  i21    0.535648     -834.94
  i22    0.533726     -834.66
  i23    0.568064     -710.03
  i24    0.612886     -530.16
  i25    0.624169     -464.17

;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
```

```
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' /  2.0196866396E-01 /
  cb2 'certified value for b2' / -6.1953516256E-06 /
  cb3 'certified value for b3' /  1.2044556708E+03 /
  cb4 'certified value for b4' / -1.8134269537E+02 /
  ce1 'certified std err for b1 ' / 1.9172666023E-02 /
  ce2 'certified std err for b2 ' / 3.2058931691E-06 /
  ce3 'certified std err for b3 ' / 7.4050983057E+01 /
  ce4 'certified std err for b4 ' / 4.9573513849E+01 /
;



*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
    b4           'coefficient to estimate'
;

equations
    fit(i)       'the non-linear model'
    obj          'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1 - b2*x(i) - arctan[b3/(x(i)-b4)]/pi;


model nlfit /obj,fit/;
option nlp=nls;

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l =     0.1;
b2.l =    -0.00001;
b3.l =  1000;
b4.l =  -100;


solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4))>0.001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4))>0.001) "Accuracy problem";



*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------


b1.l =     0.2;
b2.l =    -0.000005;
b3.l =  1200;
b4.l =  -150;
```

```
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4))>0.001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4))>0.001) "Accuracy problem";
```

12.13. **ENSO.** The model looks like[23, 18]:

$$\begin{aligned} y = {} & \beta_1 + \beta_2 \cos(2\pi x/12) + \beta_3 \sin(2\pi x/12) \\ & + \beta_5 \cos(2\pi x/\beta_4) + \beta_6 \sin(2\pi x/\beta_4) \\ & + \beta_8 \cos(2\pi x/\beta_7) + \beta_9 \sin(2\pi x/\beta_7) + \varepsilon \end{aligned} \tag{53}$$

These data are monthly averaged atmospheric pressure differences between Easter Island and Darwin, Australia. This difference drives the trade winds in the southern hemisphere. Fourier analysis of the data reveals 3 significant cycles. The annual cycle is the strongest, but cycles with periods of approximately 44 and 26 months are also present. These cycles correspond to the El Nino and the Southern Oscillation. Arguments to the SIN and COS functions are in radians.

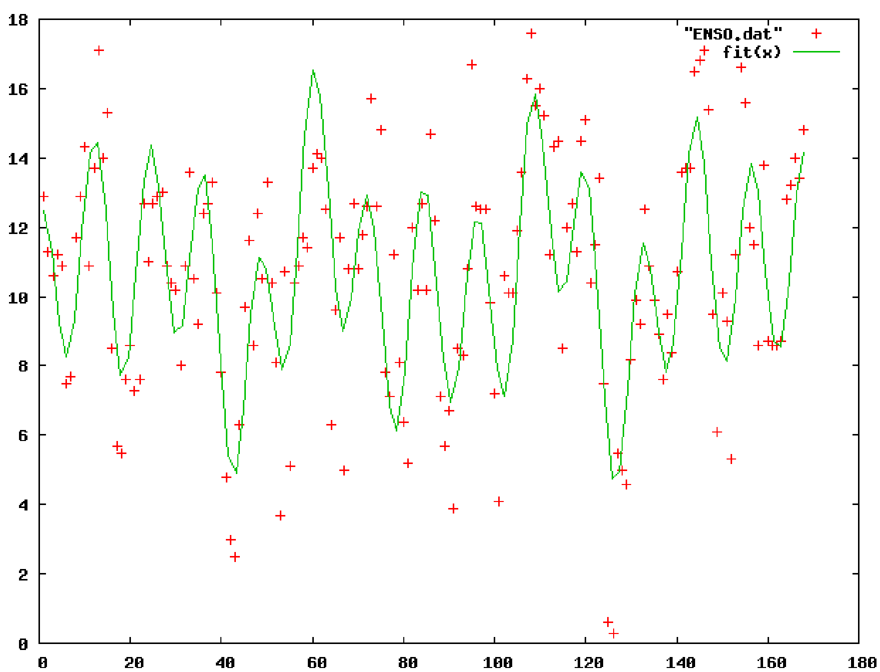

FIGURE 27. Scatter plot of model ENSO

12.13.1. *Model ENSO.gms.* [26]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   The data are monthly averaged atmospheric pressure
               differences between Easter Island and Darwin,
               Australia.  This difference drives the trade winds in
               the southern hemisphere.  Fourier analysis of the data
               reveals 3 significant cycles.  The annual cycle is the
               strongest, but cycles with periods of approximately 44
               and 26 months are also present.  These cycles
               correspond to the El Nino and the Southern Oscillation.
               Arguments to the SIN and COS functions are in radians.

Reference:     Kahaner, D., C. Moler, and S. Nash, (1989).
               Numerical Methods and Software.
               Englewood Cliffs, NJ: Prentice Hall, pp. 441-445.

Data:          1 Response  (y = atmospheric pressure)
               1 Predictor (x = time)
               168 Observations
               Average Level of Difficulty
               Observed Data

Model:         Miscellaneous Class
               9 Parameters (b1 to b9)

               y = b1 + b2*cos( 2*pi*x/12 ) + b3*sin( 2*pi*x/12 )
                      + b5*cos( 2*pi*x/b4 ) + b6*sin( 2*pi*x/b4 )
                      + b8*cos( 2*pi*x/b7 ) + b9*sin( 2*pi*x/b7 )  + e

          Starting values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   11.0        10.0         1.0510749193E+01  1.7488832467E-01
  b2 =    3.0         3.0         3.0762128085E+00  2.4310052139E-01
  b3 =    0.5         0.5         5.3280138227E-01  2.4354686618E-01
  b4 =   40.0        44.0         4.4311088700E+01  9.4408025976E-01
  b5 =   -0.7        -1.5        -1.6231428586E+00  2.8078369611E-01
  b6 =   -1.3         0.5         5.2554493756E-01  4.8073701119E-01
  b7 =   25.0        26.0         2.6887614440E+01  4.1612939130E-01
  b8 =   -0.3        -0.1         2.1232288488E-01  5.1460022911E-01
  b9 =    1.4         1.5         1.4966870418E+00  2.5434468893E-01

Residual Sum of Squares:                   7.8853978668E+02
Residual Standard Deviation:               2.2269642403E+00
Degrees of Freedom:                              159
Number of Observations:                          168

$offtext

*------------------------------------------------------------------------------
* data
*------------------------------------------------------------------------------

set i /i1*i168/;

table data(i,*)

              y            x
```

