Methods of Estimating S-Shaped Growth Functions: Algorithms and Computer Programs

M. Posch, A. Grübler and N. Nakicenovic

I. INTRODUCTION

It has been demonstrated that a large number of growth processes in biology as well as in the technology area and in the field of socio-economic activities follow regular S-shaped growth patterns. A number of alternative mathematical functions have been proposed and resulting algorithms applied as forecasting tools. The common deficiency of the proposed growth curve algorithms however lie in two areas. First, only certain growth curves or classes of growth curves are considered in a particular study and while there exist good arguments to prefer growth curves with an underlying theoretical justification (e.g. Bass, 1969 and 1980 innovator/imitator model as underlying theoretical framework for the Blackmann (logistic) technology diffusion model) over models without such a theoretical basis (as for instance the case for the Gompertz curve, Meade, 1984) no study of S-shaped growth patterns should *a priori* exclude any particular function to stand the test with the empirical data base. A second, more technical critique of existing algorithms concerns that as a rule linear transforms of the original data are used to estimate the model parameters, simply out of convenience of linear regression techniques. Equally with respect to the estimated parameters only standard statistical test (standard errors, t-statistics, D.W. tests etc.) are applied and an *explicit uncertainty analysis* of the estimated model parameters is not carried out.

The basic objective of this paper is therefore to overcome some of the deficiencies of traditional algorithms in terms that:
1) an (although not exhaustive) extended variety of S-shaped growth curves can be considered,
2) not only linear transforms of the data in a ordinary linear least squares algorithm but equally the original data in a new generalized least squares algorithm can be used to estimate the parameters of a particular growth function from empirical data,
3) uncertainty analysis of the estimations of the model parameters is explicitly considered, based on a Monte-Carlo simulation technique, without the generally assumed restricted assumption with respect to normal distribution and non-correlation of the observations,
4) a user-friendly operating environment and a high portability of the software is ensured.

Thus, the developed package is aimed to improve the analysis of empirical observed S-shaped growth patterns and to test whether the particular data analyzed can speak for themselves in determining the parameters of the underlying dynamic growth model. In the following chapters we discuss first the underlying mathematics for fitting non-linear growth curves to given (time series of) data. The different growth models, different parameter estimation algorithms (linear and non-linear least squares fit), parameter uncertainty analysis and graphics packages to plot the data are integrated into an interactive program package. The program not only computes the parameters of the desired curve (the postulated model of the growth process), but also (optionally) plots the computed curve and/or a linear transformation thereof together with the given data points. The program does not require any mathematical subroutines from a program library, and for the plotting subroutines only the graphics primitives (drawing a straight line, moving the cursor, etc.) have to be provided by the user. These features guarantee a high portability of this software package.

All curves considered are of the form

$$y = f(t) = f(\mathbf{a},t) \tag{1.1}$$

where t is the independent variable (usually time), $\mathbf{a}=(a_1,a_2,\cdots,a_m)$ is the vector of the m parameters to be determined, f is the function (the model) chosen by the user, and y is the dependent variable (growth, market share, technological performance, etc.). The general approach adopted for determining the parameters $\mathbf{a}$ is one of non-linear least square regression, i.e. a_1, a_2, ... are chosen to minimize

$$\sum_{k=1}^{N} w_k(y_k - f(\mathbf{a},t_k))^2 \tag{1.2}$$

where N is the number of observations (t_k,y_k), and w_k are positive weights ($w_k \equiv 1$ in the simplest case).

In the next chapter we describe the type of functions considered (S-shaped curves), in chapter III we outline the method for minimization and in chapter IV we shortly discuss the uncertainties connected with this kind of parameter fitting. In the Appendices additional material concerning the non-linear minimization algorithm, the required input file for the graphics subroutines as well as two tutorial sessions, describing the use of the program package, is presented.

II. TYPES OF GROWTH FUNCTIONS

Although the minimization method, described in the next chapter, works for any differentiable function, we restrict ourselves to so-called (S-shaped) growth curves, which are widely used in biology (growth and competition of species, see, e.g., Pearl, 1925; d'Ancona, 1939; Gatto, 1985), market dynamics (market penetration and substitution, see, e.g., Marchetti and Nakicenovic, 1979; Meade, 1984) and technology assessment (evolution of technological performance characteristics, see, e.g., Floyd, 1968; Martino, 1983).

All the curves discussed in this paper contain (at least) three parameters, which have the following interpretation:

(i) As t tends to infinity, y approaches an upper bound, which represents the level at which the growth process saturates, i.e.

$$\lim_{t\to\infty} f(t) = K \tag{2.1}$$

where K is positive and finite. Furthermore we consider only curves with $\lim_{t\to-\infty} f(t) = 0$.

