THE UNIVERSITY OF WISCONSIN—MILWAUKEE

CENTER
FOR
GREAT LAKES STUDIES

MILWAUKEE, WISCONSIN 53201 U.S.A.

SPECIAL REPORT NO. 14

Digital Computer Programs for Estimating Primary Production, Integrated Over Depth and Time, In Water Bodies

by

Everett J. Fee

Center for Great Lakes Studies
The University of Wisconsin--Milwaukee
Milwaukee, Wisconsin 53201 USA

December 1971

SPECIAL REPORT NO. 14

Digital Computer Programs for Estimating Primary Production, Integrated Over Depth and Time, In Water Bodies

by

Everett J. Fee

Center for Great Lakes Studies
The University of Wisconsin--Milwaukee
Milwaukee, Wisconsin 53201 USA

December 1971

TABLE OF CONTENTS

Introduction

The measurement of primary production of water bodies has long been of great concern to hydrobiologists, particularly since Lindeman (1942) explicitly formulated the theoretical basis of dynamical ecology. The basic in situ technique was pioneered by Gaardner and Gran (1927) as follows: (1) samples were collected at a number of depths in the euphotic zone; (2) these samples were enclosed in clear and darkened bottles which were then resuspended at the collection depths; (3) the bottles were allowed to incubate in situ for a length of time sufficient for photosynthetic activity to produce a measurable change in some chemical parameter; (4) the bottles were retrieved, and the uptake or evolution of some chemical species which could be related quantitatively to carbon uptake was measured; (5) the results were plotted against depth and the curves integrated planimetrically to derive areal production rates (mg C/m^2. incubation time). With some minor variations, this basic methodology has remained essentially unchanged up to the present day.

It is obvious from the description that the in situ method is exceedingly time consuming, costly in ship time, and not well adapted to cover very large water bodies, where it is frequently impractical for a ship to remain long at a single station. Moreover, day-to-day variability of areal production is often so great that daily measurements would be needed to estimate annual or seasonal rates reliably (Rodhe, et al. 1958, Fee 1971). This is clearly unfeasible with the in situ method even on small lakes. Further, the in situ method provides no rational approach for interpolating between observations since it does not incorporate the actual variations of surface irradiance. Also, technical complications associated with the use of ^{14}C as a tracer of carbon uptake (Steeman-Nielsen 1952), for example, extracellular excretion and respiration, confine incubation times to short intervals, usually 4 to 6 hours (Vollenweider et al. 1961). It is not a simple matter to extrapolate these incubation rates, which cover only a short part of the daylight period, to daily rates (Vollenweider 1970). Finally, since natural light is used, experiments can be performed only during a brief part of the day---a wasteful consequence when ship time is costly. A technique in which samples could be continuously processed has proved to be much more economical and practical.

To overcome the above difficulties, to thereby extend surveys in both space and time, a more general approach is required. The theoretical basis for such a procedure has been presented elsewhere (Fee 1969, 1971). This report gives a brief summary of the main theoretical and methodological conclusions and presents the digital computer programs necessary for routine application, to implement Vollenweider's (1970) suggestion that this method be adopted, where possible, in studies of phytoplankton productivity.

Theory

Vollenweider (1965) proposed a photosynthetic model (graphed in Figs. 1 and 2)

$$p = \frac{i}{[(1+i^2)(1+(ai)^2)^n]^{\frac{1}{2}}} \qquad [1]$$

where p = the relative rate of carbon uptake per unit volume at the relative light intensity i. Photosynthetic rates are made relative to the optimum rate of carbon uptake; and absolute light intensities are made relative to the light intensity I'_k, which is defined graphically in Fig. 1. As shown in Fig. 2, the parameters a and n describe the type and degree of inhibition of photosynthesis by high irradiances (Fee 1969). This model covers a wide range of data from the literature. It is a modification of an equation first proposed by Smith (1936) from empirical data. This was used by Talling (1957) to study integral photosynthesis in some English lakes.

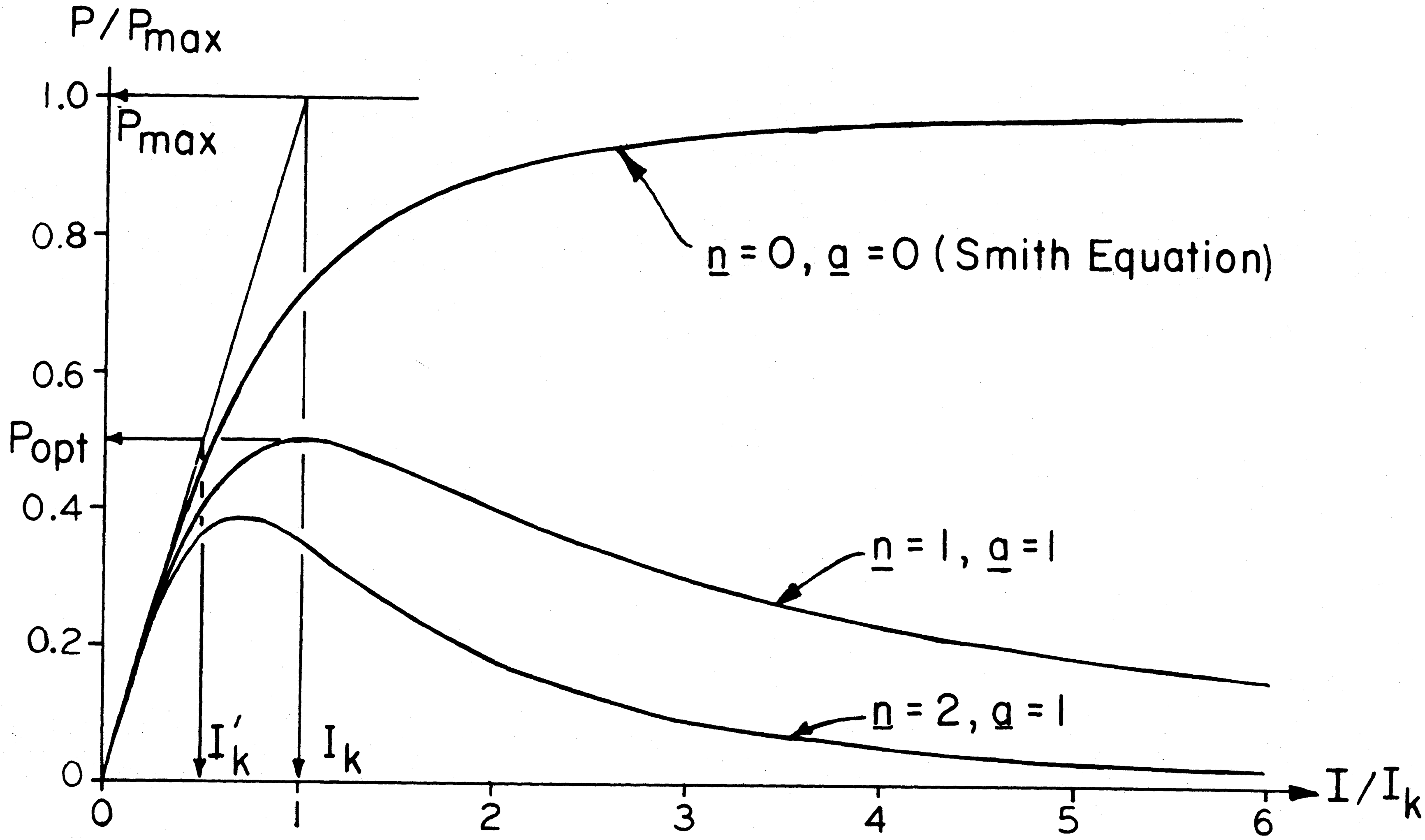

Figure 1. Photosynthesis vs. irradiance diagrams which illustrate the difference between I_k and I'_k and between P_{max} and P_{opt}. I_k is the irradiance at which the extrapolation of the nearly linear portion of the P/I curve intersects the asymptote $P = P_{max}$. I'_k is the irradiance at which this extrapolated line intersects the line $P = P_{opt}$.

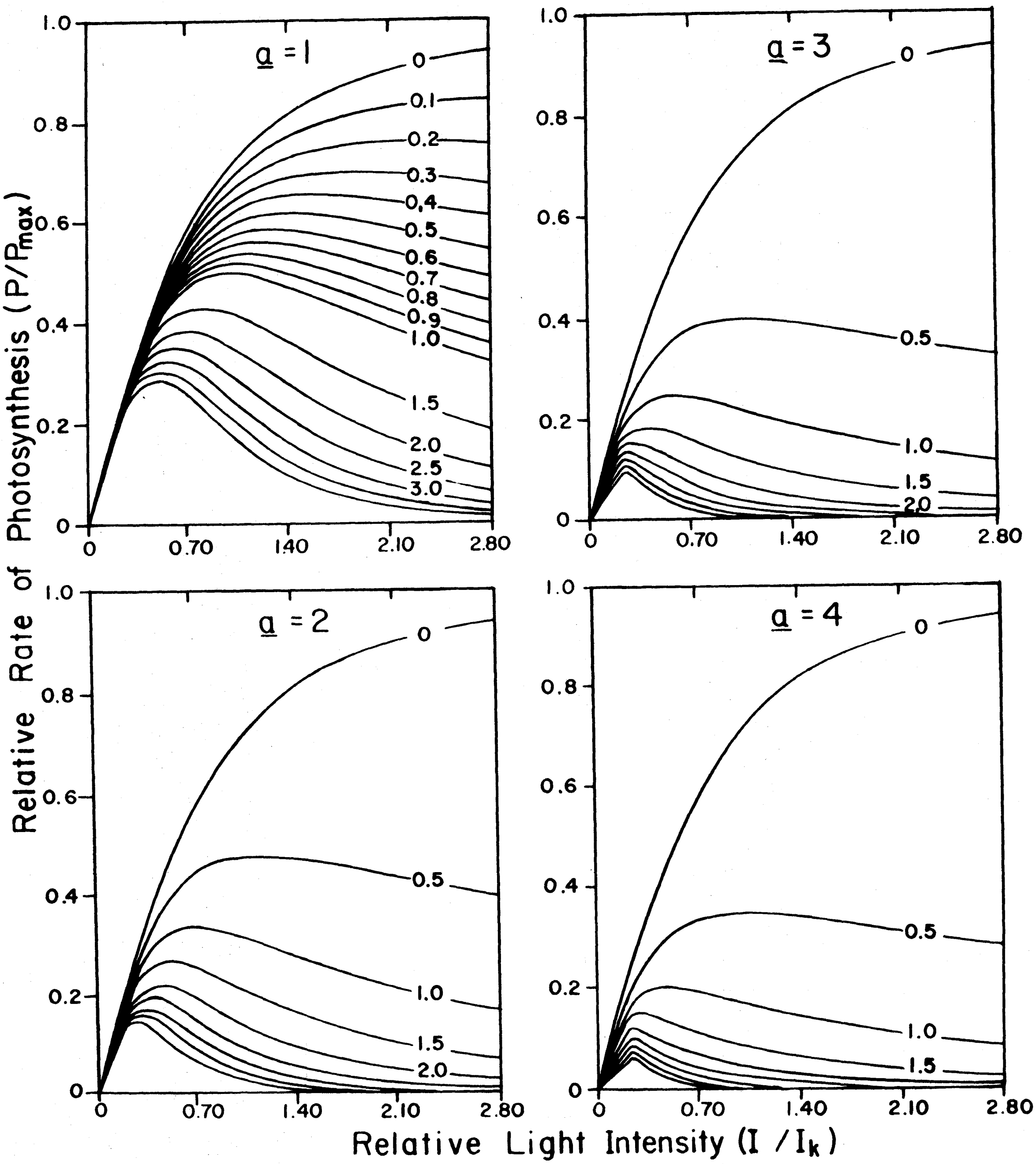

Figure 2. The photosynthesis model with different parameter combinations. The values of the parameter a are indicated at the top of each box and some representative values of n are superimposed on the appropriate lines.