[26] www.amsterdamoptimization.com/models/regression/ENSO.gms

```
i1      12.90000   1.000000
i2      11.30000   2.000000
i3      10.60000   3.000000
i4      11.20000   4.000000
i5      10.90000   5.000000
i6      7.500000   6.000000
i7      7.700000   7.000000
i8      11.70000   8.000000
i9      12.90000   9.000000
i10     14.30000  10.000000
i11     10.90000   11.00000
i12     13.70000   12.00000
i13     17.10000   13.00000
i14     14.00000   14.00000
i15     15.30000   15.00000
i16     8.500000   16.00000
i17     5.700000   17.00000
i18     5.500000   18.00000
i19     7.600000   19.00000
i20     8.600000   20.00000
i21     7.300000   21.00000
i22     7.600000   22.00000
i23     12.70000   23.00000
i24     11.00000   24.00000
i25     12.70000   25.00000
i26     12.90000   26.00000
i27     13.00000   27.00000
i28     10.90000   28.00000
i29    10.400000   29.00000
i30    10.200000   30.00000
i31     8.000000   31.00000
i32     10.90000   32.00000
i33     13.60000   33.00000
i34    10.500000   34.00000
i35     9.200000   35.00000
i36     12.40000   36.00000
i37     12.70000   37.00000
i38     13.30000   38.00000
i39    10.100000   39.00000
i40     7.800000   40.00000
i41     4.800000   41.00000
i42     3.000000   42.00000
i43     2.500000   43.00000
i44     6.300000   44.00000
i45     9.700000   45.00000
i46     11.60000   46.00000
i47     8.600000   47.00000
i48     12.40000   48.00000
i49    10.500000   49.00000
i50     13.30000   50.00000
i51    10.400000   51.00000
i52     8.100000   52.00000
i53     3.700000   53.00000
i54     10.70000   54.00000
i55     5.100000   55.00000
i56    10.400000   56.00000
i57     10.90000   57.00000
i58     11.70000   58.00000
i59     11.40000   59.00000
i60     13.70000   60.00000
i61     14.10000   61.00000
i62     14.00000   62.00000
i63     12.50000   63.00000
i64     6.300000   64.00000
i65     9.600000   65.00000
i66     11.70000   66.00000
i67     5.000000   67.00000
i68     10.80000   68.00000
i69     12.70000   69.00000
i70     10.80000   70.00000
i71     11.80000   71.00000
i72     12.60000   72.00000
```

```
i73     15.70000   73.00000
i74     12.60000   74.00000
i75     14.80000   75.00000
i76     7.800000   76.00000
i77     7.100000   77.00000
i78     11.20000   78.00000
i79     8.100000   79.00000
i80     6.400000   80.00000
i81     5.200000   81.00000
i82     12.00000   82.00000
i83    10.200000   83.00000
i84     12.70000   84.00000
i85    10.200000   85.00000
i86     14.70000   86.00000
i87     12.20000   87.00000
i88     7.100000   88.00000
i89     5.700000   89.00000
i90     6.700000   90.00000
i91     3.900000   91.00000
i92     8.500000   92.00000
i93     8.300000   93.00000
i94     10.80000   94.00000
i95     16.70000   95.00000
i96     12.60000   96.00000
i97     12.50000   97.00000
i98     12.50000   98.00000
i99     9.800000   99.00000
i100    7.200000  100.00000
i101    4.100000  101.00000
i102    10.60000  102.00000
i103   10.100000  103.00000
i104   10.100000  104.00000
i105    11.90000  105.00000
i106    13.60000   106.0000
i107    16.30000   107.0000
i108    17.60000   108.0000
i109    15.50000   109.0000
i110    16.00000   110.0000
i111    15.20000   111.0000
i112    11.20000   112.0000
i113    14.30000   113.0000
i114    14.50000   114.0000
i115    8.500000   115.0000
i116    12.00000   116.0000
i117    12.70000   117.0000
i118    11.30000   118.0000
i119    14.50000   119.0000
i120    15.10000   120.0000
i121   10.400000   121.0000
i122    11.50000   122.0000
i123    13.40000   123.0000
i124    7.500000   124.0000
i125   0.6000000   125.0000
i126   0.3000000   126.0000
i127    5.500000   127.0000
i128    5.000000   128.0000
i129    4.600000   129.0000
i130    8.200000   130.0000
i131    9.900000   131.0000
i132    9.200000   132.0000
i133    12.50000   133.0000
i134    10.90000   134.0000
i135    9.900000   135.0000
i136    8.900000   136.0000
i137    7.600000   137.0000
i138    9.500000   138.0000
i139    8.400000   139.0000
i140    10.70000   140.0000
i141    13.60000   141.0000
i142    13.70000   142.0000
i143    13.70000   143.0000
i144    16.50000   144.0000
```

```
  i145     16.80000    145.0000
  i146     17.10000    146.0000
  i147     15.40000    147.0000
  i148     9.500000    148.0000
  i149     6.100000    149.0000
  i150    10.100000    150.0000
  i151     9.300000    151.0000
  i152     5.300000    152.0000
  i153     11.20000    153.0000
  i154     16.60000    154.0000
  i155     15.60000    155.0000
  i156     12.00000    156.0000
  i157     11.50000    157.0000
  i158     8.600000    158.0000
  i159     13.80000    159.0000
  i160     8.700000    160.0000
  i161     8.600000    161.0000
  i162     8.600000    162.0000
  i163     8.700000    163.0000
  i164     12.80000    164.0000
  i165     13.20000    165.0000
  i166     14.00000    166.0000
  i167     13.40000    167.0000
  i168     14.80000    168.0000
;


*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');


*
* certified values
*

*
* certified values
*
scalars
  cb1 'certified value for b1' / 1.0510749193E+01/
  cb2 'certified value for b2' / 3.0762128085E+00/
  cb3 'certified value for b3' / 5.3280138227E-01/
  cb4 'certified value for b4' / 4.4311088700E+01/
  cb5 'certified value for b5' /-1.6231428586E+00/
  cb6 'certified value for b6' / 5.2554493756E-01/
  cb7 'certified value for b7' / 2.6887614440E+01/
  cb8 'certified value for b8' / 2.1232288488E-01/
  cb9 'certified value for b8' / 1.4966870418E+00/
  ce1 'certified std err for b1 ' /1.7488832467E-01/
  ce2 'certified std err for b2 ' /2.4310052139E-01/
  ce3 'certified std err for b3 ' /2.4354686618E-01/
  ce4 'certified std err for b4 ' /9.4408025976E-01/
  ce5 'certified std err for b5 ' /2.8078369611E-01/
  ce6 'certified std err for b6 ' /4.8073701119E-01/
  ce7 'certified std err for b7 ' /4.1612939130E-01/
  ce8 'certified std err for b8 ' /5.1460022911E-01/
  ce9 'certified std err for b9 ' /2.5434468893E-01/
;




*----------------------------------------------------------------------------
* statistical model
*----------------------------------------------------------------------------


variables
```

```
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
    b4           'coefficient to estimate'
    b5           'coefficient to estimate'
    b6           'coefficient to estimate'
    b7           'coefficient to estimate'
    b8           'coefficient to estimate'
    b9           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1 + b2*cos( 2*pi*x(i)/12 ) + b3*sin( 2*pi*x(i)/12 )
                      + b5*cos( 2*pi*x(i)/b4 ) + b6*sin( 2*pi*x(i)/b4 )
                      + b8*cos( 2*pi*x(i)/b7 ) + b9*sin( 2*pi*x(i)/b7 );

option nlp=nls;
model nlfit /obj,fit/;

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l =   11.0;
b2.l =    3.0;
b3.l =    0.5;
b4.l =   40.0;
b5.l =   -0.7;
b6.l =   -1.3;
b7.l =   25.0;
b8.l =   -0.3;
b9.l =    1.4;


solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l,b5.l,b6.l,b7.l,b8.l,b9.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4)+abs(b5.l-cb5)+abs(b6.l-cb6)
       +abs(b7.l-cb7)+abs(b8.l-cb8)+abs(b9.l-cb9))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4)+abs(b5.m-ce5)+abs(b6.m-ce6)
       +abs(b7.m-ce7)+abs(b8.m-ce8)+abs(b9.m-ce9))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =  10.0;
b2.l =   3.0;
b3.l =   0.5;
b4.l =  44.0;
b5.l =  -1.5;
b6.l =   0.5;
b7.l =  26.0;
b8.l =  -0.1;
b9.l =   1.5;


solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l,b5.l,b6.l,b7.l,b8.l,b9.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4)+abs(b5.l-cb5)+abs(b6.l-cb6)
       +abs(b7.l-cb7)+abs(b8.l-cb8)+abs(b9.l-cb9))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
```

```
      +abs(b4.m-ce4)+abs(b5.m-ce5)+abs(b6.m-ce6)
      +abs(b7.m-ce7)+abs(b8.m-ce8)+abs(b9.m-ce9))>0.0001) "Accuracy problem";
```

12.14. **Thurber.** The model[23, 37]:

$$y = \frac{\beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3}{1 + \beta_5 x + \beta_6 x^2 + \beta_7 x^3} \tag{54}$$

These data are the result of a NIST study involving semiconductor electron mobility. The response variable is a measure of electron mobility, and the predictor variable is the natural log of the density.

Note that in GAMS we need to circumvent `x**n` in this model as the argument x assumes negative values. Hence we use `sqr(.)` and `power(.)`.

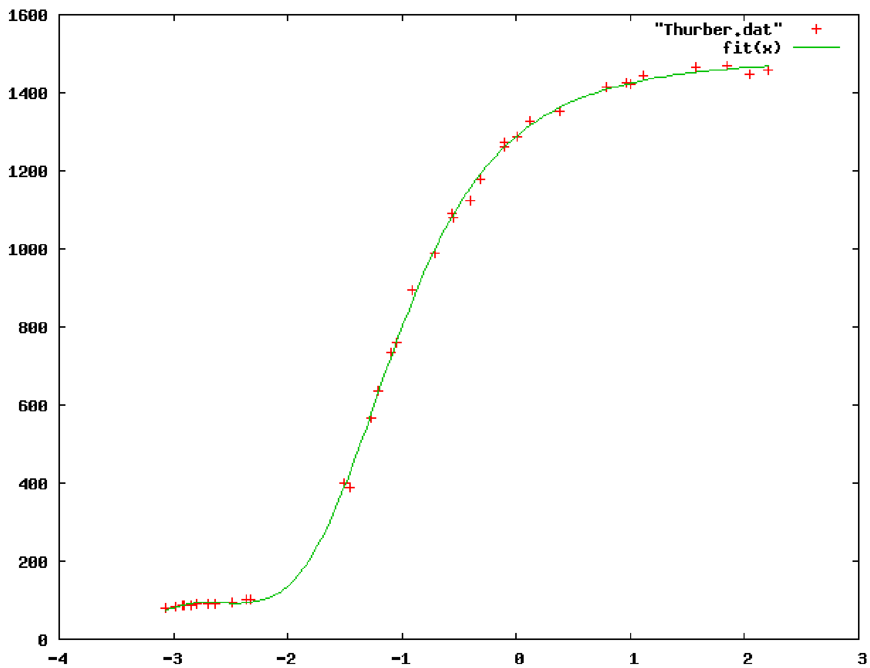

FIGURE 28. Scatter plot of model Thurber

12.14.1. *Model Thurber.gms.* [27]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression
```

[27] www.amsterdamoptimization.com/models/regression/Thurber.gms

```
Description:   These data are the result of a NIST study involving
               semiconductor electron mobility.  The response
               variable is a measure of electron mobility, and the
               predictor variable is the natural log of the density.

Reference:     Thurber, R., NIST (197?).
               Semiconductor electron mobility modeling.

Data:          1 Response Variable  (y = electron mobility)
               1 Predictor Variable (x = log[density])
               37 Observations
               Higher Level of Difficulty
               Observed Data

Model:         Rational Class (cubic/cubic)
               7 Parameters (b1 to b7)

               y = (b1 + b2*x + b3*x**2 + b4*x**3) /
                   (1 + b5*x + b6*x**2 + b7*x**3)  +  e

          Starting Values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   1000        1300          1.2881396800E+03  4.6647963344E+00
  b2 =   1000        1500          1.4910792535E+03  3.9571156086E+01
  b3 =    400         500          5.8323836877E+02  2.8698696102E+01
  b4 =     40          75          7.5416644291E+01  5.5675370270E+00
  b5 =      0.7         1          9.6629502864E-01  3.1333340687E-02
  b6 =      0.3         0.4        3.9797285797E-01  1.4984928198E-02
  b7 =      0.03        0.05       4.9727297349E-02  6.5842344623E-03

Residual Sum of Squares:                    5.6427082397E+03
Residual Standard Deviation:                1.3714600784E+01
Degrees of Freedom:                                30
Number of Observations:                            37

$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i37/;

table data(i,*)
                  y              x
  i1           80.574E0      -3.067E0
  i2           84.248E0      -2.981E0
  i3           87.264E0      -2.921E0
  i4           87.195E0      -2.912E0
  i5           89.076E0      -2.840E0
  i6           89.608E0      -2.797E0
  i7           89.868E0      -2.702E0
  i8           90.101E0      -2.699E0
  i9           92.405E0      -2.633E0
  i10          95.854E0      -2.481E0
  i11         100.696E0      -2.363E0
  i12         101.060E0      -2.322E0
  i13         401.672E0      -1.501E0
  i14         390.724E0      -1.460E0
  i15         567.534E0      -1.274E0
  i16         635.316E0      -1.212E0
  i17         733.054E0      -1.100E0
  i18         759.087E0      -1.046E0
  i19         894.206E0      -0.915E0
  i20         990.785E0      -0.714E0
  i21        1090.109E0      -0.566E0
  i22        1080.914E0      -0.545E0
  i23        1122.643E0      -0.400E0
  i24        1178.351E0      -0.309E0
  i25        1260.531E0      -0.109E0
  i26        1273.514E0      -0.103E0
```

