DRAFT

Software for Use in Calculation

of

Primary Production

in the

Oceanic Water Column

by

Trevor Platt & Shubha Sathyendranath

Software for Use in Calculation of Primary Production in the Oceanic Water Column

Address: Biological Oceanography Division, Bedford Institute of Oceanography, Box 1006, Dartmouth, N. S., Canada B2Y 4A2

September, 1995

Contents

List of Figures

Daily, Water-Column Production – Empirical Estimators

List of Tables

Preface

The programs presented here are based on those written by us and used in our own work for the past ten years or so. For the most part, they rely on theory that we have developed ourselves.

The programs have been packaged and documented for use in the broader oceanographic community by George White, to whom we are most grateful. We are also indebted to our colleagues and students who have tested, evaluated, and by constructive suggestions improved, their value (Linda Payzant, Carla Caverhill, Venetia Stewart, Nick Hoepffner, Heidi Maass, Margareth Kyewalyanga, and Cathy Porter).

Distribution of these programs without a requirement to pay license fees was made possible through the availability of reliable "free" software. We are grateful to the many individuals who have contributed to the programs used in creating this package (Appendix B). Users are reminded that use of certain of these programs is subject to license restrictions. Users should read and obey the terms of the licenses included in the software distribution.

Users are encouraged to report problems with the software and to suggest improvements or additions for future releases. Also, users are requested to cite this book in publications resulting from its use.

Trevor Platt, Shubha Sathyendranath

Dartmouth, Nova Scotia

Canada

September, 1995

Chapter 1

Introduction

1.1 Introduction

This document describes methods and software for the calculation of oceanic primary production and related properties. Programs are provided to permit calculation by a variety of methods, including those based on remotely-sensed data. Such calculations may be performed on many space and time scales. The basic unit of calculation is the daily production on a spatial scale determined by the input data. For example, the methods described here are suitable for use in basin-scale production estimates using a combination of satellite and shipboard observations.

The methods behind the algorithms presented here have already been published in the open literature. There is, however, a large gap between the formal description of a method and a practical implementation. The purpose of the current document is to provide such an implementation.

1.2 Goals

The purpose of the package is to make the methods described below available to as many users as possible. Potential users are not assumed to have expert knowledge of numerical analysis and computational methods. Since there are many possible variations within the basic framework, the software needs to balance the competing requirements of transparency, portability, generality, and efficiency.

At the current level of development of the algorithms and the support data, a major role of the software will be to explore the sensitivity of the production values to various refinements in the models and to test approximations designed to reduce computational costs. The programs should be directly useful in compiled form and should provide a suitable basis for further development and experimentation with new models, algorithms, and numerical procedures.

1.3 Software for daily production calculations

The software consists of the following key programs: a nonlinear parameter-estimation utility (gfit), surface irradiance (i0), average concentration for a layer (avgc), [planned for a future release: surface biomass (b0)], daily water-column production using various methods (dwcpa, dwcpn, and dwcpe), and [planned for a future release: ocean-colour biomass (ocb)]. The functions of these programs are as follows:

gfit: General-purpose, interactive, nonlinear, parameter-estimation module used to extract the profile parameters when given a series of chlorophyll biomass observations, $B(z)$, as a function of depth, z, or to extract the parameters α^B and P_m^B when given a series of observations of photosynthesis, $P^B(I)$, at varying light intensities, I. Although this implementation does not require expensive hardware, it will, nevertheless, give useful results. Some users may, however, find it helpful to substitute a more robust parameter-estimation

package. The models are described below in sufficient detail that it should not be difficult to configure other parameter-estimation software to replace gfit.

i0: Given a location and date, and assuming clear sky, compute the time of sunrise, sunset, noon irradiance, total daily irradiance, and surface irradiance through the day, $I_0(t)$. Summarize the result in terms of total daily irradiance, I_t and noon irradiance, I_*^m.

avgc: Given estimates for the parameters of the shifted-Gaussian profile, calculate the average attenuation coefficient for any arbitrary interval of depth (or, equivalently, find the average biomass for that interval). [In a future release, the program will support additional standardized profile models (triangular) and will determine the photic depth.]

b0: Recompute B_0 from CZCS climatology given information on the shape of the depth profile of biomass, $B(z)$. [This program is not included in the current distribution.]

dwcpa: Assuming $B(z)$ constant, given photosynthesis parameters, location and date, calculate the daily, water-column production $P_{Z,T}$ according to (non-spectral) analytic solutions derived by Platt *et al.* (1990). Also calculate the daily production for any arbitrary interval of depth.

dwcpn: Given location and date, parameters for the biomass-depth profile, and parameters for the photosynthesis-irradiance curve, calculate the total daily, water-column production by numerical integration of the spectral equations (Sathyendranath & Platt 1989, with modifications as described in Sathyendranath *et al.* 1995). In the current release, spectral dependences are specified using tabulations. In a future release the software will allow specification of spectral dependences by means of approximating polynomials as described in Platt *et al.* (1994) and will support calculation of the production in an arbitrary layer.

dwcpe: Given the necessary information, calculate the daily, water-column production by any of the empirical approximators of same using their canonical equivalents as derived by Platt & Sathyendranath (1993).

ocb: Given any profile of biomass, estimate what the ocean-colour satellite would deduce the biomass to be. [This program is not included in the current distribution.]

With the exception of gfit, all the programs support both interactive and batch operation.

Chapter 2

Input Data

2.1 Introduction

This chapter discusses the input data used to compute basin-scale estimates for daily primary production using remotely-sensed data. Examples of data sets suitable for use in generating the input parameters required for basin-scale calculations are provided, as are examples of parameter-estimation procedures. In general, data from both shipboard sampling and from remote sensing are required. The software is, however, designed for general application and does not require the particular data sets mentioned below.