By measuring the transparency of a given water body, it is possible to use equation [1] to compute the photosynthetic rate at any depth if the absolute irradiance at the surface is known. This photosynthesis/depth curve may be integrated in various ways; numerical integration is much the most convenient if a digital computer is available. This gives the integral rate at an instant in time. By measuring surface irradiance at a number of times during the day and repeating the depth integration, a collection of depth integrals is computed. The daily rate is obtained by integrating the depth integrals over time; again, any convenient integration procedure may be used.

Fee (1969, 1971) derived an analytical expression which mathematically describes the procedure just outlined verbally and is suitable for direct computer solution, viz.,

$$\Sigma\Sigma P = \frac{P_{opt}\lambda}{\varepsilon}\left[\frac{\delta}{2}\int_{-1}^{1}\int_{0}^{I_o(x)/\delta I'_k}\frac{1}{[(1+y^2)(1+a^2y^2)^n]^{\frac{1}{2}}}\,dy\,dx\right] \quad [2]$$

where

P = the rate of carbon uptake per unit volume and time,

λ = the daylength, taken with zero at midday,

P_{opt} = the optimum, or highest, rate of carbon uptake per unit volume and time (see Fig. 1),

ϵ = extinction coefficient of the waterbody,

δ = an auxilliary function of a and n (see Fee 1969 and below),

$I_o(x)$ = the absolute surface irradiance at normalized time $x = 2t/\lambda$,

I'_k = a light saturation parameter (see Fig. 1),

z = depth,

t = time,

x, y = dummy variables, and

a, n = parameters of the model (see Fee 1969).

Fee (1969, 1971) also proved that

$$\delta = [(1 + I_{opt}^2)(1 + a^2 I_{opt}^2)^n]^{\frac{1}{2}}/I_{opt} \qquad [3]$$

where

$$I_{opt} = \left[\frac{(1-n)a + \sqrt{a^2(n-1)^2 + 4n}}{2an}\right]^{\frac{1}{2}} \qquad [4]$$

Appendix 1 contains a digital computer program, written in FORTRAN IV, which accepts as input P_{opt}, λ, ϵ, I'_k, a, n and $I_o(x)$; and uses equations [2], [3] and [4] to compute daily integral photosynthesis. Simpson's rule for numerical integration is used for both the depth and time integrations. This treatment offers the considerable advantages that: (1) the model parameters a and n can assume any real values---not just those that allow an analytical solution of the depth integral; (2) no *a priori* assumption is made about the distribution of solar irradiance over time; and (3) the application to any particular system is straightforward, and indeed routine, once the parameters of the system are known or estimated.

The other major problem in implementing this solution is to obtain estimates of a, n and I'_k from experimental data relating relative photosynthetic rates to irradiance. The other parameters (P_{opt}, λ, ϵ and $I_o(x)$) can be measured directly (see Fee 1971 for complete example). When a high degree of accuracy is not necessary, suitable approximations to the values of a and n may be obtained by visual comparison of the experimental data with the family of curves presented in Fig. 2; and I'_k can be closely approximated with the method shown in Fig. 1 or by using Vollenweider's (1960) method for *in situ* data. In practice, the applications to date have shown that slight variations in a and n do not affect the computed integral rates to any great extent.

On the other hand, the I_k' values have been found to be quantitatively important and must be accurately estimated at all times (Fee 1971).

If a digital computer is available, very good estimates of $\underline{a}$, $\underline{n}$ and I_k' may be obtained starting from first principles. For any assumed combination of the parameters, one may fit the model at the experimental irradiances. Some arbitrary method of comparing the fitted model to the data points is needed and further a criterion for determining whether the fitted combination is in some sense the "best" possible fit. Because of the highly non-linear nature of the model, this is a difficult problem. Mathematically, the essence of the problem is that of finding the parameter vector

$$\underset{\sim}{\vec{X}} = (a,\ n,\ I_k')$$

such that a chosen single-valued function, $f(\vec{X})$, attains a minimum at $\underset{\sim}{\vec{X}}$. The route to a solution is further complicated by constraints on the parameters, since all must be real and non-negative. A three-dimensional representation of a problem of this sort is shown in Fig. 3. The choice of $f(\vec{X})$ is critical in this formulation. Any number of single-valued functions of parameter fits could be envisioned. For example, the summed absolute values of the differences between the fitted model and the observed data points might be a reasonable choice. Probably the most widely used and best understood of such functions is the expression

$$f(\vec{X}) = \sum_{i=1}^{m} \frac{(O_i - E_i)^2}{E_i} ,$$

where O_i is an observed value of the relative photosynthetic rate at a specified irradiance, E_i is the expected value at that irradiance under the assumed parameter combination and $\underline{m}$ is the number of observations. In words, this function represents the summed squared deviances of the observed values of relative production from the expected values under the assumed parameter combination. Each deviance is normalized by the expected value of relative production to make differences relative to the magnitude of the numbers being treated. Mood et al. (1963) prove that this widely used "goodness-of-fit"

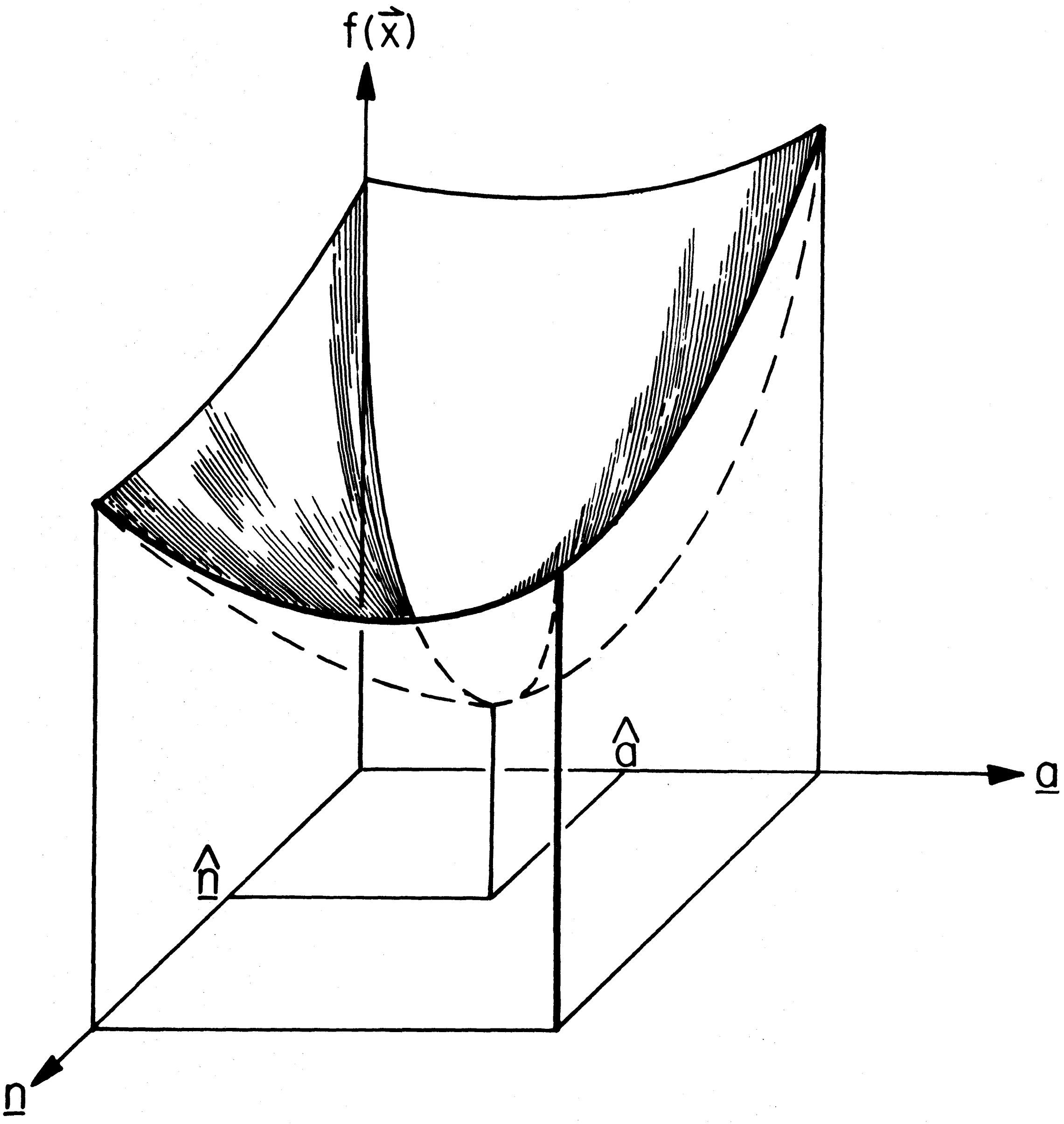

Figure 3. A hypothetical parameter space showing the type of minimization problem treated by the programs in Appendices 2 and 3. The minimum value of $f(\vec{X})$ occurs at $\vec{X} = (\hat{\underline{a}}, \hat{\underline{n}})$.

function is asymptotically distributed as the statistical chi-squared probability function; and f $(\vec{X})$ will hereafter be referred to as χ^2.

Having thus chosen f $(\vec{X})$, a computational scheme for finding the minimum of χ^2 for an arbitrary set of data must be found. Graphically this problem consists of placing a ball anywhere in the three-dimensional parameter space of Fig. 3 (this represents our first estimate) and allowing it to roll until it stops. It is immediately apparent that if there is more than one minimum the stopping point will be dependent upon the starting point. In fact, the only way of being certain of finding a global or smallest minimum---as opposed to a local minimum---is to try an infinite number of different starting points. This is obviously impossible in practice and the scheme developed below is not designed to find a global minimum but only a local minimum. Thus, the first estimates supplied by the user are of critical importance in determining the final parameter combination $\vec{\underset{\sim}{X}}$. For a and n these first estimates may be obtained by comparing the experimental P/I curves with those given in Fig. 2.

With a fixed parameter vector $\vec{X}$ the computer can calculate χ^2. If it can also determine how χ^2 varies in the vicinity of $\vec{X}$, then a scheme can be devised such that $\vec{X}$ can be changed to reduce χ^2. Specifically, if the partial derivatives of χ^2 space are known at $\vec{X}$ then the gradient of χ^2 is a vector,

$$\nabla\chi^2 = \left(\frac{\partial\chi^2}{\partial a}, \frac{\partial\chi^2}{\partial n}, \frac{\partial\chi^2}{\partial I'_k}\right)$$

which points "uphill", normal to a contour line of equal values of χ^2 including the point χ^2 $(\vec{X})$. Thus $-\nabla\chi^2$ points "downhill", or towards a minimum. By seeking along this vector we can devise a scheme for finding a better estimate of $\vec{\underset{\sim}{X}}$. The gradient vector is thus used to provide a linear approximation to the nonlinear parameter space. It is apparent that the criterion for a solution to the problem is the condition $\nabla\chi^2 = \vec{0}$, the zero vector. Thus, any local minimum is a proper solution. The constraints on the parameters previously mentioned are viewed as "walls" which intersect the parameter space and at which special steps must be taken so as not to project the next estimate into the prohibited space.

This general kind of formulation is obviously amenable to many numerical approaches. Indeed, Box (1966) has compared eight of the best modern techniques. He found that the most powerful general procedure for finding local minima is that of Fletcher et al. (1963). Their method is the one used here; they give the theoretical derivation, proofs of stability and the rate of convergence and none of this needs repetition here.