(ii) There exists a time t_0, at which the curve has a point of inflection, i.e.

$$f''(t_0) = \frac{d^2f}{dt^2}(t_0) = 0 \tag{2.2}$$

where the growth rate has its maximum. A growth curve is called symmetric, if it is symmetric around t_0, i.e. $f(t_0-t)=f(t_0+t)$. A necessary condition for symmetry is

$$y_0 := f(t_0) = \frac{K}{2} \tag{2.3}$$

(iii) A third parameter, denoted by Δt, gives the length of the time interval needed to grow from 10% of K to 90% of K. More precisely, let t_p be defined by

$$f(t_p) = \frac{p}{100} K \ , \quad 0 < p < 100 \tag{2.4}$$

then Δt is given by

$$\Delta t = t_{90} - t_{10} \tag{2.5}$$

In the following we describe those types of curves which are implemented in the program and can be interactively chosen by the user.

(1) Three parameter logistic:

This curve is given by

$$y = f(t) = \frac{K}{1+e^{-b(t-t_0)}} \tag{2.6}$$

It is equally denoted as Verhulst (1838) or Pearl (1925) curve and in its application to the study of market dynamics referred to as Blackman's model (Blackman, 1972) and for $K=1$ as the Fisher-Pry model (Fisher and Pry, 1971). The curve is symmetric around t_0 and a simple calculation shows that the parameter Δt is related to the growth rate b by

$$\Delta t = \frac{1}{b} \log 81 = \frac{1}{b} 4.39444915... \tag{2.7}$$

For later reference we also note a commonly used form of re-writing the logistic function with a linear right-hand side:

$$\log \frac{y}{K-y} = b(t-t_0) \tag{2.8}$$

As an additional option the user can select a three parameter logistic with data dependent weights w_k given by

$$w_k = \frac{1}{\sigma_k^2} \tag{2.9a}$$

with

$$\sigma_k^2 = \frac{1}{K} f(t_k)[1 - f(t_k)] \tag{2.9b}$$

This choice of weights stems from an statistical interpretation of the data (see Debecker and Modis, 1986): Considering the y_k as an observation of the random variable $Y(t_k)$, the expectation of $Y(t)$ is given by $f(t)$ and its variance by Eq.2.9b (see also chapter IV).

(2) Gompertz function:

This non-symmetric growth function (see, e.g., Stone, 1980) is given by

$$y = f(t) = K\exp(-e^{-b(t-t_0)}) \tag{2.10}$$

The value at the point of inflection is given by

$$y_0 = f(t_0) = \frac{K}{e} \quad \text{where} \quad \frac{1}{e} = 0.36787944... \tag{2.11}$$

(e ... denotes the basis of the natural logarithm), and the parameter b is related to Δt via

$$\Delta t = \frac{1}{b} \log \frac{\log 10}{\log(10/9)} = \frac{1}{b} 3.08439977... \tag{2.12}$$

However, the application of the non-linear least square fit to the Gompertz function poses a serious problem: Because of the twofold exponentiation, the algorithm diverges even for data points, which are originally derived from a Gompertz function (also in double precision arithmetic). In order to be able to estimate this type of curve, the Gompertz function was re-written in the form

$$z = F(K,y) := -\log\log\frac{K}{y} = b(t - t_0) \tag{2.13}$$

and now a linear regression could be used to estimate t_0 and b. This, however, requires the *a priori* knowledge of K. In some cases the user might *know* K from theoretical considerations (e.g., y is a fraction, so K is equal to unity), and in this case the solution is obtained by a linear regression. If K should be *determined* by the program, an approach has been chosen which is outlined in chapter III*.

Besides the logistic function (Eq.2.8) and the Gompertz function (Eq.2.10), the following additional growth curves can be estimated:

---

* Note, that this method can only determine values of K which are larger than the maximum value of the observations. On the other hand it offers the possibility to estimate the parameters of growth curves which are only implicitly defined (like the Floyd curve).

(3) Sharif-Kabir function:

This four parameter function can be only written down implicitly

$$z = F(K,\gamma,y) = \log\frac{y}{K-y} + \gamma\frac{y}{K-y} = bt+c \ , \quad 0\leq\gamma\leq 1 \tag{2.14}$$

This model (Sharif-Kabir, 1976) contains two special cases: For $\gamma=0$ is reduces to Blackman's model (logistic function), while for $\gamma=1$ it corresponds to the **Floyd curve** (Floyd, 1968). For $\gamma\neq 0$ it is a non-symmetric function; the value at the inflection point, y_0, is given by

$$y_0 = f(t_0) = \frac{2K}{3+\sqrt{1+8\gamma}} \ , \quad 0\leq\gamma\leq 1 \tag{2.15}$$

showing that y_0 drops from $K/2$ to $K/3$ when γ increases from 0 to 1. Introducing the parameter t_0, the right-hand side of Eq.2.14 has to be rewritten in the following way

$$z(K,\gamma,y) = b(t-t_0) + \log\frac{2}{1+\sqrt{1+8\gamma}} + \frac{2\gamma}{1+\sqrt{1+8\gamma}} \tag{2.16}$$

And the parameter Δt now depends on γ and is given by

$$\Delta t = \frac{1}{b}\left[\log 81 + \frac{80}{9}\gamma\right] = \frac{1}{b}(4.39444915... + \gamma 8.88888888...) \tag{2.17}$$

Some typical Sharif-Kabir curves are depicted in Figure 1. Note that in order to obtain a y-value one has to solve Eq.2.14 iteratively (e.g. with Newton's method).