```
  i27        1288.339E0       0.010E0
  i28        1327.543E0       0.119E0
  i29        1353.863E0       0.377E0
  i30        1414.509E0       0.790E0
  i31        1425.208E0       0.963E0
  i32        1421.384E0       1.006E0
  i33        1442.962E0       1.115E0
  i34        1464.350E0       1.572E0
  i35        1468.705E0       1.841E0
  i36        1447.894E0       2.047E0
  i37        1457.628E0       2.200E0

;


*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' / 1.2881396800E+03 /
  cb2 'certified value for b2' / 1.4910792535E+03 /
  cb3 'certified value for b3' / 5.8323836877E+02 /
  cb4 'certified value for b4' / 7.5416644291E+01 /
  cb5 'certified value for b5' / 9.6629502864E-01 /
  cb6 'certified value for b6' / 3.9797285797E-01 /
  cb7 'certified value for b7' / 4.9727297349E-02 /
  ce1 'certified std err for b1 ' / 4.6647963344E+00 /
  ce2 'certified std err for b2 ' / 3.9571156086E+01 /
  ce3 'certified std err for b3 ' / 2.8698696102E+01 /
  ce4 'certified std err for b4 ' / 5.5675370270E+00 /
  ce5 'certified std err for b5 ' / 3.1333340687E-02 /
  ce6 'certified std err for b6 ' / 1.4984928198E-02 /
  ce7 'certified std err for b7 ' / 6.5842344623E-03 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse         'sum of squared errors'
    b1          'coefficient to estimate'
    b2          'coefficient to estimate'
    b3          'coefficient to estimate'
    b4          'coefficient to estimate'
    b5          'coefficient to estimate'
    b6          'coefficient to estimate'
    b7          'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= (b1 + b2*x(i) + b3*sqr(x(i)) + b4*power(x(i),3)) /
                   (1 + b5*x(i) + b6*sqr(x(i)) + b7*power(x(i),3));

option nlp=nls;
model nlfit /obj,fit/;

*-----------------------------------------------------------------------------
* first set of initial values
```

```
*-----------------------------------------------------------------------------

b1.l =  1000;
b2.l =  1000;
b3.l =   400;
b4.l =    40;
b5.l =     0.7;
b6.l =     0.3;
b7.l =     0.03;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l,b5.l,b6.l,b7.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4)+abs(b5.l-cb5)+abs(b6.l-cb6)
       +abs(b7.l-cb7))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4)+abs(b5.m-ce5)+abs(b6.m-ce6)
       +abs(b7.m-ce7))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =  1300;
b2.l =  1500;
b3.l =   500;
b4.l =    75;
b5.l =     1;
b6.l =     0.4;
b7.l =     0.05;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l,b5.l,b6.l,b7.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4)+abs(b5.l-cb5)+abs(b6.l-cb6)
       +abs(b7.l-cb7))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4)+abs(b5.m-ce5)+abs(b6.m-ce6)
       +abs(b7.m-ce7))>0.0001) "Accuracy problem";
```

12.15. **BoxBOD.** The model is[23, 4]:

$$y = \beta_1(1 - exp[-\beta_2 x]) + \varepsilon \tag{55}$$

The data are described in detail in [4]. The response variable is biochemical oxygen demand (BOD) in mg/l, and the predictor variable is incubation time in days.

The first instance has a poor starting point. We therefore formulate the model in NLP format and use CONOPT to find a good solution. We also help CONOPT by adding a bound $\beta_2 \geq 0$ in order to protect the exp(.) function from overflowing. We then restart from the CONOPT solution with the NLS solver. NLS will ignore the bound (it will give a warning about the bound).

12.15.1. *Model BoxBOD.gms.* [28]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml
```

[28] www.amsterdamoptimization.com/models/regression/BoxBOD.gms

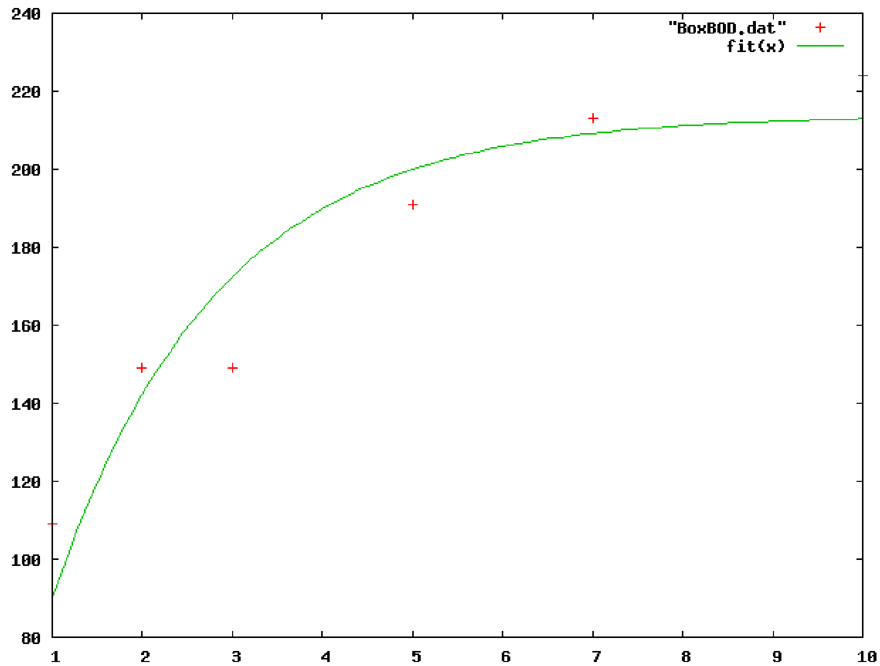

FIGURE 29. Scatter plot of model BoxBOD

```
--------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   These data are described in detail in Box, Hunter and
               Hunter (1978).  The response variable is biochemical
               oxygen demand (BOD) in mg/l, and the predictor
               variable is incubation time in days.

Reference:     Box, G. P., W. G. Hunter, and J. S. Hunter (1978).
               Statistics for Experimenters.
               New York, NY: Wiley, pp. 483-487.

Data:          1 Response  (y = biochemical oxygen demand)
               1 Predictor (x = incubation time)
               6 Observations
               Higher Level of Difficulty
               Observed Data

Model:         Exponential Class
               2 Parameters (b1 and b2)

               y = b1*(1-exp[-b2*x])  +  e

          Starting values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =    1           100          2.1380940889E+02  1.2354515176E+01
  b2 =    1             0.75       5.4723748542E-01  1.0455993237E-01

Residual Sum of Squares:                    1.1680088766E+03
Residual Standard Deviation:                1.7088072423E+01
Degrees of Freedom:                                4
Number of Observations:                            6
```

```
$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i6/;

table data(i,*)
           y     x
  i1     109     1
  i2     149     2
  i3     149     3
  i4     191     5
  i5     213     7
  i6     224    10
;


*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' / 2.1380940889E+02 /
  cb2 'certified value for b2' / 5.4723748542E-01 /
  ce1 'certified std err for b1 ' / 1.2354515176E+01 /
  ce2 'certified std err for b2 ' / 1.0455993237E-01 /
;


*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    r(i)         'residual in model 2'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =e= sum(i, sqr(r(i)));
fit(i)..  r(i) =e= y(i) - b1*(1-exp[-b2*x(i)]);

model nlfit "NLP format" /obj,fit/;

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

*
* Does not converge out of the box
* Use conopt to find a good starting point.
*

b1.l = 1;
b2.l = 1;

* to help conopt:
```

```
b2.lo = 0;
* otherwise we may get with conopt2:
*     *** ERRORS/WARNINGS IN EQUATION fit2(i6)
*        1 error(s): exp: FUNC OVERFLOW: x too large


* step 1: get close using conopt + bound on b2
option nlp=conopt;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;
* step 2: finish with NLS + ignore bound on b2
option nlp=nls;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =  100;
b2.l =    0.75;

option nlp=nls;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2))>0.0001) "Accuracy problem";
```

12.16. **Rat.** These are two NIST[23] models from [33].

12.16.1. *Rat42.* The model is:

$$y = \frac{\beta_1}{1 + exp[\beta_2 - \beta_3 x]} + \varepsilon \tag{56}$$

This model and data are an example of fitting sigmoidal growth curves taken from [33]. The response variable is pasture yield, and the predictor variable is growing time.

12.16.2. *Model Rat42.gms.* [29]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   This model and data are an example of fitting
               sigmoidal growth curves taken from Ratkowsky (1983).
               The response variable is pasture yield, and the
               predictor variable is growing time.

Reference:     Ratkowsky, D.A. (1983).
               Nonlinear Regression Modeling.
               New York, NY:  Marcel Dekker, pp. 61 and 88.

Data:          1 Response   (y = pasture yield)
```

[29] www.amsterdamoptimization.com/models/regression/Rat42.gms

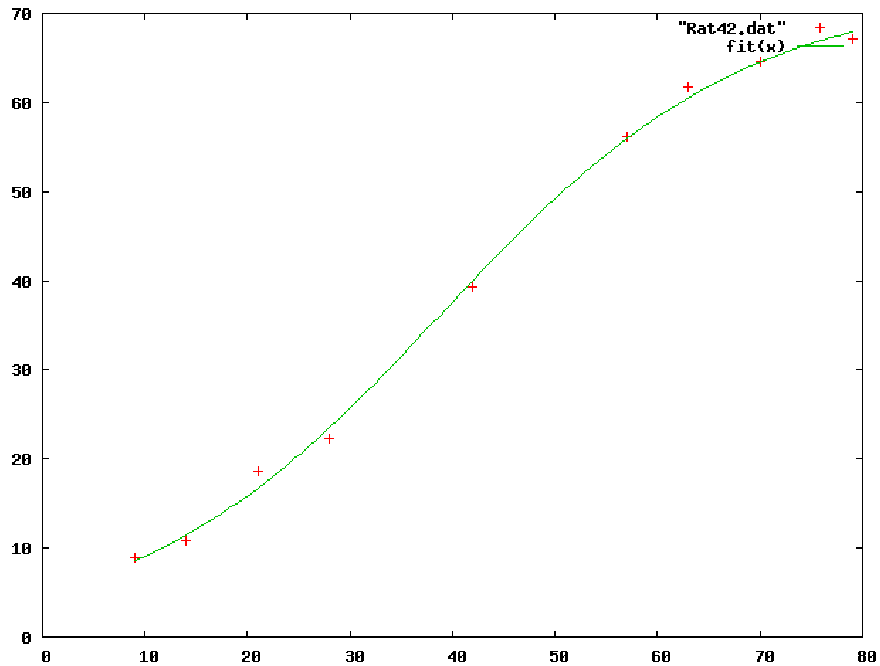

FIGURE 30. Scatter plot of model Rat42