We may now proceed to use this theoretical base by providing the gradient vector, $\nabla\chi^2$. It is proved in Fee (1971) that:

$$\frac{\partial\chi^2}{\partial q} = \sum\left[1-\left(\frac{O_i}{E_i}\right)^2\right]\frac{\partial}{\partial q}\, E_i\,, \qquad [5]$$

where q is an arbitaray parameter. Now, from equations [1], [3], and [4], the following relationships may be derived (see Fee 1971 for proofs):

$$\frac{\partial E_i}{\partial I'_k} = \frac{E_i}{I'_k}\left\{ n\left(\frac{aI}{\delta I'_k}\right)^2\left[1+\left(\frac{aI}{\delta I'_k}\right)^2\right]^{-1} - \left[1+\left(\frac{I}{\delta I'_k}\right)^2\right]^{-1}\right\}, \qquad [6]$$

$$\frac{\partial E_i}{\partial a} = \left(\frac{I}{\delta I'_k}\right)^2 E_i\left[\frac{na\left[\frac{a}{\delta}\;\frac{\partial\delta}{\partial a}\;\;-1\right]}{[1+(aI/\delta I'_k)^2]} + \frac{\frac{1}{\delta}\;\frac{\partial\delta}{\partial a}}{[1+(I/\delta I'_k)^2]}\right], \qquad [7]$$

$$\frac{\partial E_i}{\partial n} = E_i\left\{\frac{1}{\delta}\left(\frac{I}{\delta I'_k}\right)^2\frac{\partial\delta}{\partial n}\left[\frac{na^2}{[1+(aI/\delta I'_k)^2]} + \frac{1}{[1+(I/\delta I'_k)^2]}\right] - \frac{1}{2}\ln\,[1+(aI/\delta I'_k)^2]\right\}, \qquad [8]$$

$$\frac{\partial \delta}{\partial q} = \frac{-\delta}{I_{opt}} \frac{\partial I_{opt}}{\partial q} + \frac{1}{2\delta I_{opt}} \{(1+I_{opt}{}^{2}) \frac{\partial}{\partial q}[1+(aI_{opt})^{2}]^{n} + 2I_{opt} \frac{\partial I_{opt}}{\partial q} [1+(aI_{opt})^{2}]^{n}\} , \qquad [9]$$

$$\frac{\partial}{\partial a}[1 + (aI_{opt})^{2}]^{n} = 2naI_{opt}[1 + (aI_{opt})^{2}]^{n-1}[a\frac{\partial I_{opt}}{\partial a} + I_{opt}] , \qquad [10]$$

$$\frac{\partial}{\partial n} [1+(aI_{opt})^{2}]^{n} = [1 + (aI_{opt})^{2}]^{n} \{\ln [1+ (aI_{opt})^{2}] + \frac{2na^{2}I_{opt}}{1+(aI_{opt})^{2}} \frac{\partial I_{opt}}{\partial n} \} , \qquad [11]$$

$$\frac{\partial I_{opt}}{\partial a} = \frac{1}{4na^{2}I_{opt}} \left[\frac{a^{2}(n-1)^{2}(1-a) - 4n}{\sqrt{a^{2}(n-1)^{2} + 4n}} \right] , \qquad [12]$$

and finally,

$$\frac{\partial I_{opt}}{\partial n} = \frac{1}{4n^{2}a\ I_{opt}} \left[\frac{a^{2}(n-1)^{2} - 2n}{\sqrt{a^{2}(n-1)^{2} + 4n}} - a \right] . \qquad [13]$$

Appendix 2 presents a computer program, written in ALGØL for the Univac 1108, which combines equations [5] through [13] with the method of Fletcher and Powell (1963) to yield a statistical method for estimating $\underline{a}$, $\underline{n}$, and I'_k. Appendix 3 contains a FORTRAN IV program which gives identical results.

The experimental data relating relative photosynthetic rate to light, needed for input to this program, are most usefully obtained with a constant light incubator. Data from in situ measurements are confusing for two reasons: (1) it is difficult to get enough observations in the region of high irradiance because of the exponential drop of light; most of the bottles end up in the subsaturation light levels of the P/I curve; (2) the variations of surface irradiance during the incubation time make it difficult to specify to what irradiance a given rate of photosynthetic production corresponds; this is especially true of the bottles near the surface---they may be at inhibiting levels of irradiance at the start of the experiment and at subsaturation levels at the end. The ability to define and maintain the light climate are great advantages of the incubator method. The fact that the quality of light differs from the in situ light climate does not appear to be too important for most quantitative applications (Talling 1960). If it is deemed necessary to simulate the in situ spectrum, this can be done with either chemical (McAllister et al. 1964) or glass filters (Kiefer et al. 1970). An incubator which was found suitable for applying the model to the Great Lakes is described and illustrated by Fee (1971). Figure 4 shows a typical set of experimental data obtained with this incubator from Lake Michigan and the fitted model of them given by the program in Appendix 2.

In the application of these procedures to an actual field situation, one more computer program is frequently useful. Most solarimeters produce an analog chart as output while the program presented in Appendix 1 requires that data be digitally formatted and equally spaced in time. It is an exceedingly time consuming procedure to manually read these data directly from the charts and punch them onto data processing cards. A mechanical digitizer can greatly expedite this process and also give better results than can be

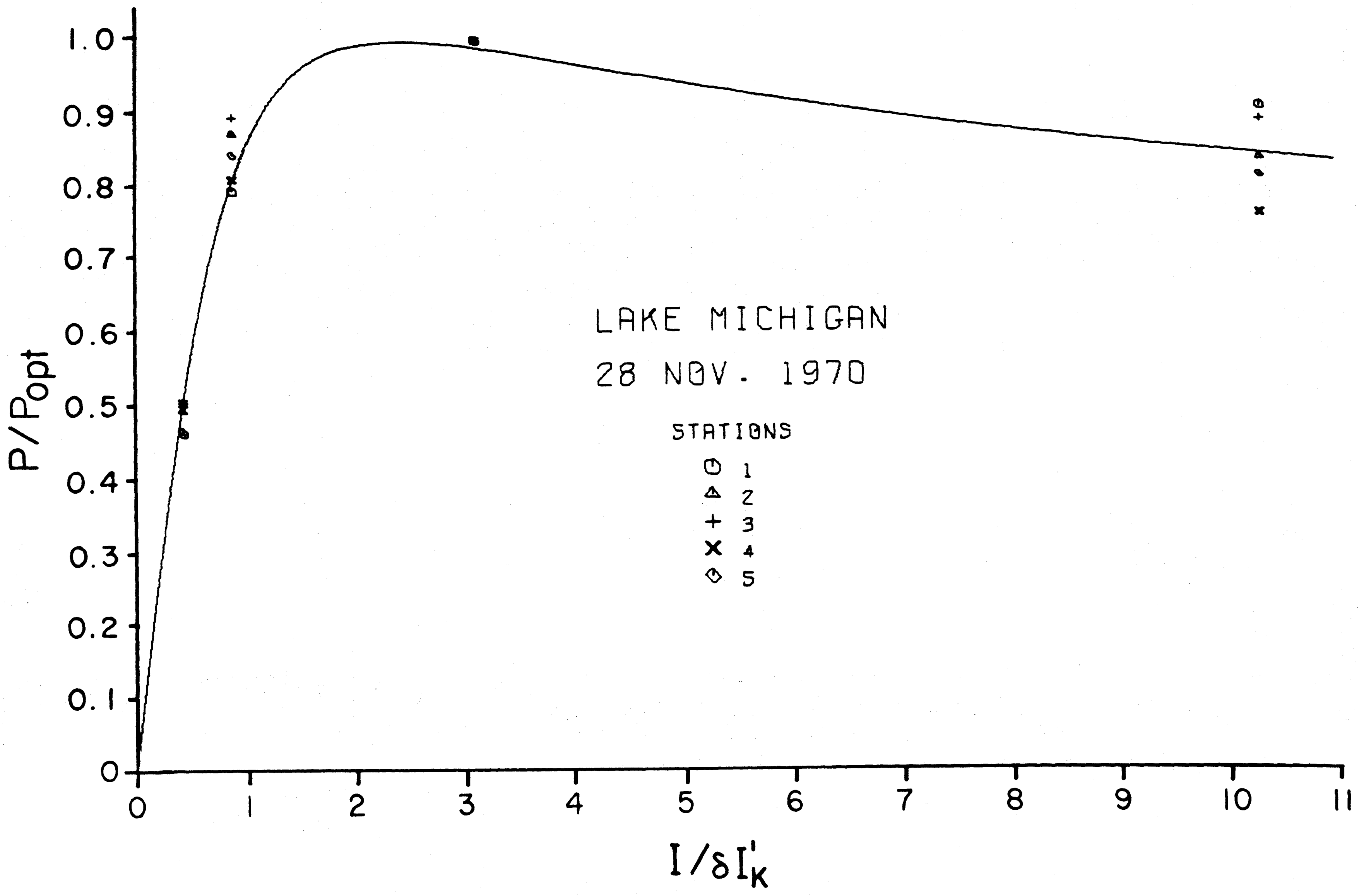

Figure 4. A typical set of experimental data relating relative photosynthesis to relative irradiance (from the data of Fee, 1971). The model fit given by the program in Appendix 2 is drawn as a solid line.

obtained manually. Fee (1971) used a d-Mac Pencil Follower for this purpose and there are a number of other machines now available that could be adapted for this. Appendix 3 presents a computer program, written in FORTRAN IV, which was developed to process the output from such a device. The program was written so that for curves which are quite smooth, only a few points need to be digitized and the program will use linear interpolation to compute the equally spaced data points. On the other hand, if the record is quite irregular the data can be numerically smoothed by digitizing the record at very close intervals of time and the program will use the trapezoidal rule for numerical integration to compute averages. There are two restrictions on the input data: (1) the data must be continuously increasing in time, i.e., the record must not double back on itself; and (2) the change between two successive ordinates must not exceed a user specified quantity. These two restrictions are imposed to ensure that the digitizer did not make a mechanical error. They also serve as a check on the quality of the digitizing. If an error of the first type occurs, the point is dropped. If an error of the second type occurs, the value for the ordinate is interpolated between the preceeding and succeeding points. Obviously, if too many errors of either kind occur, it will probably be wise to redigitize the data. This program is general, and can therefore also be used for many other applications, for example in the digitizing of water level records for use in spectral analyses (Fee 1969b).

Discussion

Early attempts to simplify the measurement of areal primary production through the use of shipboard incubators (Steemann-Nielsen 1952, Sorokin 1956) lacked a general theoretical framework. In particular, no account was taken of the actual distribution of surface light over time and empirical factors were used in the formulae. None of these formulae has been adopted for routine use by limnologists or oceanographers. The approach of the present work is general enough to apply to many water bodies.

These programs and the method based on them are strictly valid only for water bodies whose eupohtic zones are vertically isothermal and optically homogeneous. On large lakes, such as the Great Lakes, the oceans, or lakes which are too shallow to stratify, this is the usual situation. Vertical wind induced turbulence prevents the development of inhomogeneity in the upper layer where the bulk of primary production takes place. If a significant amount of light extends into a deeper layer which has different optical properties or algal populations, the lower layer would be treated separately and the total production obtained by summing the output from the two layers.

For the lakes which are so shallow that appreciable light reaches the bottom, the areal primary production by the phytoplankton given by equation [2] will be too large because the limits of the integral are from the surface down to the depth of the euphotic zone. This equation can be modified to incorporate the depth limitation as follows:

$$\Sigma\Sigma P(z') = \frac{P_{opt}\lambda}{\varepsilon}\left[\frac{\delta}{2}\int_{-1}^{1}\int_{\frac{I_o(x)e^{-\varepsilon z'}}{\delta\, I'_k}}^{I_o(x)/\delta I'_k}\frac{1}{[(1+y^2)(1+(ay)^2)^n]^{\frac{1}{2}}}\,dy\,dx\right] \qquad [14]$$

where z' is the linear depth at the station. To compute the total daily carbon fixation for that part of a lake which can be assumed to be horizontally uniform in temperature, transparency and algal biomass, the hypsographic curve A(z), which gives area as a function of depth (Welch 1948) is used as follows:

$$\int_0^{z_m} A(z)\Sigma\Sigma P(z)\;dz \qquad [15]$$

where z_m is the maximum depth of the water body. With only slight modifications, the program presented in Appendix 1 can be used to solve equation [14]. Equation [15] can be solved planimetrically or numerically. For deep lakes, equation [15] of course is just $\underline{A}$ ΣΣP, where $\underline{A}$ is the surface area of the lake and ΣΣP is the solution to equation [2] given by the program in Appendix 1.

The programs presented in Appendices 1, 3 and 4 are completely compatible and should be directly usable on any computer that will compile FØRTRAN IV. As documented in these appendices, there are limits to the number of data points that can be handled, but these limits can be changed by adjusting the DIMENSION statements as desired.