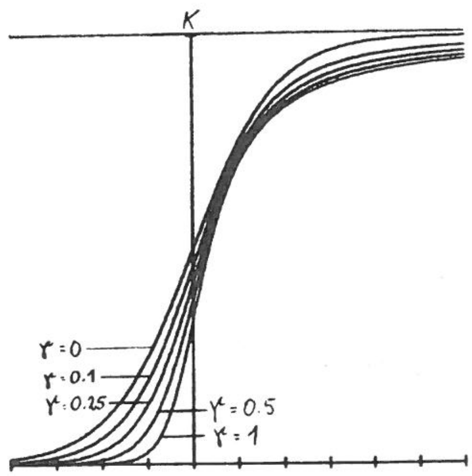


Figure 1: Sharif-Kabir functions with $\gamma=0$ (logistic curve) $\gamma=0.10$, $\gamma=0.25$, $\gamma=0.50$, and $\gamma=1$ (Floyd curve)

(4) Modified exponential:

This function, which is not a genuine S-shaped curve, is defined by

$$y = f(t) = K(1-e^{-b(t-t_0)}) \tag{2.18}$$

For this function the parameter t_0 does not indicate the point of inflection, but $f(t) < 0$ for $t < t_0$. The parameter b is related to Δt by

$$\Delta t = \frac{1}{b}\log 9 = \frac{1}{b}2.197224577... \tag{2.19}$$

and the "linear form" of the function is given by

$$z = F(K,y) = \log\frac{K}{K-y} = b(t-t_0) \tag{2.20}$$

III. SOLUTION METHOD

We consider the following minimization problem

$$V(\mathbf{a}) \longrightarrow \min\ , \quad \text{where } \mathbf{a} = (a_1,a_2,\cdots,a_m)^T \tag{3.1}$$

With $\mathbf{g}(\mathbf{a})$ we denote the gradient of V

$$\mathbf{g}(\mathbf{a}) = \left(\frac{\partial V}{\partial a_1},\frac{\partial V}{\partial a_2},\cdots,\frac{\partial V}{\partial a_m}\right)^T \tag{3.2}$$

and with $H(\mathbf{a})$ the matrix of the second derivatives (Hessian matrix)

$$H(\mathbf{a}) = \left(\frac{\partial^2 V}{\partial a_i \partial a_j}\right)\ , \quad i,j=1,...,m \tag{3.3}$$

Almost all minimization algorithms generate – starting from a vector $\mathbf{a}_0$ – a sequence of vectors $\mathbf{a}_k$, $k \geq 1$, which should approach the minimum $\bar{\mathbf{a}}$. In every step, for which $\mathbf{g}_k := \mathbf{g}(\mathbf{a}_k) \neq 0$, one determines a 'search direction' $\mathbf{s}_k$ and the next point

$$\mathbf{a}_{k+1} = \mathbf{a}_k + \lambda_k \mathbf{s}_k \tag{3.4}$$

via a 'linear minimization', i.e. the step-width λ_k is determined in such a way that

$$V(\mathbf{a}_{k+1}) \approx \min\{V(\mathbf{a}_k - \lambda \mathbf{s}_k) \mid \lambda \geq 0\} \tag{3.5}$$

The algorithms available mostly differ by the method for determining s_k. The Newton-method, e.g., selects $s_k = H(a_k)^{-1} g_k$ as a search direction and $\lambda_k = 1$ as step-width. This method has the advantage of fast (quadratic) convergence, but the disadvantage that one has to know and evaluate the Hessian matrix in every iteration step. In addition, the 'domain of attraction', i.e. the set of starting vectors a_0, for which convergence is achieved, is small -- and unknown! -- in many practical applications. Therefore one tries to replace the matrices $H(a_k)^{-1}$ by matrices H_k which are easier to compute. The method is called quasi-Newtonian if

$$H_{k+1}(g_{k+1} - g_k) = a_{k+1} - a_k \ , \quad k \geq 0 \tag{3.6}$$

In addition it is desirable that the matrices H_k are positive definite.