```
               1 Predictor (x = growing time)
               9 Observations
               Higher Level of Difficulty
               Observed Data

Model:         Exponential Class
               3 Parameters (b1 to b3)

               y = b1 / (1+exp[b2-b3*x])  +  e

          Starting Values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   100          75           7.2462237576E+01  1.7340283401E+00
  b2 =     1           2.5         2.6180768402E+00  8.8295217536E-02
  b3 =     0.1         0.07        6.7359200066E-02  3.4465663377E-03

Residual Sum of Squares:                    8.0565229338E+00
Residual Standard Deviation:                1.1587725499E+00
Degrees of Freedom:                                6
Number of Observations:                            9

$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i9/;

table data(i,*)
                  y               x
  i1          8.930E0         9.000E0
  i2         10.800E0        14.000E0
  i3         18.590E0        21.000E0
```

```
  i4         22.330E0       28.000E0
  i5         39.350E0       42.000E0
  i6         56.110E0       57.000E0
  i7         61.730E0       63.000E0
  i8         64.620E0       70.000E0
  i9         67.080E0       79.000E0

;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' / 7.2462237576E+01 /
  cb2 'certified value for b2' / 2.6180768402E+00 /
  cb3 'certified value for b3' / 6.7359200066E-02 /
  ce1 'certified std err for b1 ' / 1.7340283401E+00 /
  ce2 'certified std err for b2 ' / 8.8295217536E-02 /
  ce3 'certified std err for b3 ' / 3.4465663377E-03 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1 / (1+exp[b2-b3*x(i)]);

option nlp=nls;
model nlfit /obj,fit/;

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l =  100;
b2.l =    1;
b3.l =    0.1;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =  75;
b2.l =   2.5;
b3.l =   0.07;
```

```
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";
```

12.16.3. *Rat43.* The model is:

$$y = \frac{\beta_1}{(1+\exp[\beta_2 - \beta_3 x])^{1/\beta_4}} + \varepsilon \tag{57}$$

This model and data are an example of fitting sigmoidal growth curves taken from [33]. The response variable is the dry weight of onion bulbs and tops, and the predictor variable is growing time.

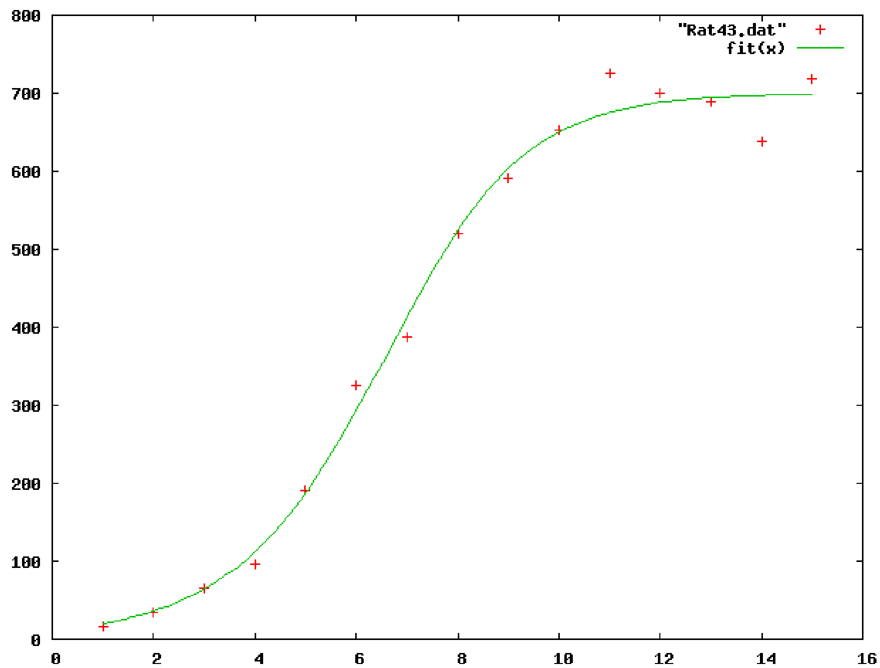

FIGURE 31. Scatter plot of model Rat43

12.16.4. *Model Rat43.gms.* [30]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml
---------------------------------------------------------------------------

Procedure:     Nonlinear Least Squares Regression
```

[30] www.amsterdamoptimization.com/models/regression/Rat43.gms

```
Description:   This model and data are an example of fitting
               sigmoidal growth curves taken from Ratkowsky (1983).
               The response variable is the dry weight of onion bulbs
               and tops, and the predictor variable is growing time.

Reference:     Ratkowsky, D.A. (1983).
               Nonlinear Regression Modeling.
               New York, NY:  Marcel Dekker, pp. 62 and 88.

Data:          1 Response  (y = onion bulb dry weight)
               1 Predictor (x = growing time)
               15 Observations
               Higher Level of Difficulty
               Observed Data

Model:         Exponential Class
               4 Parameters (b1 to b4)

               y = b1 / ((1+exp[b2-b3*x])**(1/b4))  +  e

          Starting Values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   100         700           6.9964151270E+02  1.6302297817E+01
  b2 =    10           5           5.2771253025E+00  2.0828735829E+00
  b3 =     1           0.75        7.5962938329E-01  1.9566123451E-01
  b4 =     1           1.3         1.2792483859E+00  6.8761936385E-01

Residual Sum of Squares:                    8.7864049080E+03
Residual Standard Deviation:                2.8262414662E+01
Degrees of Freedom:                                9
Number of Observations:                           15

$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i15/;

table data(i,*)
                  y         x
  i1         16.08E0     1.0E0
  i2         33.83E0     2.0E0
  i3         65.80E0     3.0E0
  i4         97.20E0     4.0E0
  i5        191.55E0     5.0E0
  i6        326.20E0     6.0E0
  i7        386.87E0     7.0E0
  i8        520.53E0     8.0E0
  i9        590.03E0     9.0E0
  i10       651.92E0    10.0E0
  i11       724.93E0    11.0E0
  i12       699.56E0    12.0E0
  i13       689.96E0    13.0E0
  i14       637.56E0    14.0E0
  i15       717.41E0    15.0E0

;



*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
```

```
* certified values
*
scalars
  cb1 'certified value for b1' / 6.9964151270E+02 /
  cb2 'certified value for b2' / 5.2771253025E+00 /
  cb3 'certified value for b3' / 7.5962938329E-01 /
  cb4 'certified value for b4' / 1.2792483859E+00 /
  ce1 'certified std err for b1 ' / 1.6302297817E+01 /
  ce2 'certified std err for b2 ' / 2.0828735829E+00 /
  ce3 'certified std err for b3 ' / 1.9566123451E-01 /
  ce4 'certified std err for b4 ' / 6.8761936385E-01 /
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
    b4           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1 / ((1+exp[b2-b3*x(i)])**(1/b4)) ;

option nlp=nls;
model nlfit /obj,fit/;

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

b1.l =  100;
b2.l =   10;
b3.l =    1;
b4.l =    1;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =   700;
b2.l =     5;
b3.l =     0.75;
b4.l =     1.3;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l,b4.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3)
       +abs(b4.l-cb4))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3)
       +abs(b4.m-ce4))>0.0001) "Accuracy problem";
```

12.17. **Eckerle4.** The model is[23, 8]:

$$y = \frac{\beta_1}{\beta_2} \exp\left[-0.5\left(\frac{x-\beta_3}{\beta_2}\right)^2\right] + \varepsilon \tag{58}$$

These data are the result of a NIST study involving circular interference transmittance. The response variable is transmittance, and the predictor variable is wavelength.

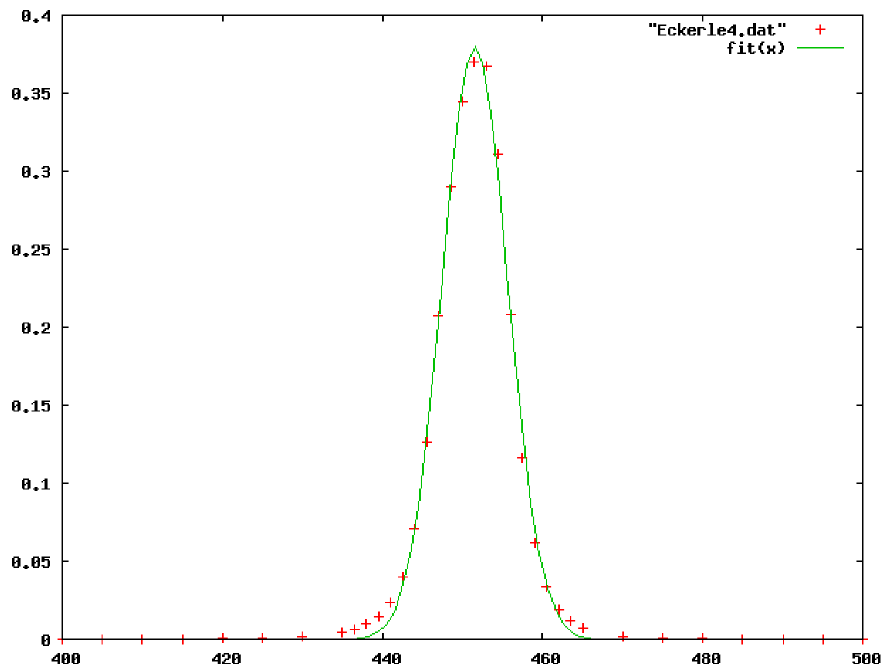

FIGURE 32. Scatter plot of model Eckerle4

12.17.1. *Model Eckerle4.gms.* [31]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml
--------------------------------------------------------------------------

Procedure:      Nonlinear Least Squares Regression

Description:    These data are the result of a NIST study involving
                circular interference transmittance.  The response
                variable is transmittance, and the predictor variable
                is wavelength.

Reference:      Eckerle, K., NIST (197?).
```

[31] www.amsterdamoptimization.com/models/regression/Eckerle4.gms

```
               Circular Interference Transmittance Study.