Estimates of daily water-column primary production using the approach of Platt & Sathyendranath (1988, with modifications as described in Sathyendranath *et al.* 1995) require four classes of information:

1. cloud coverage (not discussed here),
2. surface chlorophyll,
3. parameters describing the water-column distribution of biomass (depth profile), and
4. parameters describing the photosynthetic response of phytoplankton to light.

Typically, these data would be organized so that the appropriate values can be retrieved as functions of location and date (*e.g.*, latitude, longitude, and day number).

The spatial and temporal resolution required of the data will depend on the task. For example, total annual production might be estimated by computing production for one day in each of the twelve months on a one-degree lattice covering the globe. Another calculation might focus on a much more restricted domain, but compute estimates at more closely spaced dates and locations.

For the purpose of computing an annual estimate, production values for the 15-th day of each month can be generated using CZCS-, and cloud-coverage-, data sets provided as monthly means. The latitude and day number dictate the zenith angle of the sun (Paltridge & Platt 1976), which in turn determines surface irradiance (Bird 1984).

Depth profiles for photosynthetically active biomass, $B(z)$, and for the biomass (including phaeopigments) which contributes to light attenuation, $C(z)$, are required (for the formal distinction between B and C, see Sathyendranath *et al.* (1989)). These may be obtained from direct, shipboard observation or from a model. In the open ocean, the contribution of phaeopigments to $C(z)$ is negligible (Longhurst *et al.* 1995). Accordingly, we may use $B(z)$ to represent both quantities, and compute the diffuse, vertical, attenuation coefficient from $B(z)$ as described by Sathyendranath & Platt (1988).

Using the attenuation coefficient and surface irradiance (which may require correction for cloud cover), irradiance available at each depth throughout the photic zone can be determined.

With irradiance at each depth computed as described above, only parameters describing the photosynthetic response of chlorophyll biomass to light for a region and season are needed to estimate primary production at each depth.

2.2 Surface pigment field

Historically, NASA, Goddard Space Flight Center (courtesy Wayne Esaias, Chuck McClain, Gene Feldman, and Norman Kuring), has provided files of the CZCS monthly and seasonal climatologies for surface pigments for the years 1979–1986, gridded on a one-degree lattice. These are supplied as means, weighted means, and standard deviations of the means stored in the matching Level 3 (processed data) files.

The spectral reflectance model of Sathyendranath & Platt (1989, with modifications as described in Sathyendranath *et al.* 1995) may be used to convert the reflectance ratio obtained from the CZCS data to satellite-weighted biomass. This conversion requires information on the vertical distribution of biomass.

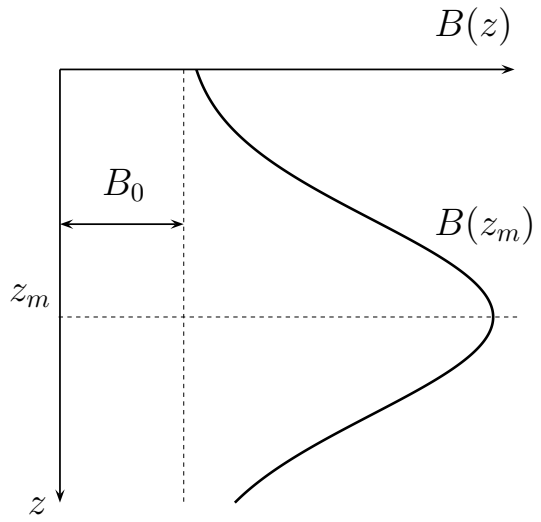

Figure 2.1: Shifted-Gaussian model for biomass depth profiles.

2.3 Relation between biomass and depth

A Gaussian profile was used by Lewis *et al.* (1983) to describe the vertical distribution of chlorophyll biomass in the ocean. Later, Platt *et al.* (1988) introduced a shifted-Gaussian, that is a Gaussian superimposed on a constant background, to provide a more versatile profile (Figure 1 of Platt *et al.* (1994)). The shifted-Gaussian model for biomass depth profiles may be described (using just one of many available, mathematically equivalent, formulations) by the equation:

$$\begin{aligned} B(z) &= f(z; h, \sigma, B_0, z_m) \\ &= B_0 + \frac{h}{\sigma\sqrt{2\pi}}\, e^{-\dfrac{(z-z_m)^2}{2\sigma^2}}, \end{aligned} \quad (2.1)$$

where $B(z)$ is chlorophyll biomass as a function of depth, z (positive downwards). B_0 represents the "background" concentration on which a Gaussian curve, whose peak is centered at depth z_m, is superimposed (Figure 2.1). In some cases, a degenerate form of the model (with $B_0 \equiv 0$ or $h \equiv 0$) may be appropriate.

The shifted-Gaussian model allows one to match the surface values without compromising the structure of the rest of the profile. This is the standardised profile that has been adopted by subsequent authors (Morel & Berthon 1989, Sathyendranath & Platt 1989, Sathyendranath *et al.* 1989, André 1992, Gordon 1992). It has been found that a shifted-Gaussian provides a suitable description of the vertical distribution of chlorophyll for both $B(z)$ and $C(z)$ (often, these are the same) at a wide range of locations in the ocean.

2.4 Relationship between photosynthesis and light

The production models provided here use two parameters to describe the relation between photosynthesis and light: the initial slope, α^B (mg C (mg Chl)$^{-1}$ h^{-1} (Wm^{-2})$^{-1}$), of the relationship between photosynthesis and irradiance, and the assimilation number, P_m^B (mg C (mg Chl)$^{-1}$ h^{-1}). Note that both parameters are restricted to positive values.

