A SPLINE LEAST SQUARES METHOD FOR NUMERICAL
PARAMETER ESTIMATION IN DIFFERENTIAL EQUATIONS

$$
80-8
$$

J. M. Varah<br>Computer Science Department University of British Columbia


#### Abstract

In this paper, we describe a straightforward least squares approach to the problem of finding numerical values for parameters occurring in differential equations, so that the solution best fits some observed data. The method consists of first fitting the given data by least squares using cubic spline functions with knots chosen interactively, and then finding the parameters by least squares solution of the differential equation sampled at a set of points. We illustrate the method by three problems from chemical and biological modelling.


1. Introduction

The general problem can be stated as follows: we are given a system of ordinary differential equations

$$
\left.\begin{array}{rl}
\frac{d y_{1}}{d t} \equiv y_{1}^{\prime} & =f_{1}(t, y, p)  \tag{1.1}\\
\vdots & \\
y_{n}^{\prime} & =f_{n}(t, y, p)
\end{array}\right\}
$$

where $p=\left(p_{1}, \ldots, p_{m}\right)$ are $m(r e a l)$ parameters whose numerical values are unknown. As well, the solution vector $\underline{y}(t)$ has been measured at certain data points $\left\{t_{i}, i=1, \ldots, N\right\}$. The problem is to find reasonable values for $p$ so that the solution of (1.1) with these parameter values, and suitably chosen initial conditions, fits the given data.

As a specific example, consider the Lotka/Volterra predator prey
model (see e.g., Clark (1976), pg. 194)

$$
\left\{\begin{array}{l}
y_{1}^{\prime}=p_{1} y_{1}-p_{2} y_{1} y_{2} \\
y_{2}^{\prime}=p_{3} y_{1} y_{2}-p_{4} y_{2}
\end{array}\right.
$$

Here $y_{1}(t)$ measures prey population, $y_{2}(t)$ predator population, and the $\left\{p_{i}\right\}$ are positive constants dealing with birth, death, and interaction rates. Typically, there are measured values for $y_{1}(t), Y_{2}(t)$ at certain times $t=t_{i}$, and we are asked to provide reasonable values for the $\left\{p_{i}\right\}$ so the solution $\left(y_{1}(t), y_{2}(t)\right)$ approximates the data. Of course there is a certain amount of error inherent in the data, so
we cannot expect to fit the data perfectly. As well there may or may not be exact initial conditions given; the first point $\left(y_{1}\left(t_{1}\right), y_{2}\left(t_{1}\right)\right)$ may be just as much in error as the other points.

Such problems arise frequently in various areas: for example, in chemical reaction equations, and in the modelling of biological and ecological processes. In this paper, we describe a simple approach to the problem, discuss its merits relative to other methods which have been proposed, and illustrate the method on three specific examples.

## 2. Method of Solution

Recently, the most popular method for solving this problem has been an initial value technique: initial estimates of the parameters are made, and equation (1.1) is integrated using these parameters and some (possibly given) set of initial conditions. Then the least squares deviation of the solution at the data points is measured, and this is treated as a function of the parameters, which one tries to minimize over the space of parameter values. This approach is described in Bard (1974), van Domselaar and Hemker (1975), and Benson (1979). Although it can work well, we feel it has some serious drawbacks:
(a) The solution of (1.1) may be very sensitive to the initial conditions, thus making it difficult to integrate the equations. This sensitivity will be worsened in cases where the initial conditions are not known accurately; often the data error is at least 10\%.
(b) The technique requires guessing the parameter values; if these are not known reasonably well in advance, again it may be difficult
to integrate the equations, and the behavior of the solution may be totally different from that obtained using "good" parameters.
(c) There is a large amount of computational work involved; each new set of parameters requires a full-scale integration of the equations, possibly with a special method (e.g., if the initial value system is stiff). This all means that the computer programs set up to solve such a problem are by necessity long and complex.
(d) When the parameters occur linearly in (1.1), as in the Lotka/ Volterra model, the method does not simplify: an iterative procedure still ensues.