The ALGØL program (Appendix 2) conforms to ALGØL 60 specifications (Naur 1963). However, since input and output statements are not yet standardized, these are different for every ALGØL compiler. Anyone familiar with the FØRMAT statements of FØRTRAN IV will be able to decipher the corresponding ALGØL statements by direct comparison of Appendices 2 and 3. Any unresolved questions can be answered by referring to the Univac 1108 ALGØL 60 reference manual (Univac 1967). The printer plotting routine (PRØCEDURE PLØT) is the only part of this program that contains statements likely to cause real difficulty in conversion. The FØRMAT statements "F3" and "FF" contain variable repeat factors and this procedure also uses STRING variables. These could be converted either by deleting the plotting capability from the program altogether, or by making the repeat factors fixed and changing the alpha data to a representation allowed by the computer being used. This is not a critical part of the program, however, since it is used only for visually comparing the input data with the fitted model. Indeed, Appendix 3, which gives identical results to Appendix 2, does not use the plotting routine. It can be removed from Appendix 2, simply by deleting PRØCEDURE PLØT and the third and fourth lines from the end of the program, which call this procedure. Differences in the character set may be the source of further confusion. Scale factors are indicated by the symbol "&". For example, the number 0.0014 would be written as 1.4 &-3. Multiplication is indicated by the symbol "*" instead of "x", and exponentiation is denoted by "**" instead of "$\uparrow$". Moreover, the relational operators $\geqslant$, $\leqslant$, $\neq$, $=$, $>$ and $<$ are coded as GEQ, LEQ, NEQ, EQL and LSS and GTR, respectively. The last deviation from ALGØL 60 is that in 1108 ALGØL labels referenced before their occurrence must be declared in a LØCAL statement. This occurs at the 14th line from the end of the program and must be removed for compatability with the Revised Report (Naur 1963).

Acknowledgments

Dr. C. H. Mortimer read the manuscript and made many helpful suggestions. R. J. Ristić prepared the figures and K. Lehnhardt typed the manuscript. I was supported by a National Science Foundation Predoctoral Fellowship.

References

Box, M. J. 1966. A comparison of several current optimization methods, and the use of transformations in constrained problems. Computer J., 9: 67-77.

Fee, E. J. 1969. A numerical model for the estimation of photosynthetic production, integrated over time and depth, in natural waters. Limnol. Oceanogr., 14: 906-911.

_______. 1969b. Digital computer programs for spectral analysis of time series. Spec. Rept. No. 6, Center for Great Lakes Studies, Univ. Wisconsin--Milwaukee, 18 pp. + 3 App.

_______. 1971. A numerical model for the estimation of integral primary production, and its application to Lake Michigan. Unpubl. Ph.D. thesis. Univ. Wisconsin Library, Madison.

Fletcher, R. and M. J. D. Powell. 1963. A rapidly convergent descent method for minimization. Computer J., 6: 163-168.

Gaardner, T. and H. H. Gran. 1927. Investigations of the production of plankton in Oslo Fjord. Rapp. et Proc.-verb., Cons. Internat. Explor. Mer, 42: 1-48.

Hutchinson, G. E. 1957. A treatise on limnology. Vol. I. Geography, Physics, and Chemistry, John Wiley and Sons, New York, xiv + 1015 p.

Kiefer, D. and J. D. H. Strickland. 1970. A comparative study of photosynthesis in seawater samples incubated under two types of light attenuator. Limnol. Oceanogr. 15: 408-412.

Lindeman, R. L. 1942. The trophic-dynamic aspect of ecology. Ecology, 23: 399-418.

McAllister, C. D., N. Shah, and J. D. H. Strickland. 1964. Marine phytoplankton photosynthesis as a function of light intensity: a comparison of methods. J. Fish. Res. Bd. Canada, 21: 159-181.

Mood, A. M. and F. A. Graybill. 1963. Introduction to the Theory of Statistics. 2nd. ed. McGraw-Hill, New York, xvi +443 p.

Naur, P. [ed.]. 1963. Revised report on the algorithmic language ALGØL 60. Communications of the Assoc. Comp. Machinery, 6: 1-17.

Rodhe, W., R. A. Vollenweider and A. Nauwerck. 1958. The primary production and standing crop of phytoplankton. In: *Perspectives in Marine Biology*, ed. by A. A. Buzzati-Traverso. pp. 299-322. Univ. California Press, Berkeley.

Smith, E. L. 1936. Photosynthesis in relation to light and carbon dioxide. Proc. Nat'l. Acad. Sci. U.S., 22: 504.

Sorokin, U. I. 1956. (On the use of radioactive carbon C-14 for the study of primary production of water basins)(in Russian) Trudy vsesoyuznovo gidrobiologicheskovo obshchestva, 7: 271-286.

Steemann-Nielsen, E. 1952. The use of radio-active carbon (C-14) for measuring organic production in the sea. J. D. Cons. Internal. Explor. Mer, 18: 117-140.

Talling, J. F. 1957. The phytoplankton as a compound photosynthetic system. New Phytol., 56: 133-149.

___________. 1960. Comparative laboratory and field studies of photosynthesis by a marine planktonic diatom. Limnol. Oceanogr., 5: 62-77.

Univac. 1967. Univac 1108 Multi-processor System ALGØL Programmer's Reference Manual. UP-7544. Sperry-Rand Corp.

Vollenweider, R. A. 1960. Beiträge zur Kenntnis optischer Eigenschaften der Gewässer und Primärproduktion. Mem. Ist. Ital. Idrobiol., 12: 201-244.

___________. 1965. Calculation models of photosynthesis--depth curves and some implications regarding day rate estimates in primary production measurements. Mem. Ist. Ital. Idrobiol., 18 (Suppl.): 425-457.

___________. 1970. Models for calculating integral photosynthesis and some implications regarding structural properties of the community metabolism of aquatic systems. pp. 455-472. In: *Prediction and Measurement of Photosynthetic Productivity*. Proc. IBP/PP Tech. Mtg., Třeboň, 14-21 Sept. 1969.

___________ and A. Nauwerck. 1961. Some observations on the C-14 method for measuring primary production. Verh. Internat. Verien. Limnol., XIV: 134-139.

Welch, P. S. 1948. Limnological Methods. McGraw-Hill, New York, xvii + 381 pp.

Appendix 1

The FORTRAN IV program for computing the double integral of photosynthetic carbon uptake over time and depth is listed in this appendix. The input data are assembled on punched cards in the format listed below. The type of data, i.e., either integer, real, or alpha, follows the description of its position. Further explanation of the input data is given by comments in the program itself.

Card #1:

Col. 1-10: The unit of mass in which the data are coded; this will normally be the milligram (alpha).

Col. 17-20: The unit of length in which the data are coded; usually the meter (alpha).

Card #2:

Col. 1-10: The optimum rate of photosynthesis, P_{opt}(real).

Col. 11-20: The extinction coefficient of the waterbody, $\mathcal{E}$ (real).

Col. 21-30: The daylength in time units the same as those used for irradiance (usually minutes), λ (real).

Col. 31-40: The parameter I'_k (real).

Col. 41-50: The parameter a (real).

Col. 51-60: The parameter n (real).

Card #3:

Col. 1-5: The number of data points on the irradiance curve. This must be an odd number and the rightmost digit must be in column 5 (integer). This number may not exceed 901 unless the DIMENSIØN statement is altered.

Card #4 and following:

The surface irradiance values at the number of data points specified on card #3. Eight data points are punched on each card, each one taking six columns, and being separated from the next value by two spaces, i.e., the format is 8 (F6.3,2X). For successive data sets, repeat cards 2 through 4. Comments in the program indicate the procedures used in the calculations. A set of test data is included at the end of the program listing.

```
C*****THIS PROGRAM COMPUTES THE DAILY INTEGRAL OF PHOTOSYNTHESIS OVER
C     DEPTH IN A BODY OF WATER.
C
      REAL IK,N,LAMBDA
      DIMENSION AP(901),AI(901),D(101),IMASS(10),ILENTH(10),ITITLE(12)
      READ (5,11) IMASS, ILENTH
   11 FORMAT(10A1,10A1)
C
C     IMASS IS THE UNIT OF MASS USED. THIS WILL NORMALLY BE A MILLIGRAM.
C     ILENTH IS THE UNIT OF LENGTH USED.  THIS WILL NORMALLY BE A METER.
C
99    READ(5,13,END=100) POPT,E,LAMBDA,IK,A,N
13    FORMAT(6F10.3)
      C1 = SQRT(((1.-N)*A+SQRT(A*A*(N-1.)*(N-1.)+4.*N))/(2.*A*N))
      DELTA = SQRT((1.+C1*C1)*(1.+A*A*C1*C1)**N)/C1
      READ(5,14) ITITLE,M
14    FORMAT(12A6,I8)
      PRINT 15,ITITLE
15    FORMAT(' ',12A6)
      IF((M/2)*2 .EQ. M) M = M+1
      PRINT 45,M,POPT,E,LAMBDA,IK,A,N,DELTA
   45 FORMAT(' M =',I4/' POPT=',F9.3,/' EXTINCTION COEFFICIENT =',F9.3/
     1 ' DAYLENTH =',F9.3/' I(K)PRIME =',F9.3/' A =',F9.3/ ' N =',F9.3/
     2 ' DELTA =',F9.3/)
C
C     POPT IS THE OPTIMUM RATE OF PHOTOSYNTHESIS.
C     E IS THE EXTINCTION COEFFICIENT OF THE WATERBODY.
C     LAMBDA IS THE DAYLENGTH.
C     IK IS THE P/I CURVE SATURATION PARAMETER.
C     A IS ONE OF THE PHOTOSYNTHESIS INHIBITION PARAMETERS.
C     N IS THE OTHER INHIBITION PARAMETER.
C     DELTA IS THE AUXILLIARY INHIBITION PARAMETER.
C     BE SURE THAT THE UNITS OF ALL THE DATA ARE THE SAME.  ESPECIALLY
C     CHECK THE UNITS USED FOR THE RATE PROCESSES.
C     THE THREE PARAMETERS A, N AND IK MAY BE ESTIMATED WITH ANOTHER
C     COMPUTER PROGRAM PRESENTED IN APPENDIX 2.
C
      READ(5,1,END=100)   (AI(I),I=1,M)
1     FORMAT(13F6.2)
C
C     M IS THE NUMBER OF IRRADIANCE VALUES TO BE USED IN THE TIME
C     INTEGRATION, IT MUST BE AN ODD NUMBER. 200 VALUES
C     WILL BE A SUFFICIENT NUMBER.  MORE THAN THIS MAY BE DESIRABLE IF
C     THE IRRADIANCE CURVE IS HIGHLY IRREGULAR BUT THIS WILL INCREASE
C     THE COMPUTER EXECUTION TIME CONSIDERABLY.
C     AI IS AN ARRAY CONTAINING THE SURFACE IRRADIANCE AT N EQUALLY
C     SPACED TIMES OVER A DAY.
C     THE FOLLOWING STEPS NORMALIZE THE IRRADIANCE DATA.
C
      C1 = IK*DELTA
      DO 3 I=1,M
    3 AI(I) = AI(I) / C1
C
C     LIGHT IS NOW NORMALIZED.
C
      C1 = A*A
      DO 4 J=1,M
      H = 0.01*AI(J)
      Y = 0.0
C
C     THE FOLLOWING STEPS INTEGRATE PHOTOSYNTHESIS OVER DEPTH AT AN
C     INSTANT IN TIME USING SIMPSONS RULE.
C
      DO 5 I=1,101
      AY = Y*Y
      D(I) = 1.0/SQRT((1.0+AY)*(1.0+C1*AY)**N)
    5 Y = Y + H
      AP(J) = D(1)
      DO 6 I=2,98,2
    6 AP(J) = AP(J) + 4.0*D(I) + 2.0*D(I+1)
    4 AP(J) = 0.0033333333*AI(J)*(AP(J) + 4.0*D(100) + D(101))
C
C     THIS COMPLETES THE DEPTH INTEGRATION.
C     THE INTEGRATION OVER TIME FOLLOWS.
C
      DALYPS = AP(1)
      L = M - 3
      DO 7 J=2,L,2
    7 DALYPS = DALYPS + 4.0*AP(J) + 2.0*AP(J+1)
      DALYPS = DELTA*POPT*LAMBDA*0.33333333*(DALYPS + 4.0*AP(M-1) + AP(M
     1))/(FLOAT(M-1)*E)
C
C     THE TIME INTEGRATION IS NOW COMPLETED.
C
      PRINT 10, DALYPS, IMASS, ILENTH
   10 FORMAT(     1X,36HTHE DAILY INTEGRAL PHOTOSYNTHESIS IS,E15.8,1X,10A
     11,1X,10HPER SQUARE , 10A1/)
      PUNCH 8, ITITLE,DALYPS
8     FORMAT(12A6,F8.2)
      GO TO 99
100   STOP
      END
```

A DATA SET THAT MAY BE USED TO TEST THE PROGRAM IS:

```
MILLIGRAM METER
     0.477       .370      940.0       .028       1.000       .254
LAKE MICHIGAN. STATION 1. 25 JUNE 1970.                                              95
   .00    .00    .00    .00    .01    .03    .04    .06    .08    .09    .13    .15    .17
   .21    .22    .22    .25    .28    .30    .33    .35    .37    .39    .41    .43    .45
   .47    .50    .51    .52    .52    .18    .21    .55    .61    .63    .66    .65    .64
   .62    .65    .67    .63    .61    .56    .62    .61    .65    .70    .68    .68    .58
   .67    .69    .70    .68    .67    .62    .59    .57    .56    .53    .46    .39    .50
   .68    .45    .34    .36    .41    .24    .39    .51    .44    .37    .32    .23    .20
   .20    .17    .11    .10    .08    .07    .05    .04    .03    .03    .03    .03    .03
   .02    .01    .01    .01
```

THE CORRECT ANSWER IS 3343.27 MILLIGRAMS CARBON/SQUARE METER PER DAY.

Appendix 2

The ALGØL 60 program, written for the UNIVAC 1108 digital computer, used for estimating the parameters a, n, and I'_k is listed in this appendix. The input data are assembled on data processing cards in the format specified below. The data on the cards can be either integer or real.

Card #1 and following: The following numbers in free format (i.e., separated by one or more spaces).

1. The number of pairs of experimental data points to be used in the computations.
2. P_{opt}, the optimum rate of photosynthetic carbon fixation per unit volume and time.
3. A first estimate of the parameter a.
4. A first estimate of the parameter n.
5. A first estimate of the parameter I'_k.
6. Pairs of data points, each consisting of an irradiance value and the photosynthetic rate corresponding to that irradiance. The program as now written allows a maximum of fifty pairs of such data points. This can be increased by changing the ARRAY declarations in the second line of the program.

As output the program gives the path followed in arriving at the final parameter combination---giving the values of the parameters and the partial derivatives at each iteration---the final fitted photosynthetic rates, and a printer plot showing both the input data and the fitted data.

The program will go through a maximum of fifty iterations in the search for a local minimum. If none is found it simply returns the parameter values given by this number of iterations. The user can either accept this as a working solution or try the computations again with different initial estimates of the parameters. A set of test data, which may be used to assure that the program is functioning properly, is included at the end of the listing.

```
BEGIN
  ARRAY LIT, P,  PEXP  [0:50], X, G[0:4];
  REAL DELTA1,POPT,IK,F,N,A;
  INTEGER NPTS,I,J,K,L,IER;
  PROCEDURE FUNCT(ARG,VAL,GRAD);  REAL VAL;  REAL  ARRAY ARG,GRAD;
    BEGIN
      REAL SUMIK,SUMA,SUMN,IOPT,H,G,DISCR,R,T,U,S,D,E,B,V,Y,DEV,PEXPEK;
      FORMAT F1(3(D8.4,' : '),D9.4,' : ',4(R13.6,' : '),A1);
      SUMIK:=VAL:=SUMA:=SUMN:=0.0;  A:=ARG[2];  N:=ARG[1];  IK:=ARG[3];
      IF N LEQ 0.0 THEN N:=ARG[1]:=1.0&-6;
      IF A LEQ 0.0 THEN A:=ARG[2]:=1.0&-6;
      S:=N-1.0;  T:=A*A;  U:=S*S;
      DISCR:=SQRT(T*U+4.0*N);
      IOPT:=SQRT((-S*A+DISCR)/(2.0*N*A));
      R:=IOPT*IOPT;
      DEV := 1.0+R;  V:=1.0+T*R;  Y:= V**N;
      DELTA1:= IOPT/SQRT(DEV*Y);        H:= 4.0*N*A*IOPT;
      B:=DELTA1*DELTA1;
      D:= (A*U/DISCR - DISCR)/(H*A);
      E:= (N*(T*S+2.0)/DISCR-DISCR-A)/(H*N);
      G:=(B*(N*A*DEV*(V**(N-1.0))*(A*D+IOPT)+Y*D)-D)/IOPT;
      H:=E*((T*N*IOPT/V-1.0/IOPT)+DELTA1*SQRT(Y/DEV))+0.5*LN(V);
      D:=B; E:=T*D; DISCR:=N*A*(A*G-1.0); T:=T*N;
      FOR I:=1 STEP 1 UNTIL NPTS DO
       IF (IOPT:=LIT[I]/IK) GTR 0.0 THEN
        BEGIN
          B:=IOPT*IOPT;  U:=E*B;  V:=1.0+U;  Y:=B*D;  R:=1.0+Y;
          PEXP[I]:=PEXPEK:=IOPT/SQRT(R*(V**N));
          DEV:=PEXPEK-P[I];
          VAL:=VAL + DEV*DEV/PEXPEK;
          S:=PEXPEK-(P[I]*P[I])/PEXPEK;
          B:=1.0/R;
          SUMA:=SUMA+S*Y*(G/R+DISCR/V);
          SUMN:=SUMN+S*(Y*H*(B+T/V)-0.5*LN(V));
          SUMIK:=SUMIK+S*(N*U/V-B)/IK;
        END;
      IF N LEQ 1.0&-5 AND SUMN GTR 0.0 THEN
        BEGIN SUMN:=0.0; SUMA:=1.0&+5*SUMA END;
      IF A LEQ 1.0&-5 AND SUMA GTR 0.0 THEN
        BEGIN SUMA:=0.0; SUMN:=1.0&+5*SUMN END;
      GRAD[1]:=SUMN;   GRAD[2]:=SUMA;  GRAD[3]:=SUMIK;
      WRITE(PRINTER,F1,A,N,1.0/DELTA1,IK,VAL,SUMA,SUMN,SUMIK);
    END COMPUTING THE CHI SQUARED FUNCTION AND ITS PARTIAL DERIVATIVES;
   PROCEDURE HUNT(N,X,F,G,EST,EPS,LIMIT,IER];    VALUE N,EST,EPS,LIMIT;
    REAL      ARRAY X,G;    INTEGER LIMIT,N,IER;  REAL F,EST,EPS;
    BEGIN INTEGER KOUNT,N3,N31,N2,NJ,KL;
      ARRAY H[0:N*(N+7)/2];
      FORMAT F1(I2,A1);
      REAL OLDF,T,DY,HNRM,GNRM,DX,AMBDA,ALFA,DALFA,FX,FY,Z,W;
      FUNCT(X,F,G);  IER:=KOUNT:=0;
      N2:=N+N;  N3:=N2+N;  N31:=N3+1;
  L1:  K:=N31;
      FOR J:=1 STEP 1 UNTIL N DO
      BEGIN H[K]:=1.0;  IF (NJ:=N-J) LEQ 0 THEN GO TO L5;
          FOR L:=1 STEP 1 UNTIL NJ DO H[K+L]:=0.0; K:=K+NJ+1;
        END;
  L5:  WRITE(PRINTER,F1,KOUNT:=KOUNT+1);  OLDF:=F;
```

```
     FOR J:=1 STEP 1 UNTIL N DO
       BEGIN H[K:=N+J]:=G[J]; H[K+N]:=X[J]; K:=J+N3; T:=0;
         FOR L:=1 STEP 1 UNTIL N DO
           BEGIN T:=T-G[L]*H[K]; IF L GEQ J THEN K:=K+1 ELSE K:=K+N-L
           END; H[J]:=T
       END;
     DY:=HNRM:=GNRM:=0.0;
     FOR J:=1 STEP 1 UNTIL N DO
       BEGIN HNRM:=HNRM+ABS(H[J]); GNRM:=GNRM+ABS(G[J]);
         DY:=DY+H[J]*G[J]   END;
     IF DY GEQ 0 OR HNRM/GNRM LEQ EPS THEN GO TO L51;
     ALFA:=2.0*(EST-F)/DY;  AMBDA:=1.0;
     IF ALFA GTR 0.0 AND ALFA LSS AMBDA THEN AMBDA:=ALFA;
     ALFA:=0.0;        FY:=F;
L16: FX:=FY;  DX:=DY;
L17: FOR I:=1 STEP 1 UNTIL N DO
       IF (PEXP[I]:=X[I]+AMBDA*H[I]) LSS -1.0&-6 THEN
         BEGIN AMBDA:=-X[I]/H[I];  GO TO L17  END;
     FOR I:=1 STEP 1 UNTIL N DO X[I]:=PEXP[I];
     FUNCT(X,F,G);  FY:=F;  DY:=0.0;
     FOR I:=1 STEP 1 UNTIL N DO DY:=DY + G[I]*H[I];
     IF DY GTR 0.0 THEN GO TO L22 ELSE
       IF DY EQL 0 THEN GO TO L36 ELSE
         IF FY GEQ FX THEN GO TO L22;
     ALFA:=AMBDA:=AMBDA+ALFA;
     IF HNRM*AMBDA LEQ 1.0&+10 THEN GO TO L16;
     IER:=2;  GO TO XIT;
L22: T:=0.0;
L23: IF AMBDA EQL 0.0 THEN GO TO L36;
     Z:=3.0*(FX-FY)/AMBDA+DY+DX;
     ALFA:=MAX(ABS(Z),ABS(DX),ABS(DY)); DALFA:=Z/ALFA;
     DALFA:=DALFA*DALFA-DX/ALFA*DY/ALFA;
     IF DALFA LSS 0.0 THEN GO TO L51; W:=ALFA*SQRT(DALFA);
     ALFA:=DY-DX+W+W;
     IF ALFA NEQ 0.0 THEN ALFA:=(DY-Z+W)/ALFA ELSE
       ALFA:=(Z+DY-W)/(Z+DX+Z+DY);  ALFA:=ALFA*AMBDA;
L24: FOR I:=1 STEP 1 UNTIL N DO
       IF (PEXP[I]:=X[I] + (T-ALFA)*H[I]) LSS -1.0&-6 THEN
         BEGIN ALFA:=T+X[I]/H[I];  GO TO L24  END;
     FOR I:=1 STEP 1 UNTIL N DO X[I]:=PEXP[I];
     FUNCT(X,F,G);
     IF F LEQ FX AND F LEQ FY THEN GO TO L36;
     DALFA:=0.0;
     FOR I:=1 STEP 1 UNTIL N DO DALFA:=DALFA + G[I]*H[I];
     IF DALFA GEQ 0 OR F GTR FX THEN GO TO L33;
     IF F EQL FX AND DX EQL DALFA THEN GO TO L36;
     FX:=F;  DX:=DALFA;  T:=AMBDA:=ALFA;  GO TO L23;
L33: IF FY NEQ F OR DY NEQ DALFA THEN
       BEGIN FY:=F; DY:=DALFA; AMBDA:=AMBDA-ALFA; GO TO L22  END;
L36: IF OLDF-F+EPS LSS 0 THEN GO TO L51;
     FOR J:=1 STEP 1 UNTIL N DO
       BEGIN  K:=N+J;   H[K]:=G[J]-H[K];
              K:=N+K;   H[K]:=X[J]-H[K] ;

       END;
     IER:=0;
     IF KOUNT LSS N THEN GO TO L42;
     T:=Z:=0.0;
     FOR J:=1 STEP 1 UNTIL N DO
       BEGIN W:=H[K:=N+J]; K:=K+N; T:=T+ABS(H[K]); Z:=Z+W*H[K] END;
     IF HNRM LEQ EPS AND T LEQ EPS THEN GO TO XIT;
L42: IF KOUNT GEQ LIMIT THEN GO TO L50;  ALFA:=0;
     FOR J:=1 STEP 1 UNTIL N DO
       BEGIN   K:=J+N3;   W:=0.0;
        FOR L:=1 STEP 1 UNTIL N DO
         BEGIN W:=W+H[N+L]*H[K]; IF L GEQ J THEN K:=K+1 ELSE K:=K+N-L
         END;
         K:=N+J; ALFA:=ALFA+W*H[K]; H[J]:=W
       END;
     IF Z*ALFA EQL 0.0 THEN GO TO L1;
     K:=N31;
     FOR L:=1 STEP 1 UNTIL N DO
       BEGIN   KL:=N2+L;
        FOR J:=L STEP 1 UNTIL N DO
         BEGIN H[K]:=H[K]+H[KL]*H[N2+J]/Z-H[L]*H[J]/ALFA; K:=K+1 END;
       END;  GO TO L5;
L50: IER:=1; GO TO XIT;
L51: FOR J:=1 STEP 1 UNTIL N DO X[J]:=H[N2+J];   K:=N2+N;
     FUNCT(X,F,G);
     IF GNRM -EPS LEQ 0 THEN BEGIN IER:=0; GO TO XIT END ELSE
     IF IER LSS 0 THEN GO TO XIT; IER:=-1; GO TO L1;
XIT: END FMFP;
 PROCEDURE PLOT(X,Y,Z,XLENTH,YLENTH,N); VALUE N,XLENTH,YLENTH;
   INTEGER N,XLENTH,YLENTH;  ARRAY X,Y,Z;
   BEGIN
     STRING LINE(PAD(10),LASTPART(130));
     INTEGER I,J,K,L,NROWS;
     REAL XMIN,XMAX,YMIN,YMAX,XSCALE,YSCALE,S;
     PROCEDURE IMPROVESCALE(XMIN,XMAX); REAL XMIN,XMAX;
       BEGIN
         XSCALE:=XMAX-XMIN;
         IF XSCALE GTR 0.0 THEN
           BEGIN
             I:=-ENTIER(0.43429448*LN(XSCALE));
             YSCALE:=10.0**I;
             I:=ENTIER(XMIN*YSCALE);
             XMIN:=I/YSCALE;
             S:=0.1/YSCALE;  XSCALE:=ENTIER(XMAX*YSCALE)/YSCALE;
             IF XSCALE LSS XMAX THEN
               FOR XSCALE:=XSCALE+S WHILE XSCALE LSS XMAX DO;
             XMAX:=XSCALE;
           END
       END;
     FORMAT F1(R10.3,'+',S130,A1), F2(X10,'I',S130,A1),
            F3(X6,:XLENTH+1:(R9.2,X1),A1),
            FF(X11,'+',:XLENTH  :('----+----+'),A1);
     XMIN:=XMAX:=X[1]; YMIN:=MIN(Z[1],Y[1]); YMAX:=MAX(Z[1],Y[1]);
     FOR I:=2 STEP 1 UNTIL N DO
       BEGIN
         IF X[I] LSS XMIN THEN XMIN:=X[I];
         IF X[I] GTR XMAX THEN XMAX:=X[I];
         IF Y[I] LSS YMIN THEN YMIN:=Y[I];
         IF Y[I] GTR YMAX THEN YMAX:=Y[I];
         IF Z[I] LSS YMIN THEN YMIN:=Z[I];
         IF Z[I] GTR YMAX THEN YMAX:=Z[I];
```

```
      END;
    IMPROVESCALE(XMIN,XMAX);  IMPROVESCALE(YMIN,YMAX);
    XSCALE:=10.0*XLENTH/(XMAX-XMIN);
    YSCALE:=6.0*YLENTH/(YMAX-YMIN);
    FOR I:=1 STEP 1 UNTIL N DO
      BEGIN
        X[I]:=ENTIER[XSCALE*(X[I]-XMIN)+0.5)+1;
        Y[I]:=ENTIER[YSCALE*(Y[I]-YMIN)+0.5);
        Z[I]:=ENTIER[YSCALE*(Z[I]-YMIN)+0.5);
      END;
    NROWS:=6*YLENTH;
    YSCALE:=5.0/YSCALE; YMAX:=YMAX+YSCALE;
    FOR I:=0 STEP 1 UNTIL NROWS DO
      BEGIN LASTPART:=' ';
        L:=NROWS-I;
        FOR J:=1 STEP 1 UNTIL N DO
          BEGIN
            IF Z[J] EQL L THEN LASTPART(X[J]) := '@' ELSE
            IF Y[J] EQL L THEN LASTPART(X[J]) := '*';
          END;
        IF I EQL NROWS THEN  WRITE(PRINTER,F1,YMIN,LASTPART) ELSE
        IF MOD(I,5) EQL 0 THEN
          WRITE(PRINTER,F1,YMAX:=YMAX-YSCALE,LASTPART) ELSE
          WRITE(PRINTER,F2,LASTPART);
      END;
    WRITE(PRINTER,FF);
    XSCALE:=10.0/XSCALE;
    WRITE(PRINTER,F3,XMIN,FOR I:=1 STEP 1 UNTIL XLENTH DO
      XMIN:=XMIN+XSCALE);
  END PLOT;
FORMAT F0(A3,108('='),A1,':',X3,'A',X4,':',X5,'N',X4,':',X3,
       'DELTA  :',X5,'IK',X4,':',X5,'CHI**2',X4,':',X3,
       'PCHI**2/PA  :',X3,'PCHI**2/PN  :',X3,'PCHI**2/PIK :',A1,
       108('='),A1),
       PAGE(E,'THE NORMALIZED(*) AND FITTED(@) DATA ARE PLOTTED BELOW,
           ,A1,'IF POINTS OVERLAP THEN ONLY AN @ APPEARS',A1.1),
       ENDPLOT(X47,'NORMALIZED IRRADIANCE, I/(DELTA*IK)',A2),
F1(E,'NDATA =',I3,X2,'POPT =',D10.6,X2,'IK =',D10.6,
          X2,'A=',D10.6,X2,'N=',D10.6,A1.2,X20,
        'LIGHT VS. PHOTOSYNTHESIS DATA FOLLOW...',A1.1,(8D10.5,A1)),
       F3('THE NORMALIZED AND FITTED DATA FOLLOW ...',A2.1,45('='),A1,
          ':I/(DELTA*IK) :  POBS/POPT   :   P(FITTED)  :',A1,
          ' I/(DELTA*IK) :',X2,'POBS/POPT',X3,':',X3,'P(FITTED)',A1,
         45('='),A1,(D10.5,X4,':',D12.8,X2,':',D12.8,A1));
LOCAL LABEL EOF, LOOP;
LIST INPUTLIST(NPTS,POPT,IK,A,N,FOR I:=1 STEP 1 UNTIL NPTS DO
       (LIT[I], P[I]));
LOOP:  READ(CARDS,EOF,INPUTLIST);  WRITE(PRINTER,F1,INPUTLIST);
WRITE(PRINTER,F0);
FOR I:=1 STEP 1 UNTIL NPTS DO P[I]:=P[I]/POPT;
X[1]:=N;  X[2]:=A;  X[3]:=IK;
HUNT(3,X,F,G,1.0&-6,1.0&-6, 50,IER);
WRITE(PRINTER,F3,FOR I:=1 STEP 1 UNTIL NPTS DO ((LIT[I]:=LIT[I]*
         DELTA1/IK),P[I],PEXP[I]));
WRITE(PRINTER,PAGE); PLOT(LIT,P,PEXP,11,8,NPTS);
WRITE(PRINTER,ENDPLOT);
GO TO LOOP;
```

```
EOF: END.

A DATA SET THAT MAY BE USED TO TEST THE PROGRAM IS:

20 100.0 0.02 0.5 0.5
0.17  90.8  0.0516 100.0  0.0146  79.5  0.0072  46.4
0.17  83.6  0.0516 100.0  0.0146  87.5  0.0072  49.5
0.17  88.9  0.0516 100.0  0.0146  89.7  0.0072  50.5
0.17  75.9  0.0516 100.0  0.0146  81.2  0.0072  50.5
0.17  81.2  0.0516 100.0  0.0146  84.4  0.0072  46.7

THE CORRECT ANSWERS ARE: A=1.0000, N=0.1806, IK=0.0128 AND CHI**2=0.0411.
```