This requirements (quasi-Newtonian, positive definiteness) can be fulfilled by the following two-parameter recursion (Oren and Luenberger, see Stoer, 1983, and the original literature quoted therein):
With the abbreviations

$$p_k := a_{k+1} - a_k \ , \quad q_k := g_{k+1} - g_k \tag{3.7}$$

and the parameters

$$\gamma_k > 0 \ , \quad \theta_k \geq 0 \tag{3.8}$$

it takes the form

$$H_{k+1} = \Psi(\gamma_k, \theta_k, H_k, p_k, q_k) \tag{3.9}$$

where

$$\begin{aligned} \Psi(\gamma,\theta,H,p,q) = {} & \gamma H + \left(1 + \gamma\theta \frac{q^{\mathrm{T}} H q}{p^{\mathrm{T}} q}\right) \frac{p p^{\mathrm{T}}}{p^{\mathrm{T}} q} \\ & - \frac{\gamma(1-\theta)}{q^{\mathrm{T}} H q} H q q^{\mathrm{T}} H - \frac{\gamma\theta}{p^{\mathrm{T}} q} (p q^{\mathrm{T}} H + H q p^{\mathrm{T}}) \end{aligned} \tag{3.10}$$

This class of algorithms contains different special cases for different choices of γ_k and θ_k. We mention only two of them:

(a) $\gamma_k \equiv 1$, $\theta_k \equiv 0$: Method of Davidon, Fletcher and Powell (DFP-algorithm)

(b) $\gamma_k \equiv 1$, $\theta_k \equiv 1$: Rank-2-method of Broyden, Fletcher, Goldfarb and Shanno (BFGS-algorithm)

Altogether a minimization algorithm of the Oren-Luenberger class has the following form

(0) Select a_0 and a positive definite $m \times m$ matrix H_0, e.g. $H_0 := I$, and set $g_0 = g(a_0)$.
For $k = 0,1,...$ compute a_{k+1}, H_{k+1} from a_k and H_k as follows:

(1) If $g_k = 0$ stop; a_k is stationary point of V. Else

(2) Compute $s_k := H_k g_k$

(3) Determine $a_{k+1} = a_k - \lambda_k s_k$ by (approximate) linear minimization according to (3.5) and set $g_{k+1} := g(a_{k+1})$, $p_k := a_{k+1} - a_k$, $q_k := g_{k+1} - g_k$.

(4) Select suitable constants $\gamma_k > 0$, $\theta_k \geq 0$ and compute H_{k+1} via (3.10), $H_{k+1} = \Psi(\gamma_k, \theta_k, H_k, p_k, q_k)$.

In the current implementation we have selected $\gamma_k \equiv 1$ and $\theta_k \equiv 1$ (BFGS-algorithm). For the starting matrix the choice $H_0 = 0.01\,I$ (I ... unit-matrix) turned out to be advantageous. The approximate linear minimization in step (3) above is done in the following way

$$V(a_{k+1}) = \min\{V(a_k - 2^{-j} s_k) \mid j \geq 0\} \tag{3.11}$$

The following is a short description of the solution method, when the function is given in the "linear form", i.e.

$$z = F(K,y) = \alpha t + \beta \tag{3.12}$$

If the parameter K is fixed, the parameters α and β are obtained by simple linear regression

$$\alpha = \frac{NS_{tz} - S_t S_z}{NS_{t^2} - S_t^2} \tag{3.13a}$$

and

$$\beta = \frac{S_{tz} S_t - S_{t^2} S_z}{S_t^2 - NS_{t^2}} \tag{3.13b}$$

where

$$S_t := \sum_{k=1}^{N} t_k \;,\quad S_z := \sum_{k=1}^{N} z_k \;,\quad S_{t^2} := \sum_{k=1}^{N} t_k^2 \;,\quad S_{tz} := \sum_{k=1}^{N} t_k z_k \tag{3.13c}$$

and t_k are the observations and $z_k := F(K, y_k)$. For a fixed value of K let us denote

$$V_0(K) = \min_{\alpha,\beta} \sum_{k=1}^{N} (z_k - \alpha t_k - \beta)^2 \tag{3.14}$$

Obviously, α and β depend on K; $\alpha = \alpha(K), \beta = \beta(K)$. For the determination of an optimal K -- which has to be larger than $y_{\max} := \max_k y_k$ -- a simple search algorithm looks for

$$K_0 := \min\{V_0(K) \mid K = y_{\max} + j\delta, j = 1,...,J\} \tag{3.15}$$

where δ is an increment ($0.01 {}^* y_{\max}$) and $J = 400$ (corresponding to five times $y_{\max}$). This K_0 -- if one can be found -- is then used as a starting value for the non-linear minimization of the one-parameter function

$$V(K) = \sum_{k=1}^{N} (z_k(K) - \alpha(K)t_k - \beta(K))^2 \tag{3.16}$$

The minimization is carried out with the algorithm described above.