Data:          1 Response Variable  (y = transmittance)
               1 Predictor Variable (x = wavelength)
               35 Observations
               Higher Level of Difficulty
               Observed Data

Model:         Exponential Class
               3 Parameters (b1 to b3)

               y = (b1/b2) * exp[-0.5*((x-b3)/b2)**2]  +  e

          Starting values                 Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =     1           1.5         1.5543827178E+00  1.5408051163E-02
  b2 =    10           5           4.0888321754E+00  4.6803020753E-02
  b3 =   500         450           4.5154121844E+02  4.6800518816E-02

Residual Sum of Squares:                   1.4635887487E-03
Residual Standard Deviation:               6.7629245447E-03
Degrees of Freedom:                               32
Number of Observations:                           35

$offtext


*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i35/;

table data(i,*)
                 y               x
   i1       0.0001575E0    400.000000E0
   i2       0.0001699E0    405.000000E0
   i3       0.0002350E0    410.000000E0
   i4       0.0003102E0    415.000000E0
   i5       0.0004917E0    420.000000E0
   i6       0.0008710E0    425.000000E0
   i7       0.0017418E0    430.000000E0
   i8       0.0046400E0    435.000000E0
   i9       0.0065895E0    436.500000E0
   i10      0.0097302E0    438.000000E0
   i11      0.0149002E0    439.500000E0
   i12      0.0237310E0    441.000000E0
   i13      0.0401683E0    442.500000E0
   i14      0.0712559E0    444.000000E0
   i15      0.1264458E0    445.500000E0
   i16      0.2073413E0    447.000000E0
   i17      0.2902366E0    448.500000E0
   i18      0.3445623E0    450.000000E0
   i19      0.3698049E0    451.500000E0
   i20      0.3668534E0    453.000000E0
   i21      0.3106727E0    454.500000E0
   i22      0.2078154E0    456.000000E0
   i23      0.1164354E0    457.500000E0
   i24      0.0616764E0    459.000000E0
   i25      0.0337200E0    460.500000E0
   i26      0.0194023E0    462.000000E0
   i27      0.0117831E0    463.500000E0
   i28      0.0074357E0    465.000000E0
   i29      0.0022732E0    470.000000E0
   i30      0.0008800E0    475.000000E0
   i31      0.0004579E0    480.000000E0
   i32      0.0002345E0    485.000000E0
   i33      0.0001586E0    490.000000E0
   i34      0.0001143E0    495.000000E0
   i35      0.0000710E0    500.000000E0
;
```

```
*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' / 1.5543827178E+00 /
  cb2 'certified value for b2' / 4.0888321754E+00 /
  cb3 'certified value for b3' / 4.5154121844E+02 /
  ce1 'certified std err for b1 ' / 1.5408051163E-02 /
  ce2 'certified std err for b2 ' / 4.6803020753E-02 /
  ce3 'certified std err for b3 ' / 4.6800518816E-02 /
;


*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= (b1/b2) * exp[-0.5*sqr((x(i)-b3)/b2)];

option nlp=nls;
model nlfit /obj,fit/;

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------


b1.l =   1;
b2.l =  10;
b3.l = 500;


solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =     1.5;
b2.l =     5;
b3.l =   450;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l,b3.l;
```

```
abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";
```

12.18. **Bennett5.** The model is[23, 3]:

$$y = \beta_1(\beta_2 + x)^{-1/\beta_3} + \varepsilon \tag{59}$$

These data are the result of a NIST study involving superconductivity magnetization modeling. The response variable is strength of the magnetic field, and the predictor variable is the log of time in minutes.

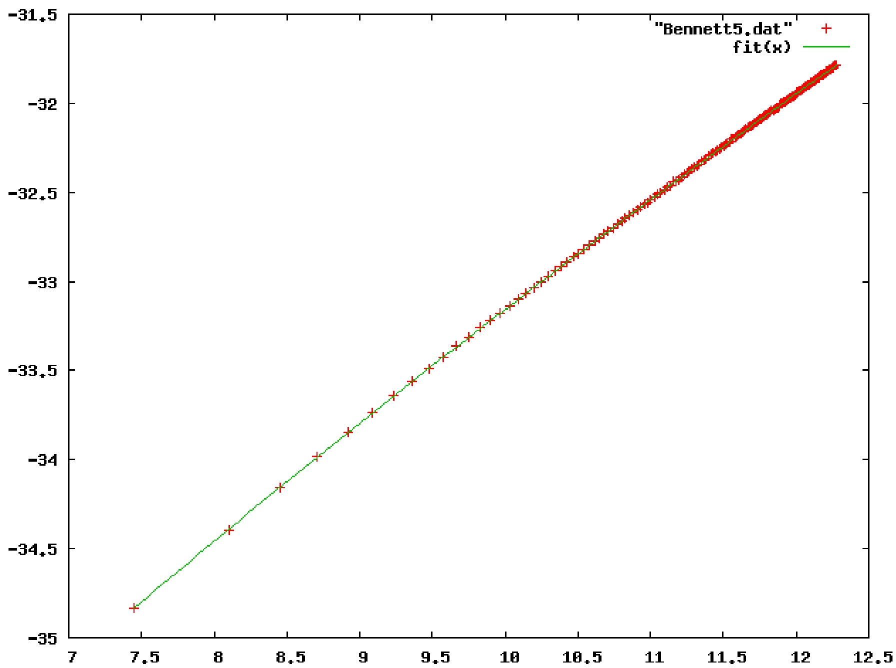

FIGURE 33. Scatter plot of model Bennett5

12.18.1. *Model Bennett5.gms.* [32]

```
$ontext

  Nonlinear Least Squares Regression example

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

--------------------------------------------------------------------------
Procedure:     Nonlinear Least Squares Regression

Description:   These data are the result of a NIST study involving
               superconductivity magnetization modeling.  The
               response variable is magnetism, and the predictor
```

[32] www.amsterdamoptimization.com/models/regression/Bennett5.gms

```
               variable is the log of time in minutes.

Reference:     Bennett, L., L. Swartzendruber, and H. Brown,
               NIST (1994).
               Superconductivity Magnetization Modeling.

Data:          1 Response Variable  (y = magnetism)
               1 Predictor Variable (x = log[time])
               154 Observations
               Higher Level of Difficulty
               Observed Data

Model:         Miscellaneous Class
               3 Parameters (b1 to b3)

               y = b1 * (b2+x)**(-1/b3)  +  e

          Starting values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   -2000       -1500        -2.5235058043E+03  2.9715175411E+02
  b2 =      50          45         4.6736564644E+01  1.2448871856E+00
  b3 =       0.8         0.85      9.3218483193E-01  2.0272299378E-02

Residual Sum of Squares:                    5.2404744073E-04
Residual Standard Deviation:                1.8629312528E-03
Degrees of Freedom:                               151
Number of Observations:                           154

$offtext

*-----------------------------------------------------------------------------
* data
*-----------------------------------------------------------------------------

set i /i1*i154/;

table data(i,*)
               y               x
  i1       -34.834702E0      7.447168E0
  i2       -34.393200E0      8.102586E0
  i3       -34.152901E0      8.452547E0
  i4       -33.979099E0      8.711278E0
  i5       -33.845901E0      8.916774E0
  i6       -33.732899E0      9.087155E0
  i7       -33.640301E0      9.232590E0
  i8       -33.559200E0      9.359535E0
  i9       -33.486801E0      9.472166E0
  i10      -33.423100E0      9.573384E0
  i11      -33.365101E0      9.665293E0
  i12      -33.313000E0      9.749461E0
  i13      -33.260899E0      9.827092E0
  i14      -33.217400E0      9.899128E0
  i15      -33.176899E0      9.966321E0
  i16      -33.139198E0     10.029280E0
  i17      -33.101601E0     10.088510E0
  i18      -33.066799E0     10.144430E0
  i19      -33.035000E0     10.197380E0
  i20      -33.003101E0     10.247670E0
  i21      -32.971298E0     10.295560E0
  i22      -32.942299E0     10.341250E0
  i23      -32.916302E0     10.384950E0
  i24      -32.890202E0     10.426820E0
  i25      -32.864101E0     10.467000E0
  i26      -32.841000E0     10.505640E0
  i27      -32.817799E0     10.542830E0
  i28      -32.797501E0     10.578690E0
  i29      -32.774300E0     10.613310E0
  i30      -32.757000E0     10.646780E0
  i31      -32.733799E0     10.679150E0
  i32      -32.716400E0     10.710520E0
  i33      -32.699100E0     10.740920E0
```

```
i34     -32.678799E0    10.770440E0
i35     -32.661400E0    10.799100E0
i36     -32.644001E0    10.826970E0
i37     -32.626701E0    10.854080E0
i38     -32.612202E0    10.880470E0
i39     -32.597698E0    10.906190E0
i40     -32.583199E0    10.931260E0
i41     -32.568699E0    10.955720E0
i42     -32.554298E0    10.979590E0
i43     -32.539799E0    11.002910E0
i44     -32.525299E0    11.025700E0
i45     -32.510799E0    11.047980E0
i46     -32.499199E0    11.069770E0
i47     -32.487598E0    11.091100E0
i48     -32.473202E0    11.111980E0
i49     -32.461601E0    11.132440E0
i50     -32.435501E0    11.152480E0
i51     -32.435501E0    11.172130E0
i52     -32.426800E0    11.191410E0
i53     -32.412300E0    11.210310E0
i54     -32.400799E0    11.228870E0
i55     -32.392101E0    11.247090E0
i56     -32.380501E0    11.264980E0
i57     -32.366001E0    11.282560E0
i58     -32.357300E0    11.299840E0
i59     -32.348598E0    11.316820E0
i60     -32.339901E0    11.333520E0
i61     -32.328400E0    11.349940E0
i62     -32.319698E0    11.366100E0
i63     -32.311001E0    11.382000E0
i64     -32.299400E0    11.397660E0
i65     -32.290699E0    11.413070E0
i66     -32.282001E0    11.428240E0
i67     -32.273300E0    11.443200E0
i68     -32.264599E0    11.457930E0
i69     -32.256001E0    11.472440E0
i70     -32.247299E0    11.486750E0
i71     -32.238602E0    11.500860E0
i72     -32.229900E0    11.514770E0
i73     -32.224098E0    11.528490E0
i74     -32.215401E0    11.542020E0
i75     -32.203800E0    11.555380E0
i76     -32.198002E0    11.568550E0
i77     -32.189400E0    11.581560E0
i78     -32.183601E0    11.594420E0
i79     -32.174900E0    11.607121E0
i80     -32.169102E0    11.619640E0
i81     -32.163300E0    11.632000E0
i82     -32.154598E0    11.644210E0
i83     -32.145901E0    11.656280E0
i84     -32.140099E0    11.668200E0
i85     -32.131401E0    11.679980E0
i86     -32.125599E0    11.691620E0
i87     -32.119801E0    11.703130E0
i88     -32.111198E0    11.714510E0
i89     -32.105400E0    11.725760E0
i90     -32.096699E0    11.736880E0
i91     -32.090900E0    11.747890E0
i92     -32.088001E0    11.758780E0
i93     -32.079300E0    11.769550E0
i94     -32.073502E0    11.780200E0
i95     -32.067699E0    11.790730E0
i96     -32.061901E0    11.801160E0
i97     -32.056099E0    11.811480E0
i98     -32.050301E0    11.821700E0
i99     -32.044498E0    11.831810E0
i100    -32.038799E0    11.841820E0
i101    -32.033001E0    11.851730E0
i102    -32.027199E0    11.861550E0
i103    -32.024300E0    11.871270E0
i104    -32.018501E0    11.880890E0
i105    -32.012699E0    11.890420E0
```