Appendix 3

A FØRTRAN IV program for estimating the model parameters $\underline{a}$, $\underline{n}$ and I'_k is listed with comments in this appendix. Unlike the ALGØL version given in Appendix 2, this program does not include the printer plotting capability. The comments given in the program apply also to the corresponding statements in the ALGØL program.

The format of the input data is given below; the type of data, i.e., either real or integer, is given in parentheses following its description.

Card #1:

Col. 1-5: The number of pairs of pairs of data points available for fitting the curve, (integer).

Col. 6-15: P_{opt}, the optimum rate of carbon uptake per unit volume and time (real).

Col. 16-25: A first estimate of the model parameter I'_k (real).

Col. 26-35: A first estimate of the model parameter $\underline{a}$ (real).

Col. 36-45: A first estimate of the model parameter $\underline{n}$ (real).

Cards #2 and following: (there must be as many cards as the number in columns 1-5 of the first card).

Col. 1-10: An experimental irradiance value (real).

Col. 11-20: The rate of photosynthesis per unit volume and time corresponding to the irradiance value in columns 1-10 (real). Be sure that the units of these rates are the same as the units used for P_{opt}.

To process successive data sets, place card decks after the initial one in the format specified. The same data set that was given at the end of Appendix 2 may be used to check the proper operation of this program.

```
      REAL LIT(50),P(50),PEXP(50),X(4),G(4),DELTA1,POPT,IK,F,A,N
      COMMON LIT,P,DELTA1,POPT,IK,A,NPTS,N
      COMMON /XX/ PEXP
10    FORMAT(///,1X,108('='),/' :',3X,'A',4X,':',5X'N',4X,':',3X,
     $ 'DELTA  :',5X,'IK',4X,':',5X,'CHI**2',4X,':',3X,'PCHI**2/PA  :',
     $ 3X,'PCHI**2/PN  :',3X,'PCHI**2/PIK :'/1X,108('='))
11    FORMAT('1NDATA =',I3,2X,'POPT =',F10.6,2X,'IK =',F10.6,2X,
     $ 'A =',F10.6,2X,'N =',F10.6//20X,'LIGHT VS PHOTOSYNTHESIS DATA FOL
     $LOW...'/(8F10.5))
12    FORMAT(' THE NORMALIZED AND FITTED DATA FOLLOW...'//1X,45('=')/
     $ ':I/(DELTA*IK) :    POBS/POPT   :   P(FITTED)  :'/1X,45('=')/
     $ (F10.5,4X,':',F12.8,2X,':',F12.8))
20    FORMAT(I5,4F10.2/2F10.2)
C     READ AND PRINT THE INPUT DATA
1     READ(5,20,END=100) NPTS,POPT,IK,A,N,(LIT(I),P(I),I=1,NPTS)
      PRINT 11,          NPTS,POPT,IK,A,N,(LIT(I),P(I),I=1,NPTS)
      PRINT 10
C     NORMLIZE THE PHOTOSYNTHETIC RATES BY POPT TO GET RELATIVE RATES.
      DO 2 I=1,NPTS
2     P(I) = P(I)/POPT
      X(1) = N
      X(2) = A
      X(3) = IK
C     FIND THE MINIMUM CHI**2 VALUE AND BEST PARAMETER COMBINATION.
      CALL HUNT(3,X,F,G,1.0E-6,1.0E-6,50,IER)
C     NORMALIZE THE LIGHT DATA
      DO 3 I=1,NPTS
3     LIT(I) = LIT(I)*DELTA1/IK
C     PRINT THE ANSWERS.
      PRINT 12,(LIT(I),P(I),PEXP(I),I=1,NPTS)
      GO TO 1
100   STOP
      END



      SUBROUTINE FUNCT(ARG,VAL,GRAD)
      REAL LIT(50),P(50),PEXP(50),ARG(1),GRAD(1),IK,N
      COMMON LIT,P,DELTA1,POPT,IK,A,NPTS,N
      COMMON /XX/ PEXP
      DOUBLE PRECISION SUMIK,SUMN,SUMA,S,T,U,DISCR,IOPT,R,DEV,V,Y,H,B,D,
     $ E,G,PEXPEK
      DOUBLE PRECISION EELTA1
1     FORMAT(' ',3(F8.4,' : '),F9.4,' : ',4(E13.6,' : '))
      VAL = 0.0
      SUMIK = 0.0
      SUMA = 0.0
      SUMN = 0.0
C     CHECK TO ASSURE THAT CONSTRAINTS ARE OBSERVED.
      IF(ARG(1) .LE. 0.0) ARG(1) = 1.0E-06
      IF(ARG(2) .LE. 0.0) ARG(2) = 1.0E-06
      N = ARG(1)
      A = ARG(2)
      IK = ARG(3)
C     COMPUTE VARIOUS CONSTANTS TO BE USED IN THE ITERATION LOOP.
      S = N-1.0
      T = A*A
      U = S*S
      DISCR =DSQRT(T*U + 4.0*N)
      IOPT =DSQRT((-S*A+DISCR)/(2.0*N*A))
      R = IOPT*IOPT
      DEV = 1.0+ R
      V = 1.0+ T*R
      Y = V**N
      EELTA1 = IOPT/DSQRT(DEV*Y)
      H = 4.0*N*A*IOPT
      B = EELTA1*EELTA1
      D = (A*U/DISCR-DISCR)/(H*A)
      E = (N*(T*S+2.0)/DISCR-DISCR-A)/(H*N)
      G = (B*(N*A*DEV*(V**(N-1.0))*(A*D+IOPT)+Y*D)-D)/IOPT
      H = E*((T*N*IOPT/V-1.0/IOPT)+EELTA1*DSQRT(Y/DEV))+0.5* ALOG(V)
      D = B
      E = T*D
      DISCR = N*A*(A*G-1.0)
      T = T*N
C     START OF THE LOOP FOR COMPUTING CHI**2 AND ITS PARTIAL DERIVATIVES
      DO 4 I=1,NPTS
      IOPT = LIT(I)/IK
      IF(IOPT .LE. 0.0) GO TO 4
      B = IOPT*IOPT
      U = E*B
      V = 1.0+U
      Y = B*D
      R = 1.0+Y
C     COMPUTE THE EXPECTED PRODUCTION FOR THE GIVEN IRRADIANCE.
      PEXPEK = IOPT/DSQRT(R*(V**N))
      PEXP(I) = PEXPEK
C     COMPUTE THE DEVIANCE OF THE OBSERVED FROM THE EXPECTED PRODUCTION.
      DEV = PEXPEK - P(I)
C     COMPUTE THE CHI**2 VALUE.
      VAL = VAL + DEV*DEV/PEXPEK
      S = PEXPEK - (P(I)*P(I))/PEXPEK
      B = 1.0/R
C     COMPUTE THE PARTIAL DERIVATIVE OF CHI**2 WITH RESPECT TO A.
      SUMA = SUMA + S*Y*(G/R + DISCR/V)
C     COMPUTE THE PARTIAL DERIVATIVE OF CHI**2 WITH RESPECT TO N.
      SUMN = SUMN + S*(Y*H*(B + T/V) - 0.5* ALOG(V))
C     COMPUTE THE PARTIAL DERIVATIVE OF CHI**2 WITH RESPECT TO IK.
      SUMIK = SUMIK + S*(N*U/V - B)/IK
4     CONTINUE
C     SEE WHETHER THE PARTIAL OF CHI**2 WITH RESPECT TO N POINTS OUT OF
C      THE REGION. IF SO, SET IT TO ZERO AND INCREASE THE PARTIAL WITH
C      RESPECT TO A BY FIVE ORDERS OF MAGNITUDE.
      IF(N.LE.1.0E-5.AND.SUMN.GT.0.) GO TO 5
      GO TO 6
5     SUMN = 0.
      SUMA = 1.0E+5 * SUMA
C     SEE WHETHER THE PARTIAL OF CHI**2 WITH RESPECT TO A POINTS OUT OF
C      THE REGION. IF SO, SET IT TO ZERO AND INCREASE THE PARTIAL WITH
C      RESPECT TO N BY FIVE ORDERS OF MAGNITUDE.
6     IF(A.LE.1.0E-5.AND.SUMA.GT.0.) GO TO 7
      GO TO 8
7     SUMA = 0.
      SUMN = 1.E+5*SUMN
```

```
8     GRAD(1) = SUMN
      GRAD(2) = SUMA
      GRAD(3) = SUMIK
      DELTA1 = EELTA1
      V = 1./DELTA1
C     PRINT THE COMPUTED VALUES.
      PRINT 1, A,N,V,IK,VAL,SUMA,SUMN,SUMIK
      RETURN
      END


      SUBROUTINE HUNT(N,X,F,G,EST,EPS,LIMIT,IER)
      DIMENSION H(15),X(1),G(1),PEXP(1)
      COMMON /XX/ PEXP
      CALL FUNCT(  X,F,G)
      IER=0
      KOUNT=0
      N2=N+N
      N3=N2+N
      N31=N3+1
1     K=N31
      DO 4 J=1,N
      H(K)=1.
      NJ=N-J
      IF(NJ)5,5,2
2     DO 3 L=1,NJ
      KL=K+L
3     H(KL)=0.
4     K=KL+1
C     START OF THE ITERATION LOOP.
5     KOUNT=KOUNT +1
      PRINT 60, KOUNT
60    FORMAT(' ITERATION NUMBER 'I3)
C     SAVE THE CHI**2 VALUE, PARAMETER VECTOR AND GRADIENT VECTOR.
      OLDF=F
      DO 9 J=1,N
      K=N+J
      H(K)=G(J)
      K=K+N
      H(K)=X(J)
C     DETERMINE THE DIRECTION VECTOR.
      K=J+N3
      T=0.
      DO 8 L=1,N
      T=T-G(L)*H(K)
      IF(L-J)6,7,7
6     K=K+N-L
      GO TO 8
7     K=K+1
8     CONTINUE
9     H(J)=T
C     CHECK WHETHER CHI**2 WILL DECREASE BY STEPPING ALONG H VECTOR.
      DY=0.
      HNRM=0.
      GNRM=0.
C     CALCULATE THE DIRECTIONAL DERIVATIVE AND THE TEST VALUES FOR THE
```

```
C      DIRECTION VECTOR H AND THE GRADIENT VECTOR G.
      DO 10 J=1,N
      HNRM=HNRM+ABS(H(J))
      GNRM=GNRM+ABS(G(J))
10    DY=DY+H(J)*G(J)
C     REPEAT THE SEARCH IN THE DIRECTION OF STEEPEST DESCENT IF THE
C      DIRECTIONAL DERIVATIVE IS POSITIVE OR ZERO.
      IF(DY)11,51,51
C     REPEAT THE SEARCH IN THE DIRECTION OF THE STEEPEST DESCENT IF THE
C      DIRECTION VECTOR H IS SMALL COMPARED TO THE GRADIENT VECTOR G.
11    IF(HNRM/GNRM-EPS)51,51,12
C     SEARCH ALONG H FOR THE POSITIVE DIRECTIONAL DERIVATIVE.
12    FY=F
      ALFA=2.*(EST-F)/DY
      AMBDA=1.
C     USE THE ESTIMATE FOR THE STEPSIZE ONLY IF IT IS POSITIVE AND LESS
C      THAN UNITY. OTHERWISE, USE 1.0 AS THE STEPSIZE.
      IF(ALFA)15,15,13
13    IF(ALFA-AMBDA)14,15,15
14    AMBDA=ALFA
15    ALFA=0.
C     SAVE CHI**2 AND THE DERIVATIVES FOR THE OLD PARAMETER VECTOR.
16    FX=FY
      DX=DY
C     STEP THE PARAMETER VECTOR ALONG H.
119   CONTINUE
      DO 17 I=1,N
      PEXP(I) = X(I) + AMBDA*H(I)
      IF(PEXP(I) .GE. -1.E-6) GO TO 17
      AMBDA = -X(I)/H(I)
      GO TO 119
17    CONTINUE
      DO 118 I=1,N
118   X(I) = PEXP(I)
C     COMPUTE CHI**2 AND THE GRADIENT VECTOR FOR THE NEW PARAMETERS.
      CALL FUNCT(  X,F,G)
      FY=F
C     COMPUTE THE DIRECTIONAL DERIVATIVE DY FOR THE NEW PARAMETERS.
C      TERMINATE THE SEARCH IF DY IS POSITIVE. IF DY IS ZERO, THE
C      MINIMUM OF THE PARAMETER SPACE HAS BEEN FOUND.
      DY=0.
      DO 18 I=1,N
18    DY=DY+G(I)*H(I)
      IF(DY)19,36,22
C     TERMINATE SEARCH IF THE CHI**2 VALUE INDICATES THAT THE MINIMUM
C      HAS BEEN FOUND.
19    IF(FY-FX)20,22,22
C     DOUBLE THE STEPSIZE AND REPEAT THE SEARCH.
20    AMBDA=AMBDA+ALFA
      ALFA=AMBDA
C     END OF THE SEARCH LOOP. TERMINATE IF THE CHANGE IN CHI**2 IS
C      VERY LARGE.
      IF(HNRM*AMBDA-1.E10)16,16,21
C     LINEAR SEARCH INDICATES THAT NO MINIMUM EXISTS.
21    IER=2
      RETURN
C     INTERPOLATE CUBICALLY IN THE INTERVAL DEFINED BY THE SEARCH TO
```

```
C      FIND THE BEST PARAMETER VECTOR.
22    T=0.
23    IF(AMBDA)24,36,24
24    Z=3.*(FX-FY)/AMBDA+DY+DX
      ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY))
      DALFA=Z/ALFA
      DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA
      IF(DALFA)51,25,25
25    W=ALFA*SQRT(DALFA)
      ALFA=DY-DX+W+W
      IF (ALFA) 250,251,250
250   ALFA=(DY-Z+W)/ALFA
      GO TO 252
251   ALFA=(Z+DY-W)/(Z+DX+Z+DY)
252   ALFA=ALFA*AMBDA
120   CONTINUE
      DO 26 I=1,N
      PEXP(I) = X(I) + (T-ALFA)*H(I)
      IF(PEXP(I) .GE. -1.E-06) GO TO 26
      ALFA = T + X(I)/H(I)
      GO TO 120
26    CONTINUE
      DO 121 I=1,N
121   X(I) = PEXP(I)
C     COMPUTE CHI**2 AND THE GRADIENT VECTOR AT THE NEW PARAMETERS.
      CALL FUNCT(  X,F,G)
C     TERMINATE IF CHI**2 IS LESS THAN CHI**2 VALUES AT THE END OF THE
C      INTERVAL. OTHERWISE, REDUCE THE INTERVAL BY SETTING ONE END POINT
C      EQUAL TO THE COMPUTED PARAMETER VECTOR AND REPEAT THE
C      INTERPOLATION. THE CHOICE OF WHICH END POINT TO USE DEPENDS ON
C      THE CHI**2 VALUE AND ITS GRADIENT AT THE OLD PARAMETER VALUES.
      IF(F-FX)27,27,28
27    IF(F-FY)36,36,28
28    DALFA=0.
      DO 29 I=1,N
29    DALFA=DALFA+G(I)*H(I)
      IF(DALFA)30,33,33
30    IF(F-FX)32,31,33
31    IF(DX-DALFA)32,36,32
32    FX=F
      DX=DALFA
      T=ALFA
      AMBDA=ALFA
      GO TO 23
33    IF(FY-F)35,34,35
34    IF(DY-DALFA)35,36,35
35    FY=F
      DY=DALFA
      AMBDA=AMBDA-ALFA
      GO TO 22
C     TERMINATE IF CHI**2 HAS NOT DECREASED DURING THE LAST ITERATION.
36    IF(OLDF-F+EPS)51,38,38
C     COMPUTE THE DIFFERENCE VECTORS OF THE PARAMETER AND GRADIENT
C      VECTORS FROM TWO CONSECUTIVE ITERATIONS.
38    DO 37 J=1,N
      K=N+J
      H(K)=G(J)-H(K)
```

```
        K=N+K
 37     H(K)=X(J)-H(K)
        IER=0
 C      IF AT LEAST 3 ITERATIONS HAVE BEEN COMPUTED, TEST THE LENGTH OF
 C       THE PARAMETER VECTOR AND THE DIRECTION VECTOR.
        IF(KOUNT-N)42,39,39
 39     T=0.
        Z=0.
        DO 40 J=1,N
        K=N+J
        W=H(K)
        K=K+N
        T=T+ABS(H(K))
 40     Z=Z+W*H(K)
        IF(HNRM-EPS)41,41,42
 41     IF(T-EPS)56,56,42
 C      TERMINATE IF THE NUMBER OF ITERATIONS WOULD EXCEED LIMIT.
 42     IF(KOUNT-LIMIT)43,50,50
 C      PREPARE TO UPDATE THE H MATRIX.
 43     ALFA=0.
        DO 47 J=1,N
        K=J+N3
        W=0.
        DO 46 L=1,N
        KL=N+L
        W=W+H(KL)*H(K)
        IF(L-J)44,45,45
 44     K=K+N-L
        GO TO 46
 45     K=K+1
 46     CONTINUE
        K=N+J
        ALFA=ALFA+W*H(K)
 47     H(J)=W
 C      REPEAT THE SEARCH IN THE DIRECTION OF STEEPEST DESCENT IF THE
 C       RESULTS ARE NOT SATISFACTORY.
        IF(Z*ALFA)48,1,48
 C      UPDATE THE H MATRIX.
 48     K=N31
        DO 49 L=1,N
        KL=N2+L
        DO 49 J=L,N
        NJ=N2+J
        H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA
 49     K=K+1
 C      END OF THE ITERATION LOOP.
        GO TO 5
 C      NO CONVERGENCE AFTER SPECIFIED NUMBER OF ITERATIONS.
 50     IER=1
        RETURN
 C      RESTORE THE OLD CHI**2 VALUE AND PARAMETER VALUES.
 51     DO 52 J=1,N
        K=N2+J
 52     X(J)=H(K)
        CALL FUNCT(  X,F,G)
 C      REPEAT SEARCH IN THE DIRECTION OF THE STEEPEST DESCENT IF THE
 C       DERIVATIVE FAILS TO BE SUFFICIENTLY SMALL.
```

```
      IF(GNRM-EPS)55,55,53
C     TEST FOR REPEATED FAILURE OF ITERATION.
53    IF(IER)56,54,54
54    IER=-1
      GOTO 1
55    IER=0
56    RETURN
      END
```