IV. PARAMETER UNCERTAINTIES

In addition to the estimated parameters of a chosen growth curve the correlation coefficient, R^2, is provided as a measure for the goodness of the fit. R^2 is calculated according to the following formula

$$R^2 = \frac{\left[\sum_{k=1}^{N} (y_k - \bar{y})(f(t_k) - \bar{f})\right]^2}{\sum_{k=1}^{N} (y_k - \bar{y})^2 \sum_{k=1}^{N} (f(t_k) - \bar{f})^2} \tag{4.1}$$

where

$$\bar{y} := \frac{1}{N} \sum_{k=1}^{N} y_k \ , \quad \bar{f} := \frac{1}{N} \sum_{k=1}^{N} f(t_k) \tag{4.2}$$

and f is the growth function found in the minimization process. Note, that in the case where the "linear form" of the growth function is minimized (see chapter III), R^2 is calculated with this linear form.

In principle, the minimization algorithm would provide the matrix of the second derivatives, and therefore allow the determination of the standard deviations on the values of the parameters and the corresponding confidence levels. However, this method of determination the confidence levels is not suitable, since it assumes that the parameters are normally distributed and, in addition, that they are uncorrelated. There is no compelling reason that such a restrictive assumption is warranted. Therefore only a numerical approach, i.e. a study based on several thousands of S-curve fits on simulated data covering the different conditions for the parameters, can circumvent this problem.

Such a study, using a Monte Carlo simulation type of approach, was carried out by Debecker and Modis (1986) for the three-parameter logistic function with data dependent weights w_k (see Eqs.1.2, 2.6 and 2.9a,b). This study provides look-up tables for determining the uncertainties associated with the three parameters K, t_0 and b. The uncertainties and the associated confidence levels are given as a function of the uncertainty on the observations and the length of the historical period. Nine ranges have been considered in the simulation: the observations are between 1% and 20% of K, ..., the observations are between 1% and 99% of K. For each range the expected error on each of the parameters K, t_0 and b as a function of the confidence level (seven values: 70%, 75%, 80%, 85%, 90%, 95%, and 99%) and the statistical error of the observations (six values: 1%, 5%, 10%, 15%, 20%, and 25% error) is given in a separate table. For intermediate values a trilinear interpolation scheme is used to determine the uncertainties of the parameters. In the general case where K and b are different from 1, the errors tabulated correspond to percentages. The units of the parameter t_0 are defined as (total historical range)/20. The uncertainty interval of Δt was calculated from the respective one of b via Eq.2.7.

Debecker and Modis (1986) conclude in their study that as a rule-of-thumb the uncertainty of the parameter K (saturation level) will be less than 20 percent within a 95 percent confidence level, provided at least half of the data are available with their precision better than 10 percent.

Since for the other growth curves no similar information was available, uncertainty bands could be included in the program only for the logistic function. It is suggested that the user starts first with a logistic fit to determine the order of magnitude of the uncertainty range, before considering a fit by alternative growth models.

CONCLUSIONS

The objective of the work documented in this paper was to formulate and implement a number of algorithms for estimation of S-shaped curves in an extremely user-friendly way by also assuring a high portability of the program. These objectives have been achieved. Through the interactive way of using the package, it does not require any *a priori* knowledge from the side of the user. The solely use of FORTRAN 77 as a programming language, the deliberate choice not to use any mathematical libraries as well as the limitation to the most elementary graphic primitives (drawing a line, moving the pen, etc.) ensures highest portability of the package.

The current version of the source code consists of about 3000 lines of code. For a maximum number of 500 observation the size of the executable is about 100 kbyte (including the graphics subroutines). Execution time on a VAX 11/780 requires less than 5 seconds of CPU-time and is thus negligible. Reasonable convergence of the non-linear least square fit (option 1) was achieved with samples generated with 100% normally distributed data errors. For the non-linear least square fit with data dependent weights, used for the uncertainty analysis (option 2), relative errors up to 50% still gave reasonable convergence. The algorithm used for the fit of the logarithmically transformed functions (options 3 to 6) shows convergence for data with errors in the 30% range.

REFERENCES

d'Ancona, U., 1939. Der Kampf ums Dasein. Gebrüder Borntraeger, Berlin.

Bass, F.M., 1969, A new product growth model for consumer durables. Management Science, 5, 215-227.

Bass, F.M., 1980, The relationship between diffusion rates, experience curves and demand elasticities for consumer durable technological innovations, Journal of Business, 3, 51-67.

Blackman, A.W., 1972. A mathematical model for trend forecasts. Technol. Forecasting Soc. Change, 3, 441-452.

Debecker, A., and Modis, T., (Digital Equipment Corporation) 1986. Determination of the uncertainties in S-curve logistic fits. Paper submitted to the Sixth International Symposium on Forecasting, Paris, June 15-18, 1986.

Fisher, J.C., and Pry, R.H., 1971. A simple substitution model of technological change. Technol. Forecasting Soc. Change, 3, 75-88.

Floyd, A., 1968. A methodology for trend-forecasting of figures of merit. In: Technological Forecasting for Industry and Government: Methods and Applications. (Edited by J. Bright), pp.95-109. Prentice-Hall, Englewood Cliffs, N.J.