```
  i106     -32.004002E0     11.899870E0
  i107     -32.001099E0     11.909220E0
  i108     -31.995300E0     11.918490E0
  i109     -31.989500E0     11.927680E0
  i110     -31.983700E0     11.936780E0
  i111     -31.977900E0     11.945790E0
  i112     -31.972099E0     11.954730E0
  i113     -31.969299E0     11.963590E0
  i114     -31.963501E0     11.972370E0
  i115     -31.957701E0     11.981070E0
  i116     -31.951900E0     11.989700E0
  i117     -31.946100E0     11.998260E0
  i118     -31.940300E0     12.006740E0
  i119     -31.937401E0     12.015150E0
  i120     -31.931601E0     12.023490E0
  i121     -31.925800E0     12.031760E0
  i122     -31.922899E0     12.039970E0
  i123     -31.917101E0     12.048100E0
  i124     -31.911301E0     12.056170E0
  i125     -31.908400E0     12.064180E0
  i126     -31.902599E0     12.072120E0
  i127     -31.896900E0     12.080010E0
  i128     -31.893999E0     12.087820E0
  i129     -31.888201E0     12.095580E0
  i130     -31.885300E0     12.103280E0
  i131     -31.882401E0     12.110920E0
  i132     -31.876600E0     12.118500E0
  i133     -31.873699E0     12.126030E0
  i134     -31.867901E0     12.133500E0
  i135     -31.862101E0     12.140910E0
  i136     -31.859200E0     12.148270E0
  i137     -31.856300E0     12.155570E0
  i138     -31.850500E0     12.162830E0
  i139     -31.844700E0     12.170030E0
  i140     -31.841801E0     12.177170E0
  i141     -31.838900E0     12.184270E0
  i142     -31.833099E0     12.191320E0
  i143     -31.830200E0     12.198320E0
  i144     -31.827299E0     12.205270E0
  i145     -31.821600E0     12.212170E0
  i146     -31.818701E0     12.219030E0
  i147     -31.812901E0     12.225840E0
  i148     -31.809999E0     12.232600E0
  i149     -31.807100E0     12.239320E0
  i150     -31.801300E0     12.245990E0
  i151     -31.798401E0     12.252620E0
  i152     -31.795500E0     12.259200E0
  i153     -31.789700E0     12.265750E0
  i154     -31.786800E0     12.272240E0
;


*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');

*
* certified values
*
scalars
  cb1 'certified value for b1' / -2.5235058043E+03 /
  cb2 'certified value for b2' /  4.6736564644E+01 /
  cb3 'certified value for b3' /  9.3218483193E-01 /
  ce1 'certified std err for b1 ' / 2.9715175411E+02 /
  ce2 'certified std err for b2 ' / 1.2448871856E+00 /
  ce3 'certified std err for b3 ' / 2.0272299378E-02 /
;
```

```
*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse         'sum of squared errors'
    b1          'coefficient to estimate'
    b2          'coefficient to estimate'
    b3          'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= b1 * (b2+x(i))**(-1/b3) ;

*-----------------------------------------------------------------------------
* first set of initial values
*-----------------------------------------------------------------------------

* We need to extend the max number of function evaluation calls.

$onecho > nls.opt
* This option is needed in Bennett5.gms
mxfcal 1000
$offecho


b1.l =   -2000;
b2.l =      50;
b3.l =       0.8;

option nlp=nls;
model nlfit /obj,fit/;
nlfit.optfile=1;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";

*-----------------------------------------------------------------------------
* second set of initial values
*-----------------------------------------------------------------------------

b1.l =   -1500;
b2.l =      45;
b3.l =       0.85;

solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;

abort$((abs(b1.l-cb1)+abs(b2.l-cb2)+abs(b3.l-cb3))>0.0001) "Accuracy problem";
abort$((abs(b1.m-ce1)+abs(b2.m-ce2)+abs(b3.m-ce3))>0.0001) "Accuracy problem";
```

12.19. **GDX file loading.** These examples will show to import the variance-covariance matrix and the confidence intervals from the GDX file `nls.gdx`.

12.19.1. *Loading the variance-covariance matrix.* In this example we show how the Variance-Covariance matrix can be loaded from the GDX file `nls.gdx` generated by the NLS solver. It is often a good idea to inspect the gdx file and see how NLS stored the covariance matrix. In this case the labels are the names of the variables in the optimization model: $b1$, $b2$ and $b3$. These names are added as elements of the set v before we can read the covariance matrix from the GDX file.

covar: variance-covariance matrix

Plane Index (empty)

	b1	b2	b3
b1	0.00146714178034175	2.1542343827303E-5	-5.50877534591619E-5
b2	2.1542343827303E-5	4.43843827415335E-7	-9.80856273607653E-7
b3	-5.50877534591619E-5	-9.80856273607653E-7	2.34219602004083E-6

FIGURE 34. Covariance matrix stored in nls.gdx

12.19.2. *Model Chwirut2cov.gms.* [33]

```
$ontext

  Nonlinear Least Squares Regression example
  Read variance-covariance matrix from GDX file

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------

$offtext

*-------------------------------------------------------------------------------
* data
*-------------------------------------------------------------------------------

set i /i1*i54/;

table data(i,*)
              y            x
  i1     92.9000E0     0.500E0
  i2     57.1000E0     1.000E0
  i3     31.0500E0     1.750E0
  i4     11.5875E0     3.750E0
  i5      8.0250E0     5.750E0
  i6     63.6000E0     0.875E0
  i7     21.4000E0     2.250E0
  i8     14.2500E0     3.250E0
  i9      8.4750E0     5.250E0
  i10    63.8000E0     0.750E0
  i11    26.8000E0     1.750E0
  i12    16.4625E0     2.750E0
  i13     7.1250E0     4.750E0
  i14    67.3000E0     0.625E0
  i15    41.0000E0     1.250E0
  i16    21.1500E0     2.250E0
  i17     8.1750E0     4.250E0
  i18    81.5000E0      .500E0
  i19    13.1200E0     3.000E0
  i20    59.9000E0      .750E0
  i21    14.6200E0     3.000E0
  i22    32.9000E0     1.500E0
  i23     5.4400E0     6.000E0
  i24    12.5600E0     3.000E0
```

[33] www.amsterdamoptimization.com/models/regression/Chwirut2cov.gms

```
  i25     5.4400E0     6.000E0
  i26    32.0000E0     1.500E0
  i27    13.9500E0     3.000E0
  i28    75.8000E0      .500E0
  i29    20.0000E0     2.000E0
  i30    10.4200E0     4.000E0
  i31    59.5000E0      .750E0
  i32    21.6700E0     2.000E0
  i33     8.5500E0     5.000E0
  i34    62.0000E0      .750E0
  i35    20.2000E0     2.250E0
  i36     7.7600E0     3.750E0
  i37     3.7500E0     5.750E0
  i38    11.8100E0     3.000E0
  i39    54.7000E0      .750E0
  i40    23.7000E0     2.500E0
  i41    11.5500E0     4.000E0
  i42    61.3000E0      .750E0
  i43    17.7000E0     2.500E0
  i44     8.7400E0     4.000E0
  i45    59.2000E0      .750E0
  i46    16.3000E0     2.500E0
  i47     8.6200E0     4.000E0
  i48    81.0000E0      .500E0
  i49     4.8700E0     6.000E0
  i50    14.6200E0     3.000E0
  i51    81.7000E0      .500E0
  i52    17.1700E0     2.750E0
  i53    81.3000E0      .500E0
  i54    28.9000E0     1.750E0
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');


*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= exp(-b1*x(i))/(b2+b3*x(i));

*-----------------------------------------------------------------------------
* initial values
*-----------------------------------------------------------------------------

b1.l = 0.1;
b2.l = 0.01;
b3.l = 0.02;

option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;
```

```
*-----------------------------------------------------------------------------
* load and display covariance matrix
*-----------------------------------------------------------------------------

set v 'variable names' /b1,b2,b3/;
parameter covar(v,v);

execute_load 'nls.gdx',covar;
display covar;
```

confint: confidence intervals

Plane Index (empty)

		LO	UP
90%	b1	0.102408	0.230746
	b2	0.004049	0.006281
	b3	0.009586	0.014714
95%	b1	0.089680	0.243474
	b2	0.003828	0.006503
	b3	0.009078	0.015222
97.5%	b1	0.078115	0.255039
	b2	0.003627	0.006704
	b3	0.008615	0.015685
99%	b1	0.064088	0.269066
	b2	0.003383	0.006948
	b3	0.008055	0.016245

FIGURE 35. Confidence intervals stored in nls.gdx

12.19.3. *Loading confidence intervals stored in nls.gdx.* In this example we show how the confidence intervals for the estimates can be loaded from the GDX file `nls.gdx` generated by the NLS solver. It is often a good idea to inspect the gdx file and see how NLS stores the confidence intervals. In this case the labels are the names of the variables in the optimization model: $b1$, $b2$ and $b3$. These names are added as elements of the set v before we can read the confidence intervals from the GDX file.

12.19.4. *Model Chwirut2confint.gms.* [34]

[34] `www.amsterdamoptimization.com/models/regression/Chwirut2confint.gms`

```
$ontext

  Nonlinear Least Squares Regression example
  Read variance-covariance matrix from GDX file

  Erwin Kalvelagen, nov 2007

  Reference:
       http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml

---------------------------------------------------------------------------

$offtext

*-------------------------------------------------------------------------------
* data
*-------------------------------------------------------------------------------

set i /i1*i54/;

table data(i,*)
              y            x
  i1     92.9000E0     0.500E0
  i2     57.1000E0     1.000E0
  i3     31.0500E0     1.750E0
  i4     11.5875E0     3.750E0
  i5      8.0250E0     5.750E0
  i6     63.6000E0     0.875E0
  i7     21.4000E0     2.250E0
  i8     14.2500E0     3.250E0
  i9      8.4750E0     5.250E0
  i10    63.8000E0     0.750E0
  i11    26.8000E0     1.750E0
  i12    16.4625E0     2.750E0
  i13     7.1250E0     4.750E0
  i14    67.3000E0     0.625E0
  i15    41.0000E0     1.250E0
  i16    21.1500E0     2.250E0
  i17     8.1750E0     4.250E0
  i18    81.5000E0      .500E0
  i19    13.1200E0     3.000E0
  i20    59.9000E0      .750E0
  i21    14.6200E0     3.000E0
  i22    32.9000E0     1.500E0
  i23     5.4400E0     6.000E0
  i24    12.5600E0     3.000E0
  i25     5.4400E0     6.000E0
  i26    32.0000E0     1.500E0
  i27    13.9500E0     3.000E0
  i28    75.8000E0      .500E0
  i29    20.0000E0     2.000E0
  i30    10.4200E0     4.000E0
  i31    59.5000E0      .750E0
  i32    21.6700E0     2.000E0
  i33     8.5500E0     5.000E0
  i34    62.0000E0      .750E0
  i35    20.2000E0     2.250E0
  i36     7.7600E0     3.750E0
  i37     3.7500E0     5.750E0
  i38    11.8100E0     3.000E0
  i39    54.7000E0      .750E0
  i40    23.7000E0     2.500E0
  i41    11.5500E0     4.000E0
  i42    61.3000E0      .750E0
  i43    17.7000E0     2.500E0
  i44     8.7400E0     4.000E0
  i45    59.2000E0      .750E0
  i46    16.3000E0     2.500E0
  i47     8.6200E0     4.000E0
  i48    81.0000E0      .500E0
  i49     4.8700E0     6.000E0
```

```
  i50    14.6200E0     3.000E0
  i51    81.7000E0      .500E0
  i52    17.1700E0     2.750E0
  i53    81.3000E0      .500E0
  i54    28.9000E0     1.750E0
;

*
* extract data
*
parameter x(i),y(i);
x(i) = data(i,'x');
y(i) = data(i,'y');


*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    sse          'sum of squared errors'
    b1           'coefficient to estimate'
    b2           'coefficient to estimate'
    b3           'coefficient to estimate'
;

equations
    fit(i)     'the non-linear model'
    obj        'objective'
;

obj..     sse =n= 0;
fit(i)..  y(i) =e= exp(-b1*x(i))/(b2+b3*x(i));

*-----------------------------------------------------------------------------
* initial values
*-----------------------------------------------------------------------------

b1.l = 0.1;
b2.l = 0.01;
b3.l = 0.02;

option nlp=nls;
model nlfit /obj,fit/;
solve nlfit minimizing sse using nlp;
display sse.l,b1.l,b2.l;


*-----------------------------------------------------------------------------
* load and display the confidence intervals
*-----------------------------------------------------------------------------


sets
   alpha /'90%','95%','97.5%','99%'/
   v /b1,b2,b3/
   interval /lo,up/
;
parameter confint(alpha,v,interval);
execute_load 'nls.gdx',confint;
display confint;
```

12.20. **Tests.** This section contains some test models.

12.20.1. *Solve Longley as NLS.* The Longley[25] dataset is a difficult problem for linear OLS solvers. In this example we try to solve it using NLS. We use a starting point of one for all variables.

12.20.2. *Model longleynls.gms.* [35]

```
$ontext

  Longley Linear Least Squares benchmark problem

  Solve as nonlinear regression problem

  Erwin Kalvelagen, nov 2004


  References:
       http://www.itl.nist.gov/div898/strd/lls/lls.shtml

       Longley, J. W. (1967).
       An Appraisal of Least Squares Programs for the
       Electronic Computer from the Viewpoint of the User.
       Journal of the American Statistical Association, 62, pp. 819-841.

               Certified Regression Statistics

                                           Standard Deviation
     Parameter         Estimate               of Estimate

        B0       -3482258.63459582         890420.383607373
        B1        15.0618722713733         84.9149257747669
        B2       -0.358191792925910E-01    0.334910077722432E-01
        B3       -2.02022980381683         0.488399681651699
        B4       -1.03322686717359         0.214274163161675
        B5       -0.511041056535807E-01    0.226073200069370
        B6        1829.15146461355         455.478499142212

     Residual
     Standard Deviation   304.854073561965

     R-Squared            0.995479004577296


               Certified Analysis of Variance Table

Source of Degrees of     Sums of             Mean
Variation  Freedom       Squares            Squares          F Statistic

Regression    6      184172401.944494   30695400.3240823   330.285339234588
Residual      9      836424.055505915   92936.0061673238

Chazam output:
-------------

Hello/Bonjour/Aloha/Howdy/G Day/Kia Ora/Konnichiwa/Buenos Dias/Nee Hau/Ciao
 Welcome to SHAZAM - Version 10.0 -  JUL 2004 SYSTEM=LINUX    PAR=   781
 |_SAMPLE 1 16
 |_READ  Y X1 X2 X3 X4 X5 X6
    7 VARIABLES AND       16 OBSERVATIONS STARTING AT OBS       1


 |_OLS Y X1 X2 X3 X4 X5 X6

 REQUIRED MEMORY IS PAR=       3 CURRENT PAR=     781
  OLS ESTIMATION
        16 OBSERVATIONS     DEPENDENT VARIABLE= Y
 ...NOTE..SAMPLE RANGE SET TO:      1,     16

  R-SQUARE =   0.9955     R-SQUARE ADJUSTED =   0.9925
 VARIANCE OF THE ESTIMATE-SIGMA**2 =   92936.
 STANDARD ERROR OF THE ESTIMATE-SIGMA =   304.85
 SUM OF SQUARED ERRORS-SSE=  0.83642E+06
 MEAN OF DEPENDENT VARIABLE =   65317.
 LOG OF THE LIKELIHOOD FUNCTION = -109.617
```

[35] www.amsterdamoptimization.com/models/regression/longleynls.gms

```
 MODEL SELECTION TESTS - SEE JUDGE ET AL. (1985,P.242)
  AKAIKE (1969) FINAL PREDICTION ERROR - FPE =     0.13360E+06
     (FPE IS ALSO KNOWN AS AMEMIYA PREDICTION CRITERION - PC)
  AKAIKE (1973) INFORMATION CRITERION - LOG AIC =   11.739
  SCHWARZ (1978) CRITERION - LOG SC =               12.077
 MODEL SELECTION TESTS - SEE RAMANATHAN (1998,P.165)
  CRAVEN-WAHBA (1979)
     GENERALIZED CROSS VALIDATION - GCV =          0.16522E+06
  HANNAN AND QUINN (1979) CRITERION =              0.12759E+06
  RICE (1984) CRITERION =                          0.41821E+06
  SHIBATA (1981) CRITERION =                        98018.
  SCHWARZ (1978) CRITERION - SC =                  0.17584E+06
  AKAIKE (1974) INFORMATION CRITERION - AIC =      0.12540E+06

                     ANALYSIS OF VARIANCE - FROM MEAN
                       SS         DF            MS                F
 REGRESSION       0.18417E+09      6.       0.30695E+08          330.285
 ERROR            0.83642E+06      9.        92936.              P-VALUE
 TOTAL            0.18501E+09     15.       0.12334E+08            0.000

                     ANALYSIS OF VARIANCE - FROM ZERO
                       SS         DF            MS                F
 REGRESSION       0.68445E+11      7.       0.97779E+10       105210.860
 ERROR            0.83642E+06      9.        92936.              P-VALUE
 TOTAL            0.68446E+11     16.       0.42779E+10            0.000


 VARIABLE   ESTIMATED  STANDARD   T-RATIO       PARTIAL STANDARDIZED ELASTICITY
   NAME    COEFFICIENT   ERROR      9 DF     P-VALUE CORR. COEFFICIENT  AT MEANS
 X1          15.062      84.91      0.1774     0.863 0.059     0.0463     0.0234
 X2       -0.35819E-01 0.3349E-01  -1.070      0.313-0.336    -1.0137    -0.2126
 X3        -2.0202     0.4884      -4.136      0.003-0.810    -0.5375    -0.0988
 X4        -1.0332     0.2143      -4.822      0.001-0.849    -0.2047    -0.0412
 X5       -0.51104E-01 0.2261      -0.2261     0.826-0.075    -0.1012    -0.0919
 X6         1829.2      455.5       4.016      0.003 0.801     2.4797    54.7342
 CONSTANT -0.34823E+07 0.8904E+06  -3.911      0.004-0.793     0.0000   -53.3132
 |_STOP


$offtext

set i 'cases' /i1*i16/;
set v 'variables' /empl,const,gnpdefl,gnp,unempl,army,pop,year/;
set indep(v) 'independent variables' /const,gnpdefl,gnp,unempl,army,pop,year/;
set depen(v) 'dependent variables' /empl/;

table data(i,v)
         empl gnpdefl   gnp    unempl    army     pop    year
  i1    60323   83.0   234289   2356     1590    107608  1947
  i2    61122   88.5   259426   2325     1456    108632  1948
  i3    60171   88.2   258054   3682     1616    109773  1949
  i4    61187   89.5   284599   3351     1650    110929  1950
  i5    63221   96.2   328975   2099     3099    112075  1951
  i6    63639   98.1   346999   1932     3594    113270  1952
  i7    64989   99.0   365385   1870     3547    115094  1953
  i8    63761  100.0   363112   3578     3350    116219  1954
  i9    66019  101.2   397469   2904     3048    117388  1955
  i10   67857  104.6   419180   2822     2857    118734  1956
  i11   68169  108.4   442769   2936     2798    120445  1957
  i12   66513  110.8   444546   4681     2637    121950  1958
  i13   68655  112.6   482704   3813     2552    123366  1959
  i14   69564  114.2   502601   3931     2514    125368  1960
  i15   69331  115.7   518173   4806     2572    127852  1961
  i16   70551  116.9   554894   4007     2827    130081  1962
;

data(i,'const') = 1;

alias(indep,j,jj,k);

table cert(*,*) 'certified values'
                   value                    stderr
```