Appendix 4

A FØRTRAN IV program for converting analog charts digitized with a mechanical digitizer to digital output equally spaced in time (or space) is listed in this appendix. The program smooths data points by using the trapezoidal rule for numerical integration to obtain an average value for each equally spaced data point output. As output, the program prints both the X and Y coordinates but punches only the Y coordinates since the X's are equally spaced. The format of the input is documented below with the type of data, i.e., real, integer or alpha, in parentheses.

Card #1:

Col. 1-10: The number of units desired between successive abscissa values (real).

Col. 11-20: The scale factor for converting digitizer units to actual ordinate units (real). This is the number of digitizer units that equals one unit of the physical quantity being digitized.

Col. 21-30: The maximum allowable difference between successive ordinates in the input data (real). This is used to insure that changes are not too abrupt and that the machine did not make a mechanical error (e.g., drop a digit).

Card #2:

Col. 1-4: Right justified, the number of data points in the deck of cards that represents the digitized chart (integer). This may not exceed 5000 unless the DIMENSIØN statements are altered.

Card #3:

Col. 1-5: The number of abscissa units spanned by the digitized record (real).

Card #3 (continued):

Col. 6-80: Any series of symbols that can be used to identify this data set (alpha). It will be used to label the output.

Card #4 and following, as necessary:

The digitized record, each point consists of an X and a Y coordinate (integer). Each part takes up 4 columns and there are no decimal points or blanks. There are 10 coordinates per card.

Cards 2 through 4 are repeated for as many data sets as desired. The total number of data points output (which equals 1 plus the total time span of the record divided by the increment between successive data points) may not exceed 1000.

```
C ** A PROGRAM FOR SMOTHING AND INTERPOLATING DIGITIZED DATA. E. J. FEE.
      DIMENSION X(5000),Y(5000),ZX(1000),ZY(1000),      IALPHA(37)
33    FORMAT(13F6.2)
   92 READ(5,91,END=41)                    AINCR,AK,ALIMIT
91    FORMAT(3F10.2)
C**** AINCR IS THE STEP DESIRED BETWEEN SUCCESSIVE ABSCISSA VALUES.
C**** AK IS THE SCALING FACTOR FOR THE ORDINATES, I.E.:
C        AK = ABSOLUTE VERTICAL AMPLITUDE/RELATIVE VERTICAL AMPLITUDE
C**** ALIMIT IS THE MAXIMUM ALLOWABLE DIFFERENCE IN ORDINATE VALUES.
      BINCR = 1.0/AINCR
    2 READ(5,4,END=41) N
    4 FORMAT (I4)
      READ 3,D,IALPHA
C**** N IS THE NUMBER OF DIGITIZED DATA POINTS TO BE INPUT.
C**** D IS THE NUMBER OF ABSCISSA UNITS COVERED BY THE DIGITIZED RECORD.
C        THAT IS, IT IS THE SPAN OF TIME OR LENGTH FROM THE FIRST POINT
C        TO THE LAST POINT IN THE INPUT DECK.
C**** IALPHA IS ANY SERIES OF 74 CHARACTERS; IT IS USED TO LABEL THE
C        OUTPUT.
    3 FORMAT(F5.2,    37A2)
      READ (5,6) (X(I),Y(I),I=1,N)
C**** X AND Y ARE THE DIGITIZED COORDINATES.
6     FORMAT(20F4.0)
      IERR1 = 0
      IERR2 = 0
C**** MAKE SURE THE ENDS ARE IN ORDER.
      IF(X(1)-X(2)) 101,101,100
100   T=X(1)
      X(1)=X(2)
      X(2)=T
      T=Y(1)
      Y(1)=Y(2)
      Y(2)=T
101   IF(X(N)-X(N-1)) 103,104,105
104   N=N-1
      GO TO 101
103   T=X(N)
      X(N)=X(N-1)
      X(N-1)=T
      T=Y(N)
      Y(N)=Y(N-1)
      Y(N-1)=T
105   AL = D/(X(N) - X(1))
      E = X(1)
      F = Y(1)
C**** TRANSFORM THE DATA FROM DIGITIZER UNITS TO ABSOLUTE UNITS
      DO 7 I=1,N
      X(I) = AL*(X(I) - E)
7     Y(I) = (Y(I) - F)/AK
      X(N) = D
      LBJ = N-1
C**** MAKE SURE THAT THE ABSCISSAS ARE IN ORDER
      DO 9 I=2,LBJ
      IF(X(I)-X(I-1))8,99,99
C**** THE DIGITIZER DOUBLED BACK SO AVERAGE OUT THE POINT
    8 X(I) = (X(I-1) + X(I+1))*0.5
      Y(I) = (Y(I-1) + Y(I+1))*0.5
```

```
C**** TALLY THE NUMBER OF ERRORS IN THE ABSCISSA VALUES
      IERR1 = IERR1 + 1
C**** CHECK TO MAKE SURE THE ORDINATES ARE OK
   99 G = Y(I)-Y(I-1 )
      IF(ABS(G) -  ALIMIT)  9,9,10
C**** THERE WAS TOO MUCH OF A CHANGE BETWEEN SUCCESSIVE POINTS SO
C****   AVERAGE THE POINT OUT
   10 Y(I) = Y(I-1)+((Y(I+1)-Y(I-1))*(X(I)-X(I-1)))/(X(I+1)-X(I-1))
C**** COUNT THE NUMBER OF ERRORS IN THE ORDINATE VALUES
      IERR2 = IERR2 + 1
    9 CONTINUE
C**** HOW MANY EQUALLY-SPACED TIME VALUES ARE TO BE COMPUTED...
      K = D/AINCR + 1.0
      ZX(1) = 0.0
C**** INTEGRATE THE RECORD TO AVERAGE OUT HIGH FREQUENCIES.
      I=1
      J=1
75    I=I+1
76    J=J+1
      SUM=0.0
      IF(J-K) 77,77,78
77    ZX(J)=ZX(J-1) + AINCR
74    IF( X(I)-ZX(J)) 71,72,73
71    SUM=SUM+ 0.5*(Y(I)+Y(I-1))*(X(I)-X(I-1))
      I=I+1
      GO TO 74
72    ZY(J) = BINCR*(SUM+0.5*(Y(I)+Y(I-1))*(X(I)-X(I-1)))
      GO TO 75
73    SLOPE=(Y(I)-Y(I-1))/(X(I)-X(I-1))
      AINTER=Y(I) - SLOPE*X(I)
      YI=SLOPE*ZX(J) + AINTER
      ZY(J)=BINCR*(SUM+0.5*(YI+Y(I-1))*(ZX(J)-X(I-1)))
      X(I-1)=ZX(J)
      Y(I-1)=YI
      GO TO 76
78    CONTINUE
      DO 80 I=1,K
      IF(ZY(I)) 81,80,80
81    ZY(I) = 0.
80    CONTINUE
      PRINT 20,    IALPHA,IERR1,IERR2,(ZX(I),ZY(I),I=1,K)
   20 FORMAT(1H0,                               37A2/5X,I3,1X,49HVALUES OF
     1 THE TIME VARIABLE WERE OUT OF SEQUENCE./5X,I3,1X,           33HD
     1ATA POINTS WERE OUT OF SEQUENCE./(12F10.3))
      PUNCH 933, IALPHA,K
933   FORMAT(37A2,I6)
      PUNCH 33, (ZY(NIXON),NIXON=1,K)
      GO TO 2
   41 STOP
      END
```

A DATA SET THAT MAY BE USED TO TEST THE PROGRAM IS:

```
10.0       1260.97   1000.0
 163
940.0MILWAUKEE SOLARIMETER. JUNE 25, 1970. UNITS ARE LY/MIN (CORRECTED).
```

```
15541038164510331726105217791069187011121944114719541124197711612004114920431224
21421242221531312219312992216132122491321227012782298134023101338235713692483145 5
```

```
15541038164510331726105217791069187011121944114719541124197711612004114920431224
21421242221531312219312992216132122491321227012782298134023101338235713692483 1455
26231523270015652779161027851585283816412900166929531679300516973054171231091000
31891796323518073257180633541877342018343436184735111816355118713567185236041912
36301743364418703664185936751895368418133692188737121662375218053756167837761822
38011759381119143840183838711659389419583903190239371949397218823992186440251962
40611856408116074094178241011850410517974118191141301862414519044153188441651894
41731866417919084202190942091926425419274284192542911855432619294354192943741640
43881880440418504408188644141796442218874432192044471803446316284476167244871848
45141697453117504552164045671896459816524603161146141715463015814634164146471505
46521716466616124677162946811581468616784715139747321377474415574755186347811962
48191830484414544857164948731651488315424895138949141397492515614933141549541414
49771764497916084987172549951503500316045016144750411322506013225069138350791331
50911422509513995114170851201680512517465151116725201162652091381521315955222156 8
52321596526914155282152752861491529814895322133153421393535412865388133154121265
54341252544313085464128954741327551312175556117456001171574011055847107959541075
60721075616310466284104 4

THE CORRECT ANSWERS ARE:

    .00    .00    .00    .00    .01    .03    .04    .06    .08    .09    .13    .15    .17
    .21    .22    .22    .25    .28    .30    .33    .35    .37    .39    .41    .43    .45
    .47    .50    .51    .52    .52    .18    .21    .55    .61    .63    .66    .65    .64
    .62    .65    .67    .63    .61    .56    .62    .61    .65    .70    .68    .68    .58
    .67    .69    .70    .68    .67    .62    .59    .57    .56    .53    .46    .39    .50
    .68    .45    .34    .36    .41    .24    .39    .51    .44    .37    .32    .23    .20
    .20    .17    .11    .10    .08    .07    .05    .04    .03    .03    .03    .03    .03
    .02    .01    .01    .01
```