Gatto, M., 1985. Introduzione all'ecologia delle populazioni. CLUP, Milano.

Marchetti, C., and Nakicenovic, N., 1979. The dynamics of energy systems and the logistic substitution model. RR-79-13, IIASA, Laxenburg.

Martino, J.P., 1983. Technological forecasting for decision making. Elsevier, New York.

Meade, N., 1984. The use of growth curves in forecasting market development - A review and appraisal. Journal of Forecasting, 3, 429-451.

Pearl, R., 1925. The biology of population growth. Knopf, New York.

Sharif, M.N., and Kabir, C., 1976. A generalized model for forecasting technological substitution. Technol. Forecasting Soc. Change, 8, 353-334.

Stoer, J., 1983. Einführung in die Numerische Mathematik I. 4th edition, Springer Verlag, Berlin, Heidelberg, New York, Tokyo.

Stone, R., 1980. Sigmoids. Bulletin in Applied Statistics, 7, 59-119, Sheffield City Polytechnic.

Verhulst, P.-F., 1838. Notice sur la loi que la population suit dans son accroissement. Correspondence Mathématique et Physique 10, 113-121.

APPENDIX A

In this Appendix we give the expressions for the gradients of the sum of squares for the functions considered in the non-linear minimization algorithm.

Let

$$V(\mathbf{a}) = \sum_{k=1}^{N} w_k (y_k - f(\mathbf{a},t_k))^2 \tag{A1}$$

be the function to be minimized. In the following we list the expressions for the first derivatives of the different functions considered:

(a) Three parameter logistic with unit weights:

$$V(a,t_0,b) = \sum_{k=1}^{N} (y_k - aq)^2 \tag{A2}$$

where

$$q = q_k(t_0,b) = \frac{1}{1+e^{-b(t_k-t_0)}} \tag{A3}$$

First derivatives:

$$\frac{\partial V}{\partial a} = -2\sum_{k=1}^{N} (y_k - aq)q \tag{A4a}$$

$$\frac{\partial V}{\partial t_0} = 2ab\sum_{k=1}^{N} (y_k - aq)q(1-q) \tag{A4b}$$

$$\frac{\partial V}{\partial b} = -2a\sum_{k=1}^{N} (t_k - t_0)(y_k - aq)q(1-q) \tag{A4c}$$

(b) Three parameter logistic with data dependent weights:

$$V(a,t_0,b) = \frac{1}{a}\sum_{k=1}^{N} \frac{1}{p}(yk - aq)^2 \tag{A5}$$

where

$$p = p_k(t_0,b) = q(1-q) \tag{A6}$$

First derivatives:

$$\frac{\partial V}{\partial a} = -\frac{1}{a^2}\sum_{k=1}^{N}\frac{1}{p}\,(y_k^2 - a^2 q^2) \tag{A7a}$$

$$\frac{\partial V}{\partial t_0} = \frac{b}{a}\sum_{k=1}^{N}\frac{1}{p}\,(y_k - aq)(y_k + aq - 2qy_k) \tag{A7b}$$

$$\frac{\partial V}{\partial b} = -\frac{1}{a}\sum_{k=1}^{N}\frac{1}{p}\,(t_k - t_0)(y_k - aq)(y_k + aq - 2qy_k) \tag{A7c}$$

APPENDIX B

Example for a file (fitpar) containing all relevant parameters needed for plotting:

```
# This file contains the parameters for plotting:
#
# Comments must have a '#' in the first column
#
# The following parameters can be specified:
# The parameters entries have to start in the 10th column, or later.
#
# gfile ..... name of the graphics output file/device (max. 14 chars)
# size ...... size of squared display area in cm
# xorig ..... x-value of the lower left corner of the plot relative
#             to the lower left corner of the device (cm)
# yorig ..... y-value of the lower left corner of the plot relative
#             to the lower left corner of the device (cm);
#             xo+size and yo+size should not exceed the
#             x- and y-dimension of the device, resp.
#             For the existing devices at IIASA the following
#             maximum values apply:
#             BBC-Plotter:                33.8 cm x 28 cm (13.3" x 11")
#             AED-Color-Graphic-Terminal: 34 cm x 34 cm (13.44" x 13.44")
#             VAX-Lineprinter             25.4 cm x 33 cm (10" x 13")
# xmin ...... minimum for x-axis
# xmax ...... maximum for x-axis
# ymax ...... maximum for y-axis (ymin=0)
# intx ...... number of intervals on the x-axis
# inty ...... number of intervals on the y-axis
# fmty ...... format for labeling the y-axis (max. 30 chars)
# ytext ..... text for labeling y-axis (max. 30 chars)
# lines ..... horizontal lines in logarithmic plot:
#             0 ... no horizontal lines are drawn
#             1 ... hor. lines corresponding to f(1-f) are drawn
#             2 ... hor. lines corresponding to f are drawn
# curve ..... curve plotting option:
#             0 ... curve NOT plotted (only data and parameters)
#             1 ... curve plotted
# title ..... title of the plot (max. 30 chars)
# author .... author (or whatever info) (max. 30 chars)
#
#234567890123456789012345678901234567890
gfile    graphfile
size     12.
xorig    1.
yorig    1.
xmin     1850
xmax     1970
ymax     100.
intx     10
inty     7
fmty     f4.2
ytext    legend for y-axis
lines    0
curve    1
title    title of graphic
author   IIASA, May 1987
```