We would like to propose a simple, straightforward method which, we feel, overcomes all of these drawbacks:
(1) First, fit the given data by least squares using cubic spline functions. That is, for each component $j=1, \ldots, n$, construct a cubic spline $s_{j}(t)$ with fixed knots $\left\{t_{k}^{*}\right\}, k=1, \ldots, q$, choosing the spline coefficients to minimize the least squares deviation at the data points. This technique is described in Chapter 14 of de Boor (1979) and we give some further details in the next section, as it is useful for data fitting in general.
(2) When these spline fits have been found (so that the data has in effect been smoothed), we then find parameters to minimize the least squares deviation in the differential equation system (1.1) measured at some set of sample points $\left\{\hat{t}_{i}\right\}, i=1, \ldots, M$. That is, we find $p$ to

$$
\begin{equation*}
\min _{p} \sum_{j=1}^{n} \sum_{i=1}^{M}\left[s_{j}^{\prime}\left(\hat{t}_{i}\right)-f_{j}\left(\hat{t}_{i}, \underline{s}, \underline{p}\right)\right]^{2} . \tag{2.1}
\end{equation*}
$$

In particular, notice that when the parameters $p$ appear linearly in (1.1), this is a linear least squares problem, which can be solved directly by setting up the overdetermined system of $n M$ equations in $m$ variables (p) and solving this by a $Q R$ factorization or by using the normal equations. Moreover, no initial value solver is needed and no specific initial conditions are required. Thus the amount of computation, and the complexity of the program needed, are much less than for the initial value method described earlier. We should add that this technique is not new: a similar method (using a different data fitting technique in (1)) was proposed by Swartz and Bremerman (1975), and other similar methods have probably been proposed earlier.

## 3. Least Squares Cubic Splines

To simplify the notation, assume we have only one y -component, so our data are $\left\{\left(t_{i}, y_{i}\right), i=1, \ldots, N\right\}$. Also let $a \leq t_{1} \leq t_{2} \leq \cdots \leq t_{N} \leq b$. We wish to approximate this by a cubic spline $s(t)$ with knots $\left\{t_{i}^{*}\right\}$, $a<t_{1}^{*} \leq t_{2}^{*} \leq \cdots \leq t_{q}^{*}<b$. Thus the spline $s(t)$ is made up of different cubic polynomials in each interval $\left(a, t_{1}^{*}\right),\left(t_{1}^{*}, t_{2}^{*}\right), \ldots,\left(t_{q}^{*}, b\right)$, matched at the knots so that $s^{\prime \prime}(t)$ is continuous throughout ( $\mathrm{a}, \mathrm{b}$ ). Since each cubic polynomial has four coefficients, and the above requires three continuity conditions per knot, we are left with ( $q$ + 4) coefficients to determine by least squares solution of the data equations $s\left(t_{i}\right)=y_{i}, i=1, \ldots, N$.

Of course, one can solve directly by expressing each cubic polynomial in powers of $t$, matching the continuity conditions at the knots, and
solving the least squares problem for the othex coefficients. However, it is much easier (technically, if not conceptually) to use a B-spline basis for the cubic splines: each cubic B-spline $B_{i}^{(4)}(t)$ is uniquely defined by 5 successive knots, and is positive inside and zero outside this range. They can be easily generated by the recurrence relation (see de Boor (1979), pg. 131)

$$
\begin{gathered}
B_{i}^{(1)}(t)= \begin{cases}1, & t_{i}^{*} \leq t \leq t_{i+1}^{*} \\
0, & \text { otherwise }\end{cases} \\
B_{i}^{(k)}(t)=\frac{t-t_{i}^{*}}{t_{i+k-1}^{*}-t_{i}^{*}} B_{i}^{(k-1)}(t)+\frac{t_{i+k}^{*}-t}{t_{i+k}^{*}-t_{i+1}^{*}} B_{i+1}^{(k-1)}(t), k=2,3,4 .
\end{gathered}
$$

Notice this generates in order the B-spline of degree 1, 2, and 3 over the appropriate set of knots. To make this complete, the endpoints a and $b$ must be included as 4-fold multiple knots. Since there are $q$ interior knots, this defines ( $q$ +4) B-splines, which then forms a basis for all cubic splines over these knots. To find the best least squares spline, we solve the overdetermined linear system

$$
\sum_{j=1}^{q+4} a_{j} B_{j}^{(4)}\left(t_{i}\right)=y_{i} \quad, i=1, \ldots, N
$$

This is the data fitting technique we advocate here. The knots $\left\{t_{k}^{*}\right\}$ must be chosen fairly carefully in order to get a reasonable fit of the data with not too many knots. It is best to do this interactively, using a graphics terminal if possible. We should also remark that the
knots may be multiple: a double knot, for example, allows the second derivative of the spline to be discontinuous. This can be helpful in fitting data which change abruptly.

Clearly it is important that the derivative $s^{\prime}(t)$ be a reasonable approximation to the rate of change of the given data. As is well known, this can be a tricky business, and it is for this reason that we have used cubic splines. Our experience indicates that such a cubic spline gives as good a derivative as can be expected from the data. As an indication of this, when we tried the method on the example of Anderssen and Bloomfield (1974), we obtained the same accuracy as they did for their Fourier or regularization method.
4. Details of the Method

To succeed, our technique should be used in an interactive enviromment. When the data are first fitted with a cubic spline, it is very important that a graph of the fit be plotted. For a given knot set $\left\{t_{k}^{*}\right\}$, we may obtain a small least squares residual at the data points, and yet the spline may deviate considerably from the "expected curve" between the data points. With an interactive plot, the knots $\left\{t_{k}^{*}\right\}$ can be adjusted to get a better "visual" or overall plot. Similarly, one cannot automate the choice of knots by minimizing the least squares residual at the data points over all possible knot sets $\left\{t_{k}^{*}\right\}$. Such a choice will not necessarily produce the "best" fit, because of the possible deviation between data points and because the final curve we are after is not the spline fit, but a solution curve of the differential
equation. This is borne out in our examples in Section 6. Similarly, the choice of sample points $\left\{\hat{t}_{i}\right\}$ is somewhat arbitrary, and should be done interactively. Enough points should be chosen that the behavior of the solution is adequately represented, and it is important to place sample points "where the action is", that is, where the solution is changing rapidly. It has been our experience that a reasonable selection of sample points $\left\{\hat{t}_{i}\right\}$ and knots $\left\{t_{k}^{*}\right\}$ is what is important, not their exact placement; that is, the method is fairly robust.

After the choice of sample points, the next step is the solution of (2.1). As we have mentioned, this is a linear least squares problem if the parameters appear linearly in (1.1), and if so, the problem can be solved directly using normal equations or a $Q R$ factorization (which is, of course, required for the spline fit earlier). If the parameters appear nonlinearly, however, there are many techniques, algorithms, and programs available. We have contented ourselves with using a simple direct search algorithm for nonlinear minimization, which has the advantage of not requiring any pratial derivative $\frac{\partial f_{i}}{\partial p_{j}}$. However, much more sophisticated techniques are available for such nonlinear least squares problems and should probably be used, since these partial derivatives are required in any case if we want to obtain confidence intervals for the parameters by solving the sensitivity equations (see eqn. (4.1) below). Particular methods are Levenbert/Marquardt (which is available in various implementations) and the algorithm of Dennis/Gay/ Welsch (1979), which was designed for large residual problems. For a
survey of such methods, see Nazareth (1980).

There is also an intermediate case which often arises: some of the parameters can appear linearly and some nonlinearly in (1.1). In this case one can use the idea of separability or variable projection (see Golub and Pereyra (1973), or Ruhe and Wedin (1980)), in which the linear parameters are implicitly solved for, the resulting (fully) nonlinear least squares problem is solved for the nonlinear parameters, and then the linear parameţers are obtained using their representation in terms of the nonlinear parameters. Since this reduces the size of the nonlinear least squares problem to be solved, it is worthwhile.

Once the parameters $p$ have been found, their validity can be checked by integrating the system (1.1) using these values. As we indicated earlier, this can present difficulties, particularly if exact initial conditions are not given. This is the case in our first two examples in Section 6, and we get around this problem by using the data at the first point $t_{l}$ as the initial data, but allowing these initial values to vary slightly, choosing those values which minimize the least squares residual over all the data points. This, of course, is another nonlinear least squares problem, but it only serves to give an appropriate estimate of the final best residual and best solution, not to change the parameter estimate.

Finally, we should mention a special problem which arises in the third problem considered below, whereby data is only given for some of the components. In our example, we handle this by converting the given system into one of higher order where only those variables appear which have data measurements. Of course, this only works if the other
variables can be explicitly solved for using the various equations of (1.1).
5. The Sensitivity Equations

After having obtained estimates of the parameters $p$ and the solution vector $\underline{y}(t)$, one can obtain estimates of the sensitivity and accuracy of the parameters. Define $z_{i j}=\frac{\partial y_{i}}{\partial p_{j}}, i=1, \ldots, n, j=1, \ldots, m$. Differentiating (1.1), it is easy to see that $z$ satisfies the first order linear equation

$$
\begin{equation*}
z^{\prime}=G(t, \underline{y}, \underline{p})+J(t, y, p) z \tag{4.1}
\end{equation*}
$$

where $\quad G_{i j}=\frac{\partial f_{i}}{\partial p_{j}}$ and $J_{i j}=\frac{\partial f_{i}}{\partial y_{j}}$. These are the sensitivity equations, and one can obtain estimates of the sensitivity of the solution to changes in the parameters (i.e., $\mathrm{Z}(\mathrm{t})$ ) by integrating (4.1) using the computed values for $\underline{p}$ and $\underline{Z}$. This is quite well known (see Bard (1974) for example), although it is not clear to this author what initial conditions should be used for (4.1). If exact initial conditions are specified, then $\mathrm{z}_{\mathrm{ij}}=0$ is appropriate and seems to always be used; however, if the initial conditions are not known exactly, then it is not clear that $z_{i j}=0$ should be used.

Once $Z(t)$ has been computed, confidence intervals for the parameters can be obtained in the usual way, by assuming that the least squares function

$$
\Phi(p)=\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{n}\left(y_{j}\left(t_{i}\right)-y_{i j}\right)^{2}
$$

is locally quadratic near the minimum $p^{*}$. Then if the uncertainty or noise level in $\Phi$ is $\varepsilon^{2}$, the confidence intervals are of the form

$$
\left|p_{i}-p_{i}^{*}\right| \leq \varepsilon \sqrt{H_{i i}^{-1}}
$$

where $H=H\left(p^{*}, y\right)$ is the Hessian matrix at the minimum. This Hessian consists of two terms:

$$
H_{k \ell}=\sum_{i=1}^{N} \sum_{j=1}^{n} z_{j \ell}\left(t_{i}\right) z_{j k}\left(t_{i}\right)+\sum_{i=1}^{N} \sum_{j=1}^{n}\left(y_{j}\left(t_{i}\right)-y_{i j}\right) \frac{\partial^{2} y_{j}\left(t_{i}\right)}{\partial p_{k} \partial p_{\ell}} .
$$

Usually the second term is considered negligible so that $H$ can be computed directly from $Z(t): H=\sum_{i=1}^{N} z^{T}\left(t_{i}\right) z\left(t_{i}\right)$. This is reasonable in cases where the residual is fairly small, or the problem is (nearly) linear in $p$. In other cases, however, the second term can materially affect $H$, and it is probably wise to use a nonlinear least squares routine which computes an approximate Hessian as the minimization proceeds (again, see Nazareth (1980)). Near-singularity of the Hessian can be caused by (for example) nearly linearly dependent parameters, or insufficient data to separate the parameters. In any case, this indicates the problem is poorly conditioned and should be revised.

## 6. Numerical Examples

A. Barnes' problem (see van Domselaar and Hemker (1975)) .

$$
\begin{aligned}
& y_{1}^{\prime}=p_{1} y_{1}-p_{2} y_{1} y_{2} \\
& y_{2}^{\prime}=p_{2} y_{1} y_{2}-p_{3} y_{2}
\end{aligned}
$$

| t | 0.0 | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 | 3.5 | 4.0 | 4.5 | 5.0 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| $\mathrm{y}_{1}$ | 1.0 | 1.1 | 1.3 | 1.1 | 0.9 | 0.7 | 0.5 | 0.6 | 0.7 | 0.8 | 1.0 |
| $\mathrm{y}_{2}$ | 0.3 | 0.35 | 0.4 | 0.5 | 0.5 | 0.4 | 0.3 | 0.25 | 0.25 | 0.3 | 0.35 |

This problem as originally given represented chemical reaction equations; however, it is also the well known Lotka/Volterra predator-prey model in ecology (see e.g., Clark (1976), pg. 194). The solution components to this system are oscillatory in nature and out of phase with each other. Moreover, the data are only accurate to about $10 \%$, so it is clear that we should not try too hard to fit the data closely.

Below is a sample of results obtained, using up to four knots in the spline approximation:

| $\begin{gathered} \text { knot } \\ \text { positions } \end{gathered}$ | spline residuals | DE residual (\#sample points) | parameters found | integrated residual | ```best init.cond.``` |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 3.0 | .16, . 11 | 1.3 (20) | .85,2.13,1.91 | . 35 | 1.02,.25 |
| 3.0 | .16, . 11 | 1.7 (40) | . $80,2.06,1.86$ | . 36 | 1.05,.26 |
| 1.5,3.0 | .14, . 04 | 1.0 (20) | . $85,2.20,2.04$ | . 38 | 1.02,.24 |
| 1.5,3.0 | .14, . 04 | 1.5 (40) | .83,2.17,2.01 | . 37 | 1.04,.24 |
| *0.4,2.5 | .10,.09 | 3.0 (40) | . $62,1.73,1.60$ | . 44 | 1.16,.29 |
| 0.9,2.1,3.6 | .11, . 04 | 1.6 (40) | .80,2.11,1.94 | . 36 | 1.05,.24 |
| 1.0,2.0,3.0,4.0 | .09, . 02 | 1.6 (40) | . $85,2.21,2.02$ | . 365 | 1.02,.24 |
| *.14,.97,3.2,3.9 | . $06, .01$ | 40 (40) | -. $01,2.19,1.47$ | 1.9 | 1.02,.24 |

Knot selection was made visually, except for the cases marked
with an asterisk; these knots were found by minimizing the least squares deviation of the spline fit for the first component. Notice that although these gave better spline fits, the corresponding parameter values were very poor. In some sense, we are trying "too hard" to fit the data, and produce a curve which is not close to an integral curve of the differential equations.

This problem is linear in $p$, so the minimization (2.1) is a linear least squares problem. Its residual is given in the third column; we used 20 (or 40) equally spaced sample points. In each case, the parameter values were checked by integrating the system from the first data point $(t=0)$ and varying the initial condition so as to obtain the smallest least squares deviation in the integrated residual. These results are given in the last two columns. We also give plots of the first one knot spline fit and the corresponding integration in Figures 1 and 2 (for the first component $y_{1}$ ) and for both 4-knot cases in Figures 3-4 and 5-6.

The Hessian matrix for this problem is not ill-conditioned, so in that sense the problem is well-conditioned. The rather large variation in parameter values obtained (for about the same residual) is due to the inaccuracy in the data, and the correspondingly large residual (relative to the size of the data).

Figure 1


Figure 2


Figure 3


Figure 4


Figure 5


B. Bellman's problem (see Bellman et al (1967))

$$
y_{1}^{\prime}=p_{1}\left(126.2-y_{1}\right)\left(91.9-y_{1}\right)^{2}-p_{2} y_{1}^{2}
$$

| $t$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $y_{1}$ | 0.0 | 1.4 | 6.3 | 10.4 | 14.2 | 17.6 | 21.4 | 23.0 |
| $t$ | 10 | 12 | 15 | 20 | 25 | 30 | 40 |  |
| $y_{1}$ | 27.0 | 30.4 | 34.4 | 38.8 | 41.6 | 43.5 | 45.3 |  |

This problem arises from a chemical reaction, and is also treated in van Domselaar and Hemker (1975). This is somewhat easier to solve than problem $A$, and we give results below:

| knot <br> positions | spline <br> residuals | DE residual <br> (\#sample points) | parameters <br> found | integrated <br> residual | best <br> init.cond. |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 20.2 | 2.7 | $.86(15)$ | $.46 \times 10^{-5, .27 \times 10^{-3}}$ | 3.9 | -1.10 |
| 20.2 | 2.7 | $.76(20)$ | $.46 \times 10^{-5}, .30 \times 10^{-3}$ | 4.0 | -.98 |
| 20.2 | 2.7 | $3.0(20)$ | $.47 \times 10^{-5, .31 \times 10^{-3}}$ | 3.7 | -1.49 |
| $5.0,15.0$ | 1.6 | 3.1 | $(40)$ | $.41 \times 10^{-5, .23 \times 10^{-3}}$ | 6.4 |

Notice that we get a better result using only one knot, and this is at
least indicated by the relative size of the $D E$ residual in column 3. In Figures $7-10$, we give the best results for both one and two knots. Again
the Hessian matrix was not ill-conditioned; however, in this case the data were fitted very well, so the variation in the parameters is much less than in Example A.



Figure 9


Figure 10
C. Enzyme Effusion Problem (see van Domselaar and Hemker (1975))

$$
\begin{aligned}
& y_{1}^{\prime}=p_{1}\left(27.8-y_{1}\right)+\frac{p_{4}}{2.6}\left(y_{2}-y_{1}\right)+\frac{4991}{t \sqrt{2 \pi}} \exp \left(-0.5\left(\frac{\log (t)-p_{2}}{p_{3}}\right)^{2}\right) \\
& y_{2}^{\prime}=\frac{p_{4}}{2.7}\left(y_{1}-y_{2}\right)
\end{aligned}
$$

| $t$ | 0.1 | 2.5 | 3.8 | 7.0 | 10.9 | 15.0 | 18.2 | 21.3 | 22.9 | 24.9 |
| :--- | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
| $y_{1}$ | 27.8 | 20.0 | 23.5 | 63.6 | 267.5 | 427.8 | 339.7 | 331.9 | 243.5 | 212.0 |
| $t$ | 26.8 | 30.1 | 34.1 | 37.8 | 42.4 | 44.4 | 47.9 | 53.1 | 59.0 | 65.1 |
| $y_{1}$ | 164.1 | 112.7 | 88.1 | 76.2 | 62.3 | 58.7 | 41.9 | 40.2 | 31.3 | 30.0 |
| $t$ | 73.1 | 81.1 | 91.2 | 101.9 | 115.4 | 138.7 | 163.2 | 186.7 |  |  |
| $y_{1}$ | 30.6 | 23.5 | 24.8 | 26.1 | 33.3 | 17.8 | 16.8 | 16.8 |  |  |

This problem represents the modelling of enzyme concentrations in the blood, inside and outside the heart, over a period of time. A complication here is that observations are only available on $y_{1}$. We remedy this (as indicated earlier) by solving the first equation for $y_{2}$, differentiating, and substituting in the second equation to get a single second order equation for $y_{1}$.

This was more difficult to solve than the first two, both because of the nonlinear parameters and because of the difficulty in obtaining a good spline fit to the data. Results were as follows:

| knot <br> positions | spline <br> residual | DE residual <br> (\#samp.le points) | parameters <br> found | integrated <br> residual |
| :---: | :---: | :---: | :---: | :---: |
| $8.0,11.0,23.0,43.0$ | 64 | $7.8(28)$ | $.326,2.674, .40, .198$ | 94 |
| $8.0,11.0,23.0,43.0$ | 64 | $15.5(40)$ | $.257,2.62, .364, .29$ | 70 |
| $9.2,11.25,22.7,42.8$ | 62 | $5.3(28)$ | $.36,2.72, .40, .04$ | 111 |
| $9.2,11.25,22.7,42.8$ | 62 | $15.7(40)$ | $.266,2.65, .353, .228$ | 64 |
| $8.0,12.0,18.0,24.0,43.0$ | 60 | 60 | $12.3(40)$ | $.278,2.673, .378, .193$ |

For each knot set, we used 28 sample points (= data points) and 40 sample points (skewed to represent the function better). The first knot set represents the best we could do with four knots chosen visually. We then tried optimizing the knot locations by minimizing the least squares deviation at the data points as a function of the four knots. This gave the second knot set which, with 40 sample points, gave the best result. We plot this spline fit and result in Figures 11 and 12. The third set of 5 knots was again chosen visually and gave equally good results (notice, however, that the last parameter has changed appreciably without affecting the residual). We again optimized the knots, but this final knot set was not as successful (although still reasonable). We plot this for comparison in Figures 13 and 14.

Figure 11




Figure 14


## References

Anderssen and Bloomfield (1974): Numerical differentiation procedures for non-exact data. Num. Math. 22, 157-182.
Y. Bard (1974): Nonlinear Parameter Estimation. Academic Press, New York.
R. Bellman et al (1967): Quasilinearization and the estimation of chemical rate constants from raw kinetic data. Math. Biosc. 1, 71-76.
M. Benson (1979) : Parameter fitting in dynamic models. Ecol. Mod. 6, 97-115.
C. de Boor (1979): A Practical Guide to Splines. Springer-Verlag, New York.

Dennis, Gay, and Welsch (1979): An adaptive nonlinear least squares algorithm. MRC Tech. Rep. \#2010, Univ. of Wisconsin, Madison.

Golub and Pereyra (1973): The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. SIAM J. Num. Anal. 10, 413-432.
L. Nazareth (1980): Some recent approaches to solving large residual nonlinear least squares problems. SIAM Review 22, 1-11.

Ruhe and Wedin (1980): Algorithms for separable nonlinear least squares problems. SIAM Review 22, 318-337.

Swartz and Bremerman (1975): Discussion of parameter estimation in Biological modelling: algorithms for estimation and evaluation of the estimates. J. Math. Biol. 1, 241-257.
van Domselaar and Hemker (1975): Nonlinear parameter estimation in initial value problems. Report NW18/75, Mathematical Centrum, Amsterdam.