Various models that have been proposed for the relationship between photosynthesis and irradiance could be employed to obtain the required estimates. Here, as an example, the following model (Platt *et al.* 1980) is used:

$$P/B = P^B = P^B_m(1 - \exp(-\alpha^B I/P^B_m)),$$

where P is the instantaneous rate of primary production per unit volume of water, B is the photosynthetically-active biomass, I the irradiance, and the parameters are α^B and P^B_m.

2.5 Parameter estimation

This section provides advice and examples for estimating parameters required in the daily, water-column production calculations.

Parameter estimation for nonlinear models is often more art than science. For a given model and program there is almost always a data set that will cause the program to fail. Many suitable programs exist for fitting the models. Some of these support a specific set of model equations, including one or more parameterizations of the shifted-Gaussian model. Programs which support only specific models are able to use exact expressions to evaluate partial derivatives required by the iterative fitting procedure. Although such direct calculation of derivatives is considerably more efficient than the numerical approximations used by more general programs, the latter provide greater flexibility to explore variations, such as different parameterizations, in the model equations.

Given a suitable data set, the following general steps are required to produce an estimate for the parameters of a model:

1. define the model equations,
2. plot the data,
3. determine the initial parameter estimates,
4. run the numerical procedure, and
5. evaluate the resulting estimates.

In practice, this is seldom a linear process. Several iterations of certain steps may be required to obtain useful results. Thus, success requires some understanding of the process and the software.

2.5.1 Sources of difficulty

Difficulties in obtaining parameter estimates using nonlinear curve fitting software can often be traced to one of the following sources:

- data of marginal quality (too few observations or excessive variability);
- errors in entering the model equations; or
- poor initial parameter estimates.

A composite plot of the data and the model (using the initial parameter estimates) will often help identify a problem before the numerical parameter-estimation routine is invoked. The gfit utility provided here facilitates this interactive approach. As a rule of thumb, three data points are required for each significant feature in a model. Thus, in the case of biomass profiles, there should be at least three observations defining the location of the chlorophyll maximum. In practice, there are data sets in which all but one of the data points lie close to the background level. For such data sets, similar quality fits may be obtained for which the single point sampled in the maximum layer lies on the upper or the lower limb of the Gaussian peak. Such data sets should not be used for fitting the shifted-Gaussian model.

When comparing the numerical values of parameter estimates obtained using different programs (including different implementations of an algorithm), it is important to recognize conditions which may lead to divergent numerical estimates. Differences that are small in relation to the confidence intervals of the estimates are seldom meaningful. Some data sets admit several different parameterizations. In the case of the shifted-Gaussian model this problem is evident in cases where the depth interval of the samples is large compared with the width of the

peak, so that all but one observation have concentrations close to the background level. In such cases it is possible to choose the initial parameter values so that the estimated location for the maximum is either above, at, or below the depth of the highest concentration. Composite plots of the data and the model for different parameter estimates can help determine whether differences in the estimates are meaningful.

2.5.2 Parameterization of models

This section provides background for the specific function definitions used for the examples below. These examples serve to illustrate the application of the parameter-estimation program to obtain input parameters for the production calculations. Readers who are not mathematically inclined may wish to skip over the remainder of Chapter 2, which is not required to enable use of the gfit program in straightforward cases. The remainder of this chapter will, however, be useful to readers wishing to adapt other programs for use in generating parameter estimates for the production calculations.

It is often necessary to modify the parameterization of the model equation. For example, negative values for some parameters of a model (*e.g.*, parameters representing a concentration) may have no physical meaning. Many parameter-estimation routines assume that parameters can take on both positive and negative values, and may attempt to calculate a value of the model equations using parameter values for which the equation is not defined. In some cases, (*e.g.*, when the model equation uses a function such as $\log(x)$ that is undefined for $x < 0$), a numerical error will occur. Although it is possible to design programs in such a way that numerical errors are "trapped" and suitable actions taken, there are also cases in which it is possible to compute a numerical value for a model using a value for parameter that has no physical interpretation (*e.g.*, negative concentration). To avoid this, it is customary to introduce a new parameterization of the model. For example, consider the model function $f(x; p, q)$ with parameters p and q. Suppose the parameter p can assume only positive values. Define $p(a) = \exp(a)$ so that any value for the new parameter a produces a positive value for p and replace the function f by $g(x; a, q) = f(x; \exp(a), q)$ with parameters a and q.

2.5.3 Shifted-Gaussian model for biomass profiles

The non-degenerate case corresponds to the constraints $h > 0$, $\sigma > 0$, and $B_0 > 0$. The following change of variables:

$$\begin{aligned} a &= \log(h), \\ b &= \log(\sigma), \text{and} \\ c &= \log(B_0) \end{aligned}$$

will avoid potential problems using parameter-estimation software that does not support parameter constraints. Using the above parameterization, (2.1) may be written in the following form:

$$\begin{aligned} B(z) &= f(z; e^a, e^b, e^c, z_m) \\ &= g(z; a, b, c, z_m) \\ &= e^c + \frac{1}{\sqrt{2\pi}} e^{a - b - e^{-2b} \frac{(z - z_m)^2}{2}}. \end{aligned}$$

2.5.4 Relationship between photosynthesis and light

As above, in view of the restrictions $\alpha^B > 0$ and $P_m^B > 0$, a change of variables is recommended to ensure that the nonlinear parameter-estimation program does not attempt to evaluate the function using a (meaningless) negative value for one of the parameters. The following change of variables:

$$\begin{aligned} p &= \log(P_m^B) \quad \text{and} \\ q &= \log(\alpha^B) \end{aligned}$$

will be used in the example provided below.

Chapter 3

Algorithms

3.1 Introduction

This section provides a brief description of each program in the ocean production package. Data requirements and output variables are presented for each of the major calculations. The individual program appendices provide theoretical notes and algorithmic details for each of the programs. The references provided in these appendices should be consulted for the scientific basis of the algorithms.

3.1.1 Consistency of input data

With the exception of gfit, the interactive nonlinear parameter-estimation utility, calculations are performed using batch-oriented programs. Support for interactive use is provided using a single, separate, general purpose data-entry utility which simply creates small data files to be processed using the batch-oriented programs.

gfit provides a single set of parameter values from a flat input file consisting of paired values of the dependent and independent variables. A typical data file may range in size from a few hundred to a few thousand bytes. For the remaining programs, a single input record (generally less than 100 bytes) will produce a single output record, also on the order of 100 bytes. Depending on the calculation, a data file may have from one to many thousand records.

The programs are not designed to detect incorrect or inconsistent input data. The batch-oriented programs do not perform any checks on the input data. The interactive data-entry utility truncates a value that is outside a range that is specified for each data item. Since it is often difficult to determine a realistic range for a particular variable, the specified ranges are not restrictive. There is no check for consistency between variables. Such tests would make the programs larger and slower, thus limiting the utility of the programs for users with inexpensive computer hardware and increasing the cost of calculations involving large data sets.

3.1.2 Note on geographical coordinates

Although many of the programs require values for latitude and longitude as input, longitude values are not used for any of the calculations. Provision for longitude is included as a convenience because many data sets are indexed by geographical location.

3.2 Parameter estimation

A simple, general-purpose nonlinear parameter-estimation utility, gnuplot, is provided to assist in preparing suitable input files for the analysis programs. The parameter-estimation module in gnuplot uses the Levenberg-Marquardt nonlinear least-squares algorithm with finite-difference approximation of the partial derivatives. The MS-DOS[1] implementation of gnuplot is a compromise to support low-cost computer systems.

A simple script, gfit, is used to invoke the gnuplot program for a specific model and data file. Available models include the shifted-Gaussian model for biomass-depth profiles and

[1] MS-DOS is a trademark of Microsoft Corporation.

a model for the relationship between photosynthesis and irradiance. Both applications are standard problems in nonlinear parameter-estimation. They could be performed more effectively (given suitable hardware) using many other commercial or free packages.

3.2.1 Biomass profiles

Given a series of chlorophyll biomass observations $B(z)$, extract the profile parameters for the shifted-Gaussian model. This model admits three cases, characterized by the number of parameters:

Four parameters:

The most general form of the model describes a constant (background) plus a Gaussian, and is modelled by the following equation:

$$B(z) = B_0 + \frac{h}{\sigma\sqrt{2\pi}}\, e^{-\dfrac{(z-z_m)^2}{2\sigma^2}};$$

Three parameters:

The first simplification is to eliminate the background component, leaving only the Gaussian. The model equation is:

$$B(z) = \frac{h}{\sigma\sqrt{2\pi}}\, e^{-\dfrac{(z-z_m)^2}{2\sigma^2}}; \text{and}$$

One parameter:

The simplest model assumes that $B(z)$ is constant, and is modelled by the following equation:

$$B(z) = B_0.$$

Table 3.1 lists the input variables and resulting parameters for the shifted-Gaussian model.

3.2.2 Photosynthesis and irradiance

Given a series of observations for the rate of photosynthesis, normalized to biomass, for varying light intensities, obtain estimates for the parameters α^B and P_m^B for the relationship between photosynthesis and irradiance (Table 3.2). Superscript B indicates normalisation to biomass.

Variables	
Input	Output
depths, $z_1, ..., z_n$	depth, z_m, of chlorophyll maximum
biomass, $B(z_1), ..., B(z_n)$	Gaussian height parameter, h
	Gaussian scale parameter, σ

Table 3.1: Input and output variables for the program gfit when used to estimate parameters for the shifted-Gaussian biomass profile model.

Variables	
Input	Output
irradiance values, $I_1, ..., I_n$	initial slope, α^B, for the relationship of photosynthesis to irradiance
specific production, $P^B(I_1), ..., P^B(I_n)$	assimilation number, P_m^B

Table 3.2: Input and output variables for the program gfit when used to estimate parameters for the relationship between photosynthesis and irradiance.

3.3 Depth-averaged concentration

For given biomass profile parameters and the upper and lower boundaries of an arbitrary depth interval, calculate the average biomass concentration, $\langle B \rangle$, for that interval. Table 3.3 lists the input and output variables for the program.

Variables	
Input	Output
depth, z_m, of chlorophyll maximum	$\langle B \rangle$
background component, B_0	
Gaussian height parameter, h	
Gaussian scale parameter, σ	
depth to top of layer, l	
depth to bottom of layer, u	

Table 3.3: Input and output variables for the program avgc.

Variables	
Input	Output
latitude	day length, D
longitude	total daily irradiance, I_T
day number	noon irradiance, I_0^m

Table 3.4: Input and output variables for the program i0.

3.4 Surface irradiance

For a given location and date, and assuming clear sky, compute daylength, D; noon irradiance, I_0^m; and total daily irradiance, I_T. Table 3.4 lists the input and output variables for the program.

3.5 Daily, water-column production

Estimates of total, daily, water-column production may be computed using various models, some of which admit analytic solution. Three programs, representing a range of models, are included in the package: a) dwcpa using the analytic solution for the case of a constant biomass profile, b) dwcpn using direct numerical integration of the full spectral model for shifted-Gaussian depth profiles, and c) dvcpe using various empirical estimators of historical interest. Input and output variables for each of these programs are shown in Table 3.5. Note that, although the programs perform similar calculations, the input data requirements are quite different.

3.5.1 Constant $B(z)$

For given photosynthesis parameters, location and date, calculate the daily, water-column production $P_{Z,T}$ according to (non-spectral) analytic solutions derived by Platt *et al.* (1990). Also calculate the daily production for any arbitrary interval of depth.

3.5.2 Gaussian profile

Given location and date, profile parameters, and photosynthesis parameters, calculate total daily, water-column production by numerical integration of the spectral equations.

3.5.3 Empirical approximations

Given the necessary information, calculate the daily, water-column production by any of the empirical approximators of same using their canonical equivalents, $f(I_*^m)$, as derived by Platt & Sathyendranath (1993).

Method	Variables	
	Input	Output
Analytic		
dwcpa	latitude	latitude
	longitude	longitude
	day number	day number
	initial slope, α^B, for relationship of photosynthesis to irradiance	day length, D
	assimilation number, P_m^B	dimensionless noon irradiance, I_*^m
	biomass, B	daily production for a layer, $P_{Z_1,Z_2,T}$
	attenuation coefficient, K	
	depth to top of layer, Z_1	
	depth to bottom of layer, Z_2	
Numerical		
dwcpn	latitude	latitude
	longitude	longitude
	day number	day number
	initial slope, α^B, for the relationship of photosynthesis to irradiance	day length, D
		total daily irradiance, I_T
	assimilation number, P_m^B	daily water-column production, $P_{Z,T}$
	depth of phytoplankton maximum, z_m	
	background biomass, B_0	
	Gaussian height parameter, h	
	Gaussian scale parameter, σ	
Empirical		
dwcpe	initial slope, α^B, for the relationship of photosynthesis to irradiance	values (by various methods) for the canonical function, $f(I_*^m)$, as defined in Platt & Sathyendranath (1993)
	assimilation number, P_m^B	
	peak (noon) surface irradiance, I_0^m	

Table 3.5: Input and output variables for daily water-column production programs.

Chapter 4

Implementation Notes

4.1 Introduction

This chapter provides implementation details for the programs included in the package. For information on using a specific program, consult the appropriate Appendix. Many readers can safely skim over the more technical portions of section 4.2. For users who are not familiar with Fortran, a brief description of Fortran format strings is provided in section 4.3.1.

4.2 Programs

The programs in this package can be used under current versions of either MS-DOS[1] or OS/2[2] As described below, a wide range of MS-DOS configurations is supported, with the expectation that very few users will need to modify their existing system configurations in order to use the programs in this package. With the exception of gnuplot, the actual numerical calculations are performed by batch-oriented programs. Each of these programs reads input data from a single file and writes results to an output file. A single output record is created for each input record. In most cases, a status field in the output record is used to indicate cases where the calculations could not be performed (usually due to some problem with the input data). The programs perform a minimum of checks for consistency of the input data, so it is necessary to exercise care in preparing input data files. A record of the processing is written to a log file. This file should always be examined to ensure that the expected number of records has been processed and to determine whether any processing anomalies occured.

To facilitate interactive experimentation, a simple interactive data entry utility is provided. Command scripts, provided as MS-DOS `.bat` files and as OS/2 `.cmd` files, allow a user to edit data, run an analysis, and view results interactively. These scripts also serve to hide many of the configuration details required to run the numerical programs.

In creating a software package for general use, it is necessary to make some assumptions about the hardware which will be used to run the programs. In the scientific community, personal computers running the MS-DOS operating system and based on the Intel 80386 processor with a numeric coprocessor, at least 2M bytes of memory, and a hard disk are inexpensive and widely available. Such systems are, in principle, quite adequate for all but the most ambitious applications of the programs in this package. In practice, however, effective numerical calculation requires that the CPU execute 32-bit instructions in protected mode. Since 32-bit mode is outside the realm of MS-DOS, special software is required to exploit the full capability of the 80386 processor. Many strategies have been developed to permit the use of 32-bit code under MS-DOS, and a number of products, including recent versions of MS-DOS, support one or more of these mechanisms. Such extensions are, however, difficult to configure in a way that is reliable and that avoids conflicts with the other software used on the system.

[1] MS-DOS is a trademark of Microsoft Corporation.

[2] OS/2 is a trademark of International Business Machines, Incorporated.

Given the complexity of the mechanisms supporting 32-bit applications under MS-DOS, it was desirable that as many of the programs as possible be suitable for use on this class of hardware without requiring changes to existing configurations. Device drivers and network utilities often consume a significant portion of low DOS memory (the first 640k bytes), while extended memory on such MS-DOS systems is generally available to application programs. Thus, applications that can use 32-bit memory may require fewer changes to existing system configurations than those compiled for the 16-bit, Intel 8086 segmented architecture. In addition, 32-bit, Intel 80386 level code is potentially faster and makes efficient use of memory outside the 640k low DOS memory region. As a result, such programs can be used on a wide range of existing systems without the need to modify an existing memory and device configuration.

Further advantages of the 32-bit, flat, 80386 memory model over 16-bit code include the ability to create significantly larger programs without changing the compiler and access to higher quality libraries. Historically, run-time libraries used for 16-bit compilers have required compromises to satisfy the limitations imposed by the segmented memory model and the need to conserve memory. By contrast, the run-time libraries developed for 32-bit environments have been optimized for performance and reliability under greatly relaxed constraints on memory. Even simple programs often benefit from the more sophisticated runtime library afforded by a 32-bit compilation environment. These benefits are most visible for programs which perform significant numerical calculations.

Since the introduction of the Intel 80386, a variety of third-party extensions have appeared to support 32-bit applications under MS-DOS. In general, two components are used: a *memory manager* such as emm386, which is loaded from `config.sys` (*i.e.*, considered part of the system configuration), and a DOS *extender*, which is usually provided with each application program (often the code for the DOS extender is embedded inside each executable file). In order to support a variety of DOS memory managers, two different, but compatible, DOS extenders are included in this package. To permit a selection of the appropriate DOS extender at run-time, they are provided in the form of separate executables (rsx.exe and emx.exe). In most cases, the application programs will run the appropriate DOS extender without intervention by the user. This flexibility reduces the number of instances in which users would be required to change their system configuration in order to use the programs. It should be noted that the application programs have been compiled specifically for use with the two DOS extenders included with the package, and should not be expected to work with DOS extenders intended for use with other compilers.

4.2.1 Language and compiler

The numerical programs developed by the authors are maintained using the FWEB literate programming system. This system allows program source and documentation to reside in a single file from which a Fortran source program or documentation in TEX format may be extracted. The data-entry utility is a small C-language program created specifically for this package. The remaining binary executables were either obtained as binary files or were compiled from freely available C-language source distributions.

With the exception of the DOS version of the nonlinear parameter-estimation utility, gnuplot, the numerical programs are compiled using the Intel 80386, flat, 32-bit memory model to run under OS/2 and (extended) DOS. The DOS version of gnuplot was compiled as a 16-bit DOS application using Borland C++ version 3.0 to support HGC, CGA, EGA and VGA graphics adapters; the OS/2 version is a 32-bit Presentation Manager[3] application.

The choice of a compiler was dictated by the requirement to support a wide range of DOS memory configurations and the quality of the runtime library. These considerations favor the EMX port of the Free Software Foundation's g77 Fortran front end and gcc compiler (Appendix

[3]Presentation Manager is a trademark of International Business Machines, Incorporated.

B). Programs created using this compiler require an 80386 (with 80387 NDP), 80486, or Pentium processor and will execute under MS-DOS (when a suitable DOS extender is supplied) or OS/2. To accommodate a wide range of existing system configurations, two different DOS extenders are included with the package. When run under MS-DOS, the programs will, in most cases, automatically invoke the appropriate DOS extender for the memory configuration in effect when the program is run. The programs can, as a result, be used without reconfiguration under DOS, OS/2, and Windows environments on a single system.

The Fortran runtime libraries are from AT&T's Fortran-to-C translator (Feldman *et al.* 1993), which is, in turn, based on the UNIX `f77` compiler. This library is freely available in source form and has been in widespread use for more than a decade. A useful discussion of possible causes for runtime errors which may be encountered using this library is provided by Loukides (1990, Appendix D). A runtime error while using a program created using `g77` will generally result in the creation of a *core* file (one exception is when the error is the result of exhausting the available space on the disk). These files are sometimes useful in conjunction with the GNU debugger to trace the source of an error; most users will, however, want to delete the file called "`core`" whenever one is created.

4.3 Files

The batch-oriented programs each require a single input data file and produce two output files, a data file and a processing log. These files are identified by a three letter *extension*. Input data files use the `.dat` extension, output data files use `.out`, and log files use `.log`.

4.3.1 Data file formats

The programs use standard Fortran record formats. Input and output data files are ASCII text and consist of a two line header record followed by one or more data records. The header consists of one line of text followed by a line containing the Fortran format string required to read the data record. Historically, Fortran formats were designed for use with punched cards. Thus data fields are rigorously aligned in columns. In constructing data records for use with Fortran programs it is important to maintain column alignment by inserting spaces as required. Character variables must be aligned to the left of the field by adding trailing spaces, while numeric values must be aligned on the decimal point (*i.e.*, to the right for integers) by adding leading spaces.

The first line of the header may contain arbitrary text of up to 128 characters, and may be used to list the variables or otherwise identify the data set. The output files (`.out` extension) created by the programs described here use the first line of the header to list the variables.

The second line of the header provides the data record format. This is a Fortran format string suitable for reading the data records in the following lines of the file. The record format is a list of format elements, separated by commas, and bracketed by parentheses. Only the restricted set of Fortran format elements shown in Table 4.1 are permitted in the data record format. There is one element for each field. Thus, a data file consisting of an eight-character string followed by an integer value and two floating point values might appear as follows (spaces in the data records are shown as "␣" for clarity):

```
string␣␣␣int␣␣fixed␣␣also␣fixed
(a8,x,i3,2x,f6.2,x,f9.3)
abcdefgh␣123␣␣-12.12␣12345.123
example␣␣␣12␣␣␣␣1.12␣-1234.123
```

It is good practice to place one or more spaces between each data item. This produces a more readable file and makes it easier to manipulate the file using other software.

4.4 Performance

The algorithms chosen for use in the ocean production calculations provide a range of tradeoffs

Data type	Template	Examples	
		Format	Formatted value
string	aw	a4	abcd
		a8	abcde␣␣␣
integer	iw	i4	1234
		i3	␣-1
real	f$w.p$	f6.2	-12.12
		f7.3	␣12.123
space	nx	2x	␣␣

Table 4.1: Fortran data format templates and examples. Each template consists of a single letter (a, i, or f) which serves to identify the data type or, in the case of a blank, x. Data fields have a width, w, which gives the total number of columns. Non-integral fields also have a precision, p, which gives the number of columns after the decimal point. Multiple blanks are indicated by a count, n, placed before the "x".

between numerical accuracy and computational resources. A program which demands fewer resources permits more extensive testing, makes feasible the use of lower cost hardware, and facilitates exploration of new approaches, Monte Carlo simulation studies, and more extensive testing and profiling than might otherwise be feasible. It is not unusual to see a variation of several orders in magnitude in execution time among different algorithms for the same problem. In practice, different algorithms may have different data requirements and different patterns of memory access while executing as well as different instruction counts. The range of capabilities represented by the programs provided in this package should be sufficiently broad to meet the majority of practical requirements.

Table 4.2 compares execution time for the analytic and numerical calculations using various processors.

It is instructive to consider a simple estimate for the overall computational effort required for a global calculation of primary production using the complete, numerical integration. Assuming the calculation will be performed for a specific day using a one-degree grid, and omitting the third of the cells corresponding to land, $43200 = 2/3 \times 180 \times 360$ individual production estimates are required.

The innermost loop in the Fortran code used to estimate daily primary production by direct numerical integration contains fewer than 100 floating point operations, most of which are addition and multiplication. A much smaller operation count would be obtained by examining the theoretical form of the integrand, so this number is an overestimate. On average, a global calculation for a specific location and day executes this inner loop less than 3×10^4 times (60 wavebands $\times$ 40 depths $\times$ 12 time intervals). Desktop systems based on microprocessors capable of executing a floating point instruction in less than 10^{-6}s are widely available. For such a processor, a conservative estimate for the total processing time required to execute the floating point instructions is three seconds for each daily water-column production estimate, or 130,000 s (36 hours) for the global calculation. The fastest processors are more than a thousand times quicker than the benchmark processor used in this estimate. Furthermore, the global calculation is easily subdivided (*e.g.*, by geographical region) into independent calculations, so the time required to complete a global calculation can easily be reduced by using a number of processors, each working on different geographical region, at the same time.

The bulk of input data required for such a calculation consists of a record containing approximately 10 values for each one degree cell,

Solution method	Execution time (s) for various processors	
	80386sx + 80387 (25mhz)	80486 (66mhz)
Analytic (dwcpa)	0.15	0.02
Numerical (dwcpn)	58.5	7.6

Table 4.2: Performance of different algorithms for the calculation of daily water-column production using different processors. The time (s) required to compute one daily production estimate is shown for two different processors. The 80486 processor has an internal floating point unit.

so that the program must read on the order of 10×43200 values and write approximately 43200 output values. Assuming 10 bytes per value yields a total of roughly five million bytes of I/O. A modern workstation can read or write data at more than a million bytes per second, whereas a personal computer is typically two orders of magnitude slower. In either case, the time required to read and write data is expected to require a small fraction of the total execution time (in some multitasking environments, however, the granularity of these operations becomes a significant factor – the current implementation interleaves I/O with calculation and may adversely impact performance).

Bibliography

André, J.-M. (1992), 'Ocean color remote-sensing and the subsurface vertical structure of phytoplankton pigments', *Deep-Sea Res. I* **39**, 763–779.

Bird, R. E. (1984), 'A simple, solar spectral model for direct-normal and diffuse horizontal irradiance', *Solar Energy* **32**, 461–471.

Feldman, S.I., D.M. Gay, M.W. Maimone, and N.L. Schryer (1993), A Fortran-to-C Converter, Computing Science Technical Report 149, AT&T Bell Laboratories. Issued May, 1990, updated March, 1993.

Gordon, H. R. (1992), 'Diffuse reflectance of the ocean: Influence of non-uniform phytoplankton pigment profile', *Appl. Optics* **31**, 2116–2129.

Lewis, M. R., J. J. Cullen, and T. Platt (1983), 'Phytoplankton and thermal structure in the upper ocean: Consequence of nonuniformity in chlorophyll profile', *J. Geophys. Res.* **88**, 2565–2570.

Longhurst, A., S. Sathyendranath, T. Platt, and C. Caverhill (1995), 'An estimate of global primary production in the ocean from satellite radiometer data', *J. Plankton Res.* **17**, 1245–1271.

Loukides, M. (1990), *UNIX for FORTRAN Programmers*, O'Reilly & Associates, Inc., Sebastopol, CA.

Morel, A., and J.-F. Berthon (1989), 'Surface pigments, algal biomass profiles, and potential production of the euphotic layer: Relationships reinvestigated in view of remote-sensing applications', *Limnol. Oceanogr.* **34**, 1545–1562.

Paltridge, G. W., and C. M. R. Platt (1976), *Radiative processes in meteorology and climatology*, Elsevier Scientific Publishing Company, Amsterdam.

Platt, T., and S. Sathyendranath (1988), 'Oceanic primary production: Estimation by remote sensing at local and regional scales', *Science* **241**, 1613–1620.

Platt, T., and S. Sathyendranath (1993), 'Estimators of primary production for interpretation of remotely sensed data on ocean color', *J. Geophys. Res.* **98**, 14,561–14,576.

Platt, T., C. L. Gallegos, and W. G. Harrison (1980), 'Photoinhibition of photosynthesis in natural assemblages of marine phytoplankton', *J. Mar. Res.* **38**, 687–701.

Platt, T., S. Sathyendranath, C. M. Caverhill, and M. R. Lewis (1988), 'Ocean primary production and available light: Further algorithms for remote sensing', *Deep-Sea Res. I* **35**, 855–879.

Platt, T., S. Sathyendranath, and P. Ravindran (1990), 'Primary production by phytoplankton: analytic solutions for daily rates per unit area of water surface', *Proc. R. Soc. Lond. Ser. B* **241**, 101–111.

Platt, T., S. Sathyendranath, G. N. White III, and P. Ravindran (1994), 'Attenuation of visible light by phytoplankton in a vertically-structured ocean: Solutions and applications', *J. Plankton Res.* **16**, 1461–1487.

Sathyendranath, S., and T. Platt (1988), 'The spectral irradiance field at the surface and in the interior of the ocean: A model for applications in oceanography and remote sensing', *J. Geophys. Res.* **93**, 9270–9280.

Sathyendranath, S., and T. Platt (1989), 'Remote sensing of ocean chlorophyll: Consequence of non-uniform pigment profile', *Appl. Optics* **28**, 490–495.

Sathyendranath, S., T. Platt, C. M. Caverhill, R. E. Warnock, and M. R. Lewis (1989), 'Remote sensing of oceanic primary production: Computations using a spectral model', *Deep-Sea Res. I* **36**, 431–453.

Sathyendranath, S., A. Longhurst, C. M. Caverhill, and T. Platt (1995), 'Regionally and seasonally differentiated primary production in the North Atlantic', *Deep-Sea Res. I* **42**, 1773–1802.

Appendix A

Notation

Notation for Ocean Production Models

Symbol	Description
α	initial slope of P-I curve, defined as $\partial P/\partial I\|_{I\to 0}$, mg C h^{-1} (W $m^{-2})^{-1}$.
α^B	initial slope of P^B-I curve, defined as $\partial P^B/\partial I\|_{I\to 0}$, mg C (mg Chl)$^{-1}$ h^{-1} (W $m^{-2})^{-1}$.
A	scale factor, $A \equiv (BDP_m^B)/K$, in canonical form of the solution for daily primary production, mg C m^{-2}.
B	biomass, as concentration of chlorophyll a, mg Chl m^{-3}.
$B(z)$	depth-dependent, photosynthetically active biomass, as concentration of chlorophyll a, mg Chl m^{-3}.
B_0	parameter in shifted-Gaussian model (2.1), representing the background component of depth-dependent chlorophyll biomass, mg Chl m^{-3}.
$\langle\, B \,\rangle$	average concentration of chlorophyll biomass over a depth layer, $Z_1 < z < Z_2$, mg Chl m^{-3}.
$C(z)$	depth-dependent biomass, including phaeopigments, as concentration of chlorophyll a, mg Chl m^{-3}.
D	day length, hours.
$f(I_*^m)$	function arising in dimensional analysis and in canonical form, of solution for daily primary production, dimensionless.
h	height (amplitude) parameter in the shifted-Gaussian model (2.1), mg Chl m^{-2}.
I	irradiance in the photosynthetically active range (PAR), W m^{-2}.
I_*	dimensionless surface irradiance, $I_*(t) \equiv I_0(t)/I_k$.

continued on next page

NOTATION *continued from preceeding page*

SYMBOL	DESCRIPTION
I_*^m	dimensionless irradiance at local noon, $I_*^m \equiv I_0^m / I_k$.
$I_0(t)$	time-dependent surface irradiance, where t is the time of day, W m^{-2}
I_0^m	maximum surface irradiance at local noon, W m^{-2}.
I_k	adaptation parameter of the P-I curve, $I_k \equiv P_m^B / \alpha^B \equiv P_m / \alpha$, W m^{-2}.
I_T	total daily irradiance at the surface, W m^{-2} d^{-1}.
K	vertical attenuation coefficient for irradiance, $K = -I^{-1}(dI/dz)$, m^{-1}.
$P(I)$	generic functional form of the photosynthesis–light curve, mg C m^{-3} h^{-1}.
P	primary production rate per unit volume, mg C m^{-3} h^{-1}.
P^B	primary production rate normalized to biomass, $P^B \equiv P/B$, mg C (mg Chl)$^{-1}$ h^{-1}.
P_m^B	assimilation number, specific production at saturating light, in the absence of photoinhibition, $P_m^B = P^B\|_{I \to \infty}$, mg C (mg Chl)$^{-1}$ h^{-1}.
P_m	primary production (maximum) at saturating light, $P_m \equiv B\ P_m^B$, mg C m^{-3} h^{-1}.
$P_{Z,T}$	daily primary production for the water column, mg C m^{-2} d^{-1}.
$P_{Z_m,T}$	daily primary production for the mixed layer, mg C m^{-2} d^{-1}.
$P_{Z_1,Z_2,T}$	daily primary production for a layer between Z_1 and Z_2, mg C m^{-2} d^{-1}.
σ	scale parameter for the width of the peak in the shifted-Gaussian model (2.1), m.
t	time, hours.
z	depth, m.
Z_1	an arbitrary depth, m.
Z_2	an arbitrary depth, $Z_2 > Z_1$, m.
z_m	location parameter in the shifted-Gaussian model (2.1), representing the depth of the chlorophyll maximum, m.

Appendix B

Note on the Compiler and Utilities

The quality of the executable programs provided in this package is in no small part due to the quality of the compiler and libraries used to create them. In compiling this Appendix, we have attempted to list the names of people who have assumed leading roles in the development and enhancement of the compiler, libraries, and other tools used to prepare the programs, as well as the utilities included with the programs. Such a list must necessarily be incomplete. The Free Software Foundation, by promoting continued development of the GNU software tools, has played a crucial role in making the programs listed below accessible and useful.

Program and version	Contributors
EMX port of gcc, version 0.9a	Eberhard Mattes
rsx DOS extender for EMX	Rainer Schnitker
GNU g77, version 0.5.16	James Craig Burley
EMX ports of GNU software	Michael Holzapfel, and Kai Uwe Rommel
fweb, version 1.39	John Krommes
ftnchek, version 2.8	Robert Moniot
gnuplot, versions 3.5 and 3.6	Thomas Williams, Alex Woo, Russell Lang, David Kotz, Dick Crawford, David Denholm, and Carsten Grammes
GNU awk (gawk), version 2.15	David Trueman, Arnold Robbins, Scott Deifik, Kai Uwe Rommel, and Darrel Hankerson
termlib	George Sherouse