```
   const     -3482258.63459582        890420.383607373
   gnpdefl    15.0618722713733        84.9149257747669
   gnp       -0.358191792925910E-01   0.334910077722432E-01
   unempl    -2.02022980381683        0.488399681651699
   army      -1.03322686717359        0.214274163161675
   pop       -0.511041056535807E-01   0.226073200069370
   year       1829.15146461355        455.478499142212

;


variables
   b(indep)  'parameters to be estimated'
   sse
;

equation
   fit(i)    'equation to fit'
   sumsq
;

*
* starting point
*
b.l(indep) = 1;

sumsq..   sse =n= 0;
fit(i)..  data(i,'empl') =e= sum(indep, b(indep)*data(i,indep));

option nlp = nls;
model leastsq /fit,sumsq/;
solve leastsq using nlp minimizing sse;
option decimals=8;
display b.l;

scalar err1 "Sum of squared errors in estimates";
err1 = sum(indep, sqr(b.l(indep)-cert(indep,'value')));
display err1;
abort$(err1>0.0001) "Solution not accurate";

scalar err2 "Sum of squared errors in standard errors";
err2 = sum(indep, sqr(b.m(indep)-cert(indep,'stderr')));
display err2;
abort$(err2>0.0001) "Standard errors not accurate";
```

12.20.3. *largeregnls.* This model is very large linear regression problem generated by using simulated data. It has 20000 observations with 100 parameters. Although a linear regression problem we estimate with NLS for testing purposes.

12.20.4. *Model largeregnls.* [36]

```
option sysout=on;
$ontext

   Test regression solver against a large simulated data set

$offtext

set i 'cases' /case1*case20000/;
set j 'parameters' /p0*p100/;
set j0(j) 'constant term' /p0/;
set j1(j) 'non-constant term' /p1*p100/;


parameter x(i,j) 'data, randomly generated; first column is constant term';
x(i,j0) = 1;
x(i,j1) = uniform(-100,100);
```

[36] www.amsterdamoptimization.com/models/regression/largeregnls.gms

```
parameter p_sim(j) 'values of parameters to construct simulation';
p_sim(j) = ord(j);

parameter y(i) 'data, simulated';
y(i) = sum(j, p_sim(j)*x(i,j)) + normal(0,10);

variables
   p_est(j)  'parameters, to be estimated'
   sse       'sum of squared errors'
;
equation
   obj       'dummy objective'
   fit(i)    'equation we want to fit'
;

p_est.l(j) = 1;

obj.. sse =n= 0;
fit(i).. y(i) =e= sum(j, p_est(j)*x(i,j));

$onecho > nls.opt
* increase default limits
maxn 30000
maxp 200
$offecho

option nlp=nls;
model ols1 /obj,fit/;
ols1.optfile=1;
solve ols1 minimizing sse using nlp;
```

12.20.5. *largeregnls2.* This model is very large non-linear regression problem generated by using simulated data. It has 20000 observations with 100 parameters all of them appearing non-linearly in the regression.

12.20.6. *Model largeregnls2.* [37]

```
option sysout=on;
$ontext

   Test regression solver against a large simulated data set

$offtext

set i 'cases' /case1*case20000/;
set j 'parameters' /p0*p100/;


parameter x(i,j) 'data, randomly generated';
x(i,j) = uniform(1,10);

parameter p_sim(j) 'values of parameters to construct simulation';
p_sim(j) = 1+ord(j)/50;
display p_sim;

parameter y(i) 'data, simulated';
y(i) = sum(j, x(i,j)**p_sim(j)) + normal(0,10);

variables
   p_est(j)  'parameters, to be estimated'
   sse       'sum of squared errors'
;
equation
   obj       'dummy objective'
   fit(i)    'equation we want to fit'
;

p_est.l(j) = 1;
```

[37] www.amsterdamoptimization.com/models/regression/largeregnls2.gms

```
obj.. sse =n= 0;
fit(i).. y(i) =e= sum(j, x(i,j)**p_est(j) );

$onecho > nls.opt
* increase default limits
maxn 30000
maxp 200
$offecho

option nlp=nls;
model ols1 /obj,fit/;
ols1.optfile=1;
solve ols1 minimizing sse using nlp;
```

References

1. K. Arrow, H. Chenery, B. Minhas, and R. Solow, *Capital-Labor Substitution and Economic Efficiency*, Review of Economics and Statistics **43** (1961), 225–250.
2. D. M. Bates and D. G. Watts, *Nonlinear regression analysis and its applications*, Wiley, New York, 1988.
3. L. Bennett, L. Swartzendruber, and H. Brown, *Superconductivity Magnetization Modeling*, NIST, 1994.
4. G. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters*, Wiley, New York, NY, 1978.
5. Barry W. Brown and Lawrence B. Levy, *Certification of Algorithm 708: Significant Digit Computation of the Incomplete Beta*, ACM Transactions on Mathematical Software **20** (1994), no. 3, 393–397.
6. C. Daniel and F. S. Wood, *Fitting Equations to Data*, 2nd ed., John Wiley and Sons, New York, NY, 1980.
7. Armido R. Didonato and Alfred H. Morris Jr., *Algorithm 708: Significant Digit Computation of the Incomplete Beta Function Ratios*, ACM Transactions on Mathematical Software **18** (1992), no. 3, 360–373.
8. K. Eckerle, *Circular Interference Transmittance Study*, NIST, 1979.
9. E.S.Keeping, *Introduction to Statistical Inference*, Van Nostrand Company, Princeton, NJ, 1962.
10. John Fox, *Nonlinear Regression and Nonlinear Least Squares, Appendix to* An R and S-PLUS Companion to Applied Regression, `http://cran.r-project.org/doc/contrib/Fox-Companion/appendix-nonlinear-regression.pdf`, January 2002.
11. M. Fuss, D. McFadden, and Y. Mundlak, *A survey of functional forms in the economic analysis of production*, The Theory of Production (M. Fuss and D. McFadden, eds.), Production Economics: A Dual Approach to Theory and Applications, vol. I, North-Holland, 1978.
12. William H. Greene, *Econometric Analysis*, 5th ed., Prentice-Hall, 2003.
13. W. E. Griffiths, R. C. Hill, and G. G. Judge, *Learning and Practicing Econometrics*, Wiley, 1993.
14. T. Hahn, *Copper thermal expansion study*, NIST, 1979.
15. G. W. Hill, *ACM Algorithm 396: Student's t-Quantiles*, Commun. ACM **13** (1970), no. 10, 619–620.
16. Jr. John E. Dennis, David M. Gay, and Roy E. Welsch, *Algorithm 573: NL2SOL – An Adaptive Nonlinear Least-Squares Algorithm*, ACM Trans. Math. Softw. **7** (1981), no. 3, 369–383.
17. Dale W. Jorgenson, *Econometrics: Econometric Modeling of Producer Behavior*, MIT Press, 2000.
18. D. Kahaner, C. Moler, and S. Nash, *Numerical Methods and Software*, Prentice Hall, Englewood Cliffs, NJ, 1989.
19. Erwin Kalvelagen, *A Linear Regression Solver for GAMS*, `http://www.amsterdamoptimization.com/pdf/regression.pdf`.
20. ———, *New special functions in GAMS*, `http://www.gams.com/~erwin/specfun.pdf`.
21. R. Kirby, *Scanning electron microscope line width standards*, NIST, 1979.

22. J.S. Kowalik and M. R. Osborne, *Methods for Unconstrained Optimization Problems*, Elsevier North-Holland, New York, NY, 1978.
23. NIST Information Technology Laboratory, *Nonlinear Regression Dataset Archive*, `http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml`.
24. C. Lanczos, *Applied Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1956.
25. J. W. Longley, *An Appraisal of Computer Programs for the Electronic Computer from the Point of View of the User*, Journal of the American Statistical Association **62** (1967), 819–841.
26. H. van Maaren and T. Terlaky, *Inverse barriers and CES-functions in linear programming*, Tech. Report 95-76, Delft University of Technology, 1995.
27. R. R. Meyer, *Theoretical and computational aspects of nonlinear regression*, Nonlinear Programming (Mangasarian Rosen and Ritter, eds.), Academic Press, New York, NY, 1970.
28. D. Misra, *Dental research monomolecular adsorption study*, NIST, 1978.
29. J. J. More, B. S. Garbow, and K. E. Hillstrom, *Testing unconstrained optimization software*, ACM Transactions on Mathematical Software **7** (1981), no. 1, 17–41.
30. W. Nelson, *Analysis of Performance-Degradation Data*, IEEE Transactions on Reliability **R-30** (1981), no. 2, 149–155.
31. M. R. Osborne, *Some aspects of nonlinear least squares calculations*, Numerical Methods for Nonlinear Optimization (Lootsma, ed.), Academic Press, New York, NY, 1972.
32. R Development Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2004, ISBN 3-900051-00-3.
33. D.A. Ratkowsky, *Nonlinear Regression Modeling*, Marcel Dekker, New York, NY, 1983.
34. L. Roszman, *Quantum Defects for Sulfur I Atom*, NIST, 1979.
35. G. A. F. Seber and A. J. Wild, *Nonlinear regression*, Wiley, New York, 1989.
36. B. Shea, *Algorithm AS 239: Chi-squared and Incomplete Gamma Integral*, Applied Statistics **37** (1988), no. 3, 466–473.
37. R. Thurber, *Semiconductor electron mobility modeling*, NIST, 1979.
38. Hal R. Varian, *Intermediate Microeconomics. A modern approach*, fifth ed., W. W. Norton & Co., 1999.
39. R. H. Wampler, *A Report of the Accuracy of Some Widely-Used Least Squares Computer Programs*, Journal of the American Statistical Association **65** (1970), 549–565.
40. Diana Whistler, Kenneth J. White, S. Donna Wong, and David Bates, *SHAZAM Version 9 User's Reference Manual*, Northwest Econometrics, 2001.

Amsterdam Optimization Modeling Group, Washington DC USA
E-mail address: `erwin@amsterdamoptimization.com`