APPENDIX C

In this Appendix two tutorial sessions of using the interactive growth curve fitting program are provided. User inputs are underlined. In order to illustrate the quality of the fitting routine, synthesized time series data were generated and used as input for the program. These synthesized data are derived from values of a Gompertz and of a logistic function, (saturation level $K = 100$, inflection point $t_0 = 1900$ and $dt = 100$ years) respectively, onto which a normally distributed random error term (with variance equaling 20% of the mean value) was added.

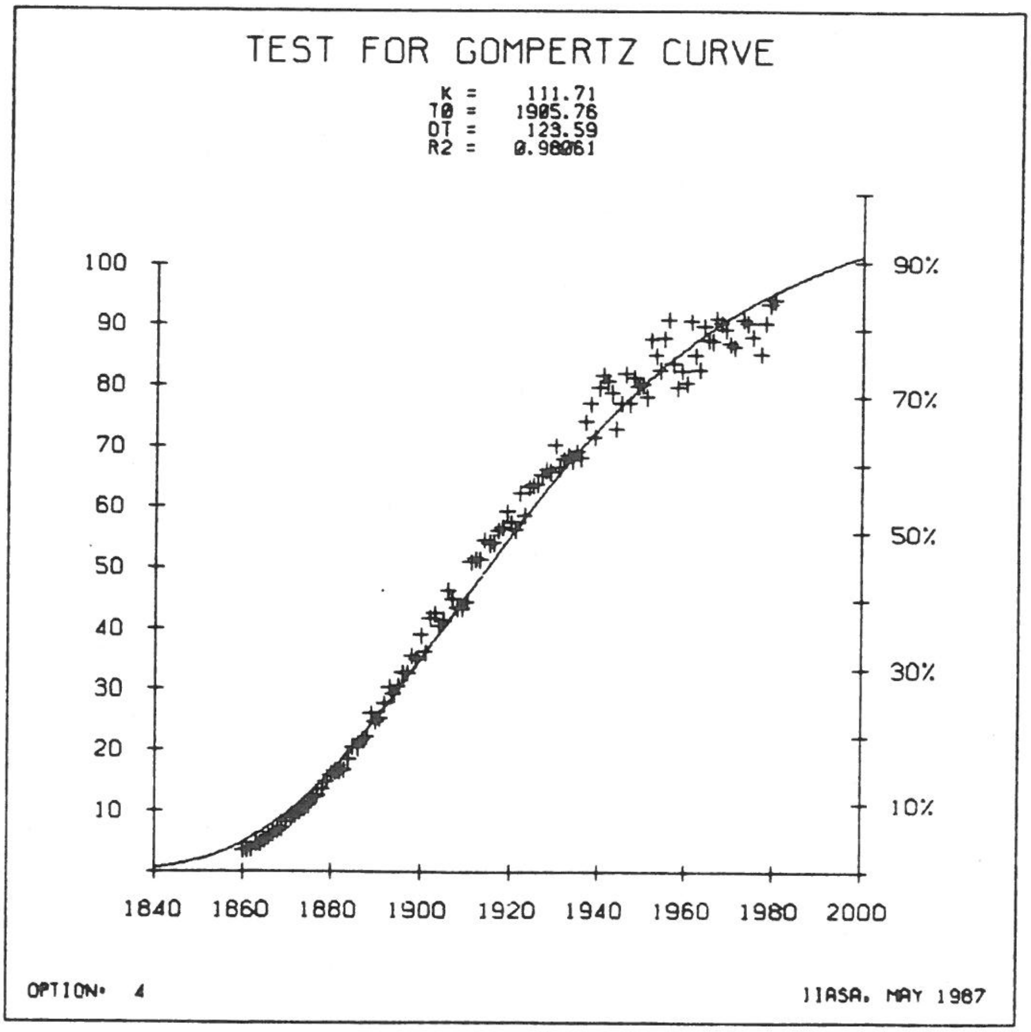
Input files in the examples described consist of the data files ***gompertz.data*** and ***logistic.data*** as well as the graphics parameter file ***fitpar*** (see printout below). The program generates two output files: one data file containing the function parameters as well as estimated and observed values (denoted by the postscript ***.res***) for further processing. The second file generated (***graphfile***, or any other name specified in ***fitpar***) is the device independent graphics output file, which then can be sent to a suitable plotting device.

In the first example data of a Gompertz function are read and the appropriate parameters estimated with a specified accuracy of 10^{-6}. Then a plot of the observations and the estimated curve is generated. In the second example data of a logistic function are used to estimate the parameters through the non-linear least square fit with data dependent weights starting from initial parameter estimates provided by the user (specified accuracy 10^{-6}). Then uncertainty ranges for the estimated parameters are interpolated from look-up tables for the specified data error and confidence level (20% and 90% in our example, respectively). Finally a plot in logarithmic transform $\log f/(1-f)$, where $f = K/y$, is produced.

Script of the interactive session for estimating a Gompertz function (user input is underlined):

```
% FIT

Do you want to ...

      1 ... compute (and ev. plot) parameters
      2 ... plot the observations only

Your choice: 1

Name of data-file: gompertz.data

You can estimate a

      1 ... logistic function
      2 ... weighted logistic function
      3 ... logarithm of the logistic function
      4 ... double logarithm of the Gompertz function
      5 ... logarithm of the exponential function
      6 ... logarithm of a Sharif-Kabir function

Your choice: 4

Do you want to fix the saturation level? [y/n]: n

accuracy < 10**-6

r2 =     0.98061

 K =     111.712
t0 =    1905.759
dt =     123.588

Do you want to have a graphic display? [y/n]: y

Name of parameter-file (RETURN if none): fitpar

Do you want to make ...

      1 ... a normal plot
      2 ... a logarithmic plot

Your choice: 1

Minimum of x-values of input data:      1860
Maximum of x-values of input data:      1980
Maximum of y-values of input data:       111.71

The following parameters for plotting are specified:

      1     gfile     ...   graphfile
      2     size      ...   12.
      3     xorig     ...   1.
      4     yorig     ...   1.
      5     xmin      ...
      6     xmax      ...
      7     ymax      ...   100.
      8     intx      ...   10
      9     inty      ...   10
      10    fmty      ...   i3
      11    ytext     ...
      12    lines     ...   0
      13    curve     ...   1
      14    title     ...
      15    author    ...   IIASA, May 1987

Input numbers of parameters you want to change
(RETURN if none): 5-8 14

Minimum value for x-axis: 1840

Maximum value for x-axis: 2000

Maximum value for y-axis: 100.

Number of intervals on x-axis: 8

Title (max. 30 chars): test for gompertz curve

      ... plotting .....

Do you want to have another display? [y/n]: n

Results written to file                gompertz.data.res
%
```

Graphic output (graphfile) of interactive session for estimating a Gompertz function:



Script of the interactive session for estimating a logistic function (user input is underlined):

```
% FIT

Do you want to ...

     1 ... compute (and ev. plot) parameters
     2 ... plot the observations only

Your choice: 1

Name of data-file: logistic.data

You can estimate a

     1 ... logistic function
     2 ... weighted logistic function
     3 ... logarithm of the logistic function
     4 ... double logarithm of the Gompertz function
     5 ... logarithm of the exponential function
     6 ... logarithm of a Sharif-Kabir function

Your choice: 2

Do you want to fix the saturation level? [y/n]: n

Input estimate of K, t0 and dt (10-90%): 100 1900 100

accuracy < 10**-6

r2 =    0.99247

Do you want to estimate uncertainties? [y/n]: y

input estimated data error (1%<=...<=25%): 20

input confidence level (70%<=...<=99%): 95

K  =     103.390      (     91.188   ...    115.592)
t0 =    1902.175      (   1899.836   ...   1904.515)
dt =     105.291      (     97.436   ...    114.524)

Do you want to have a graphic display? [y/n]: y

Name of parameter-file (RETURN if none): fitpar

Do you want to make ...

     1 ... a normal plot
     2 ... a logarithmic plot

Your choice: 2

Minimum of x-values of input data:      1850
Maximum of x-values of input data:      1960
Maximum of y-values of input data:       103.39

The following parameters for plotting are specified:

      1     gfile     ...    graphfile
      2     size      ...    12.
      3     xorig     ...    1.
      4     yorig     ...    1.
      5     xmin      ...
      6     xmax      ...
      7     ymax      ...    100.
      8     intx      ...    10
      9     inty      ...    10
     10     fmty      ...    i3
     11     ytext     ...
     12     lines     ...    0
     13     curve     ...    1
     14     title     ...
     15     author    ...    IIASA, May 1987

Input numbers of parameters you want to change
(RETURN if none): 5 6 8 14

Minimum value for x-axis: 1840

Maximum value for x-axis: 1980

Number of intervals on x-axis: 7

Title (max. 30 chars): test for logistic curve

     ... plotting .....

Do you want to have another display? [y/n]: n

Results written to file            logistic.data.res
%
```

Graphic output (graphfile) of interactive session for estimating a logistic function: