User's Manual

BLOOM II: a mathematical model to compute phytoplankton blooms

WABASIM Project R1310

December 1985

Release 2
GENERAL INTRODUCTION

This manual describes how to use the computer program of the phytoplankton model BLOOM II. This model is an extended and modified (growth water) version of the Algae Bloom Model developed by the Rand Corporation for the Oosterschelde sea estuary. The model computes the maximum potential biomass consistent with a number of environmental conditions. The concept of this model is explained by Los [1982]. Some major modifications of BLOOM II will be described in a second report [Los, in prep.].

BLOOM II is one of the results of a multidisciplinary project called WATER BASIn Models (WABASIM), which is financed by the Delta Department and carried out in cooperation between the Environmental Division of the Delta Department and the Water resources and Environment Division of the Delft Hydraulics Laboratory. Several members of the Rand Corporation, in particular J. H. Bigelow, have also contributed to this project. Other mathematical models developed for WABASIM concern water chemistry (CHARON) and bottom bi-chemistry (SEDMOD).

There exists, of course, a close relationship between the various models. In most applications an integrated version of BLOOM II and CHARON has been used. This model version, which has recently been named JSBACH (Joint Simulation of Biology And Chemistry) is not described in the current manual, but in the manual on CHARON [De Rooij, in prep.]. Recently a dynamic version of BLOOM II has been developed, which was also linked to the DELMAQ model. It is, however, too early yet to include this model version in the current manual.

Many coefficients of the models were determined from measurements collected in four separate bodies of water in the drinking water reservoir 'Grote Rug' by the National Institute for Drinking water supply and the Environmental Division. Several coefficient values have, however, been modified in recent years when the models were applied during various projects. Among these projects are

- A study on the impacts of management measures in Lake IJssel, R1552 [De Rooij et al., 1982].
- A study on the waterquality of the (future) Zoommeer and the fresh water lake Grevelingen, R1645 [De Rooij et al., 1982].
- Performance of the 1982/1983 versions of the models for the 'Grote Rug' reservoirs, 'WABASIM Cases' project, S594 [Smits et al., 1984].
- A study on the impacts of management measures in a Dutch polder area called Rijnland, R1651 [Smits et al., 1985].

The computer program of BLOOM II was mainly written by F. J. Los of the Delft Hydraulics Laboratory, based upon Rand's Algae Bloom computer program. Several other people have, however, contributed to the current computer program. In particular Norman Katz should be mentioned, who wrote a number of subroutines which increase the program's user friendliness significantly.

This manual consists of the following main sections:

1. An introduction to the model's concepts.
2. A description of the pre-processing programs.
3. A description of the input of BLOOM II, divided into
   a. Standard input.
   b. Optional input.
4. A (short) description of the computational section of the pro-
   gram.
5. A description of the output.
6. Several appendices with examples, subroutine calling sequence,
   description of subroutines etc.
SUMMARY OF CHANGES BETWEEN RELEASES 1 AND 2

The first release of the BLOOM II manual, which was published in 1984, appeared in a period when BLOOM II underwent some major revisions:

1. The fixed stochiometry of phytoplankton species was abandoned and replaced by a variable stochiometry concept.
2. The results of the incubator research project became the basis for the computation of the growth rates of the phytoplankton species.
3. The day length dependence of phytoplankton species was explicitly included.

The latest versions of BLOOM II therefore differ rather strongly from those used until 1983 in for example the Lake IJssel or WABASIM cases projects. So the theoretical sections of this manual had to be rewritten.

In the latest versions of the model phytoplankton types rather than species are the main variables of the model. In addition it proved necessary to enlarge the flexibility with which variables can be described. For example maximum growth rates can now be specified both as exponential as well as linear functions. Therefore changes in the in- and outputs of the model could not be avoided.

Obviously the number of changes in the model since 1983 is so large, that a new release of the manual had to be written. The most important differences between release 1 and release 2 of the BLOOM II manual are in 'Structure of the phytoplankton model BLOOM II' on page 1, 'The standard input of BLOOM II' on page 51, 'The outputs of BLOOM II' on page 81, and various appendices.
TERMINOLOGY AND NOTATIONAL CONVENTIONS

It is difficult to write about computers and computer programs without using a lot of technical expressions. In this manual we have, however, tried to minimize the amount of computer jargon and when it is used, we have included explanations in the text or in foot notes.

The first time a program or command name (instruction) is used in this manual it is (1) enclosed in single quotation marks, (2) capitalized and (3) highlighted (printed in bold characters). Thus for example the word 'HELP' in the text signifies the definition of this instruction. When we use an instruction which has been previously defined in the remainder of the text, it is no longer highlighted. It is distinguished from the rest of the text only by the quotation marks and the capitals as for example in 'HELP'.

Included in this manual are several examples of interactive usage of computer programs where an instruction typed by the user on his terminal is followed by a response message of the computer. To distinguish between the two we have highlighted the information entered by the user throughout this manual. Thus in the following sequence of lines

Parameter cmd:
  gr
Growth cmd:

it is immediately clear that the user has typed the instruction 'gr' and that the computer has prompted him with the message 'Growth cmd:'.

CONTENTS

1.0 Structure of the phytoplankton model BLOOM II ............. 1
1.1 Problem description ..................................... 1
1.2 Purpose of BLOOM II .................................... 1
1.3 Selection of phytoplankton species ....................... 3
1.4 Nutrient requirements of phytoplankton species .......... 4
   1.4.1 Nutrient mass-balances ............................ 4
   1.4.2 Nutrient recycling ................................ 6
   1.4.3 Formulation of nutrient constraints .............. 8
1.5 Energy requirements of phytoplankton species .......... 9
1.6 The energy budget of phytoplankton .................... 10
   1.6.1 Maximum growth and temperature ................... 11
   1.6.2 Respiration ...................................... 12
   1.6.3 Natural mortality .................................. 13
   1.6.4 Grazing by zooplankton ............................ 14
   1.6.5 Growth rates ..................................... 16
   1.6.6 Growth efficiency and light ....................... 17
   1.6.7 Averaging the production .......................... 20
   1.6.8 Steady state assumption ............................ 21
   1.6.9 Formulation of energy constraints ................. 24
1.7 Formulation of growth constraints ..................... 25
   1.7.1 Net rate of increase .............................. 26
1.8 Solution algorithm .................................... 27
1.9 Behavior and performance ................................ 28
1.10 Inputs to the model ................................... 29
   1.10.1 Lake specific inputs .............................. 29
   1.10.2 Universal inputs ................................ 29

2.0 BLOOM II and related computer programs ................. 33

3.0 The production efficiency programs .................... 34
3.1 Purpose ................................................ 34
3.2 The integration program BLEFFPRO ....................... 34
   3.2.1 Input of BLEFFPRO ............................... 34
   3.2.2 Output of BLEFFPRO ............................... 40
3.3 The combination program BLCOMBIN ....................... 40
   3.3.1 Input of BLCOMBIN ................................ 41
   3.3.2 Output of BLCOMBIN ............................... 42
3.4 The transformation program BLEFUNFO .................... 42
3.5 The transformation program BLEFFFORM ................... 42

4.0 BLOOM II computer program ............................. 43
4.1 General description of the BLOOM II program .......... 43
4.2 Interactive and batch processing ....................... 43
4.3 System dependent program features ..................... 45
   4.3.1 Special interactive functions .................... 45
   4.3.2 System commands .................................. 46
4.4 Job control for BLOOM II ................................ 46
   4.4.1 Standardized procedures ......................... 46
   4.4.2 File definitions for BLOOM II ................. 47
   4.4.3 Creating a terminal log file ...................... 49

5.0 The standard input of BLOOM II ......................... 51
5.1 Input file FT05F001 .................................. 51
5.2 Input file FT11F001 .................................. 59
5.3 Input file FT12F001 .................................. 62

6.0 The optional input of BLOOM II ......................... 63
6.1 Differences between batch and interactive mode ........................................ 63
6.2 How instructions should be entered ......................................................... 64
6.3 Control of model options ........................................................................... 66
6.4 Control of model coefficients ..................................................................... 67
   6.4.1 General coefficients ........................................................................... 68
   6.4.2 Phytoplankton species stoichiometric coefficients .............................. 69
   6.4.3 Phytoplankton species growth coefficients .......................................... 72
   6.4.4 Zooplankton coefficients ..................................................................... 74
7.0 The computations of BLOOM II ................................................................. 76
7.1 Extinction roots computations .................................................................... 76
7.2 Solution of the linear programs .................................................................. 78
8.0 The outputs of BLOOM II .......................................................................... 81
  8.1 The standard output of BLOOM II ............................................................. 81
     8.1.1 Title page (FT09F001) ..................................................................... 81
     8.1.2 Summarized solution (FT10F001 and FT21F001) .............................. 83
     8.1.3 Zooplankton iterations (FT15F001) ................................................... 86
     8.1.4 Nutrient fractions (FT14F001) ........................................................... 88
  8.2 The optional output of BLOOM II ............................................................. 90
     8.2.1 The dump output (FT06F001) ............................................................ 90
     8.2.2 Assessment of maximum grazing rates (FT16F001) ............................ 93
     8.2.3 Steady state carbon and oxygen fluxes (FT17F001) ......................... 95
     8.2.4 Diurnal oxygen budget (FT18F001) .................................................. 97
     8.2.5 Output for oxygen model OXYMOD (FT19F001) .............................. 99
     8.2.6 Live and dead phytoplankton pools (FT20F001) ............................... 101
     8.2.7 File used by the editor (FT31F001) .................................................... 103
     8.2.8 Print-plot of results (FT45F001) ......................................................... 103
9.0 Appendix 1: Help for program options ..................................................... 105
10.0 Appendix 2: Help for model coefficients ................................................ 108
11.0 Appendix 3: Help for stoichiometric coefficients ..................................... 109
12.0 Appendix 4: Help for growth coefficients ............................................... 110
13.0 Appendix 5: Help for zooplankton coefficients ........................................ 111
14.0 Appendix 6: Worksession of BLOOM II ................................................ 112
15.0 Appendix 7: Subroutine calling sequence of BLOOM II ......................... 119
16.0 Appendix 8: Subroutine description ....................................................... 122
17.0 References ............................................................................................... 139
18.0 Index ...................................................................................................... 141
TABLES

Table 1.1  Names of groups and species in BLOOM II . . . .  4
Table 1.2  Dry weight conversion, spec. ext. and Pnmax(T)  31
Table 1.3  Stochiometric coefficients of phytoplankton types  32
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Basic nutrient cycle for BLOOM II</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>Growth efficiency curves of BLOOM II</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Example of tabulated input for BLEFFPRO</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>Formats for tabulated input of BLEFFPRO</td>
<td>37</td>
</tr>
<tr>
<td>3.3</td>
<td>Example of Eilers-Peeters input for BLEFFPRO</td>
<td>39</td>
</tr>
<tr>
<td>3.4</td>
<td>Formats for Eilers-Peeters input of BLEFFPRO</td>
<td>40</td>
</tr>
<tr>
<td>3.5</td>
<td>Example of input file for BLCOMBIN</td>
<td>41</td>
</tr>
<tr>
<td>4.1</td>
<td>File definitions for BLOOM II</td>
<td>48</td>
</tr>
<tr>
<td>5.1</td>
<td>Example of standard input file FT05F001</td>
<td>58</td>
</tr>
<tr>
<td>5.2</td>
<td>Example of standard input file FT11F001</td>
<td>61</td>
</tr>
<tr>
<td>6.1</td>
<td>Differences between batch and interactive mode</td>
<td>64</td>
</tr>
<tr>
<td>8.1</td>
<td>Example of output file FT09F001</td>
<td>82</td>
</tr>
<tr>
<td>8.2</td>
<td>Example of output file FT10F001 or FT21F001</td>
<td>84</td>
</tr>
<tr>
<td>8.3</td>
<td>Example of output file FT10F001 (interactive)</td>
<td>85</td>
</tr>
<tr>
<td>8.4</td>
<td>Example of output file FT15F001</td>
<td>87</td>
</tr>
<tr>
<td>8.5</td>
<td>Example of output file FT14F001</td>
<td>89</td>
</tr>
<tr>
<td>8.6</td>
<td>Example of output file FT06F001</td>
<td>92</td>
</tr>
<tr>
<td>8.7</td>
<td>Example of output file FT16F001</td>
<td>94</td>
</tr>
<tr>
<td>8.8</td>
<td>Example of output file FT17F001</td>
<td>96</td>
</tr>
<tr>
<td>8.9</td>
<td>Example of output file FT18F001</td>
<td>98</td>
</tr>
<tr>
<td>8.10</td>
<td>Example of output file FT19F001</td>
<td>100</td>
</tr>
<tr>
<td>8.11</td>
<td>Example of output file FT20F001</td>
<td>102</td>
</tr>
<tr>
<td>8.12</td>
<td>Example of output file FT45F001</td>
<td>104</td>
</tr>
<tr>
<td>15.1</td>
<td>Subroutine diagram of BLOOM II</td>
<td>119</td>
</tr>
</tbody>
</table>
1.0 STRUCTURE OF THE PHYTOPLANKTON MODEL BLOOM II

1.1 PROBLEM DESCRIPTION

Man's activities have for a long time affected the environmental conditions on earth but never on such a global scale as witnessed during the last one or two centuries. Since then the amounts of various substances, which are released into the environment, have increased dramatically. This is partly because the human population has grown more or less exponentially, but also because industrial and agricultural production per capita grew to unprecedented levels.

Many of these substances are more or less toxic. But other chemicals, which are on the contrary beneficial to certain organisms, are also released in enormous quantities. Among these are several compounds of the elements nitrogen and phosphor which are required by species of phytoplankton, among others. As a result the concentrations of these plants have increased up to a level where they are considered a nuisance. This process, which is called eutrophication, is accompanied by several objectionable symptoms: it gives the water a green, turbid appearance; it can cause bad odours; it may harm other organism because the minimum daily oxygen level can become extremely low during the night due to phytoplankton respiration; it can even cause the water to become completely deprived of oxygen (anaerobic) when a bloom declines rapidly, since the biological degradation processes consume large amounts of oxygen; it may cause clogging of filters in water transportation systems.

In the Netherlands the situation is worse than in many other countries because (1) it is densely populated, (2) it receives a major part of its water from the polluted and nutrient loaded river Rhine and (3) most of its lakes are shallow. Hence it was decided that eutrophication would be the first major topic of the Water BASIn Models (WABASIM) joint research project of the Environmental Division of the Delta Department and the Environmental Hydraulics Branch\(^1\) of the Delft Hydraulics Laboratory. This manual describes usage of one of the results of this study: the phytoplankton model BLOOM II. Other models concern water chemistry (CHARDON) and bottom biochemistry (SEDMOD).

Many coefficients of the models were determined from measurements collected in four separate bodies of water in the drinking water reservoir 'Grote Rug' by the National Institute for Drinking water supply and the Environmental Division.

1.2 PURPOSE OF BLOOM II

Biologically speaking, phytoplankters are relatively primitive organisms. They require considerable amounts of nitrogen, phosphor, solar energy and sometimes silicon to become a nuisance. In theory each of these factors could become limiting, but the question is where and when. Also the physiological data indicate that

---

\(^1\) Due to an internal reorganization WABASIM work is currently performed by the 'Water quality and Ecology' segment of the new 'Water Resources and Environment' department.
species of phytoplankton greatly differ in nutrient requirements, efficiency of solar energy fixation (photosynthesis) and (potential) net growth rates.

To understand this complexity and predict the impacts of changing circumstances, a modified and extended version (BLOOM II) was developed from Rand's Algae Bloom Model, which was applied to the Oosterschelde sea estuary by Bigelow et al. [1977]. The model uses an optimization technique called linear programming to compute the maximum total biomass concentration of several phytoplankton species in a certain time-period consistent with the environmental conditions.

BLOOM II is unconventional for its purpose (maximizing biomass) and solution technique (linear programming), which were mainly chosen for practical reasons. There was (and still is) a strong need for a management model, which could provide answers to such questions as

- What will be the size of a potential bloom?
- Which species are likely to become dominant?
- What is the expected impact of management measures?

Dynamic, differential equation models with the same number of phytoplankton species as included in BLOOM II were out of the question when the latter was first applied and are still not operational, as far as we know:

1. They require more physiological data (for example Monod constants).
2. Their mathematical and numerical structures are more complicated.
3. They are more difficult to calibrate.
4. Their computation times are (considerably) larger.

To compute values for the environmental constraints, BLOOM II needs information on the concentrations of total available nutrients, temperature, the influx of solar radiation and certain lake-specific characteristics (depth and turbidity). These conditions can all be determined directly or indirectly from measurements and are sufficient for the model's calibration and validation. In the coupling of BLOOM II and CHARON the latter provides the total available nutrients.

Often phytoplankton species can achieve high net growth rates enabling them to double their biomass several times a week, sometimes even a day. Thus the nominal version of the model assumes steady states for phytoplankton biomass with a time-step of one week. In addition a second steady state is assumed for nutrient recycling, which in theory is less well justified. However, numerous comparisons between the results of BLOOM II and JSBACH, in which nutrient recycling is solved dynamically, have shown that in general the results of BLOOM II are not significantly influenced by this second steady state assumption. In BLOOM II, therefore, succeeding time-steps are completely independent, although slowly changing environmental conditions tend to give the model's output a smooth appearance.

These two fundamental steady state assumptions cannot be justified under all conditions. For example in waters, where average light intensities are extremely low, either due to extinction by phytoplankton, background material or a large water depth, phytoplankton growth rates are also low. To deal with violations of the steady state assumptions we have

Structure of the phytoplankton model BLOOM II
1. Extended BLOOM II with several optional alternatives.
2. Developed a pseudo-dynamical version of BLOOM II.

Three options developed in the past years are still important. The user can:

1. Increase its nominal time-step during the entire year or some parts of it.
2. Solve the equations for nutrient recycling dynamically, maintaining the steady state for phytoplankton.
3. Include constraints for the maximum growth rates of individual species during one time-step (the so called growth constraints).

This last option in particular is used very frequently.

To couple BLOOM II with other models it proved necessary to develop a pseudo-dynamical version. Ultimately one formulation was selected, which is now used for

1. The coupling between BLOOM and CHARON (JSBACH).
2. The coupling between BLOOM and DELWAQ.
3. A stand-alone, dynamic version of BLOOM.

In all these cases BLOOM II can no longer dispose over all the available nutrients and light energy in the system. Resources tied up in existing phytoplankton cells, which will survive during the next time-step, are no longer available to BLOOM. We call this the incremental version of BLOOM, because this way the model only computes the increment of each phytoplankton species. Added to the remaining levels, a new value for the total biomass of each species is obtained for a time-step. Consequently the incremental version of BLOOM maximizes the total production rather than biomass, provided that the time-step is short enough (about half a week or less).

In addition to the abiotic conditions, zooplankton biomasses may be provided to the model to compute losses due to grazing. So far no attempt was made to actually compute zooplankton concentrations with the model because the impact of zooplankton on the computed blooms in the Grote Rug cases was negligible under the assumptions that were made. These assumptions will be re-evaluated later in WABASIM.

1.3 SELECTION OF PHYTOPLANKTON SPECIES

Natural blooms often consist of assemblages of different phytoplankton species, sometimes belonging to rather distinct groups of species such as diatoms, green or blue-green algae. However, groups or even species within a group, often have rather distinct characteristics. Therefore many of the adverse impacts of eutrophication depend on the dominant (group of) species. For example a bloom of blue-green algae is usually considered far worse than one which is dominated by for example diatoms or green algae.

Some groups cannot be considered homogeneous either. This is true in particular for blue-green algae, because the impacts of a bloom of, for example, Oscillatoria agardhii are quite distinct from a bloom of Microcystis aeruginosa. A bloom of the latter is
much more likely to collapse suddenly which might lead to anaerobic conditions. *Microcystis* is also the most important blue-green algal species for which toxic effects have been reported by Gorham [1964] and others. Thus it is a major advantage if a management model includes a moderately large number of species with distinct ecological characteristics. Typically the model considers between six and ten different phytoplankton species, selected from various (Dutch) literature sources. These species may differ in their growth versus light response, temperature dependence, and requirements for nitrogen, phosphor and silicon.

Depending upon the external conditions each model species can adjust its nutrient requirements (variable stochiometry). To this purpose each model species is split into several 'types' (usually 3) corresponding to extreme conditions of for example phosphor limitation. The stochiometry of the model species is, however, not restricted to these extremes, but can vary continuously between them because the model can select any combination of a species' types. The phytoplankton groups and species, which are currently included in the model are shown Table 1.1.

Table 1.1

Names of groups and species in BLOOM II.

<table>
<thead>
<tr>
<th>Group name</th>
<th>Species name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diatoms</td>
<td>Asterionella</td>
</tr>
<tr>
<td>Flagellates</td>
<td>Chlamydomonas</td>
</tr>
<tr>
<td>Greens</td>
<td>Scenedesmus</td>
</tr>
<tr>
<td>Blue-greens</td>
<td>Aphanizomenon</td>
</tr>
<tr>
<td>Blue-greens</td>
<td>Microcystis</td>
</tr>
<tr>
<td>Blue-greens</td>
<td>Oscillatoria</td>
</tr>
</tbody>
</table>

1.4 NUTRIENT REQUIREMENTS OF PHYTOPLANKTON SPECIES

1.4.1 Nutrient mass-balances

To calculate how much phytoplankton biomass can be sustained, it is necessary to determine (1) the total amount of each nutrient in the water, and (2) the available and unavailable fractions. As indicated in Figure 1.1 on page 5, the total nutrient amounts in the water change by means of two important processes, watertransport (inflow and outflow) and exchange with the bottom. Both are modelled in JSBACH [de Rooij, in prep.]. In the stand-alone version of BLOOM II a relatively simple approach using measured concentrations has been adopted.
Figure 1.1 Basic nutrient cycle for BLOOM II. The model computes dissolved nutrients and those in live and dead phytoplankton and in zooplankton.

Define the following symbols:

- $x_k$ is the amount of phytoplankton type $k$,
- $a_{i,k}$ is the amount of nutrient $i$ per unit of type $k$,
- $y_i$ is the amount of nutrient $i$ which is temporarily unavailable for growth, because it is incorporated in dead phytoplankton,
- $z$ is the amount of zooplankton,
- $h_i$ is the amount of nutrient $i$ per zooplankton biomass unit,
- $e_i$ is the surplus amount of nutrient $i$; it is directly available,
- $b_i$ is the total amount of nutrient $i$ in the water.

For the total amount of nutrient $i$ in the water, the following mass-balance equation should hold:

$$
\Sigma_k (a_{i,k} x_k) + y_i + h_i z + e_i = b_i \tag{1.1}
$$

Estimates for $b_i$ are based upon, but not necessarily equal to, measured total concentrations. Often a considerable amount of
nitrogen (between 0.5 and 1.0 mg/l) is refractory (non-reactive), as we have found by

1. Linear regression of particulate nitrogen against chlorophyll.
2. Comparison between observed and by JSBACH computed Kjeldahl N.

Total silicon cannot be measured at all. Here we estimate this amount by adding to the (measured) dissolved fraction the estimated amount of silicon in living and dead diatoms. When the diatom biomass cannot be estimated, we usually use the winter maximum of dissolved silicon as an estimate for total available silicon at least during the spring diatom bloom.

In the case of phosphorus we make no correction whatsoever, but simply use the measured total amount, as there are no indications of a large refractory pool.

We assume that none of the b_i values change within a time-step: there is no net transport to or from the bottom and connecting waterways^2. This assumption had to be adopted as the model receives new total available nutrient concentrations for each time-step. Thus in the (exceptional) case when there is a large change in some or all of the concentrations, the model's biomass computations may be incorrect for that period, but in the next time-step these transports are implicitly accounted for in the new inputs of b_i.

1.4.2 Nutrient recycling

Frequently a substantial amount of nutrients is tied up in dead phytoplankton cells (detritus) and thus temporarily unavailable to living cells. In BLOOM II this amount is denoted by y_i. Modeling detritus is difficult for two reasons: (1) there is no way to separate detritus from other fractions in a sample, and hence the computed number cannot be compared to an observation; and (2) there is considerable uncertainty about the rates at which detritus is generated and removed.

To simplify the discussion, we shall usually refer to a single detritus pool, but actually the model distinguishes four detritus pools: one for each of the nutrients and one for chlorophyll.

A certain amount of each type k dies during each time-step, withdrawing its nutrient content from the live phytoplankton pool. Only a fraction of this nutrient remains a part of the detritus pool, however. The rest becomes directly available to growth of new individuals, because the dead cells break apart (a process called autolysis), spilling some of their contents into the water in dissolved form. Detritus may be removed to the bottom or to the dissolved nutrient pools in proportion to its concentration. We can express this mathematically as follows. Define:

- M_i is the natural mortality rate constant per day of type k, which includes all death processes except grazing,
- q_i is the nutrient fraction which becomes detritus when a phytoplankter dies, since it is not immediately released to the dissolved nutrient pool after cell lysis,
- u_i is the mineralization rate constant of detritus per day,

^2 Zero nutrient transports are only assumed by BLOOM II, but several transports are included in the BLOOM II - CHARON integrated model JSBACH.

Structure of the phytoplankton model BLOOM II
s is the sedimentation rate constant of detritus per day.

Then the rate of change of the amount of nutrient \( i \) in detritus can be described by the following equation:

\[
\frac{d y_i}{dt} = \sum_k (q_i \cdot a_{i,k} \cdot M_k \cdot x_k) - u_i y_i - s y_i \tag{1.2}
\]

For each \( q_i \) we adopt a value of 0.5. Thus, 50 percent of each nutrient becomes (almost) instantaneously available when a cell dies. The nominal value for \( s \) is 0.0, since as previously explained, BLOOM II assumes no transports to or from the water during a time-step. The remineralization rate constants \( u_i \) are computed as:

\[
\begin{align*}
    u_i &= 0.006 \times T \quad \text{(P and N)} \\
    u_i &= 0.025 \quad \text{(Si)}
\end{align*}
\]  

(1.3)

where \( T \) is temperature in degrees centigrade. More details on these coefficient values are provided by Los [1982].

Because \( y_i \) depends on the main variable of the model, the amount of phytoplankton \( x_k \), there is no simple analytic solution to Eq. (1.2) unless we make an additional assumption. One possibility is to assume that detritus achieves a steady state in one time-step of the model. As an alternative it could be assumed that the total live phytoplankton biomass is constant, which makes it possible to solve Eq. (1.2) analytically. In both cases \( y_i \) can be solved from Eq. (1.2) and both solutions are optional in BLOOM II.

Using the (nominal) steady state assumption for recycling we obtain:

\[
y_i = \frac{q_i \sum_k (a_{i,k} \cdot M_k \cdot x_k)}{u_i + s} \tag{1.4}
\]

Assuming that the total amount of nutrient \( i \) in phytoplankton is nearly constant for a week\(^3\), Eq. (1.2) may also be solved as an ordinary first order, inhomogeneous differential equation. Because the coefficients in the equations change step-wise, there is a (usually small) discontinuity in the computed \( y_i \) values at the end of each time step, when the model starts with new values. This second solution is:

\[
y_i = y_{i_{e}} + (y_{s_{i}} - y_{e_{i}}) \text{ EXP}[-(u_i + s)\Delta t] \tag{1.5}
\]

\(^3\) If the changes in \( x_k \) within a week were considered too large, the algorithm could be extended with an additional corrector step, using the average values of the \( x_k \)'s of two time-steps to recalculate the values of \( y_i \). These new value of \( y_i \) could then be used in the linear program to recompute the \( x_k \)'s, then the \( y_i \)'s could be recalculated once more etc.

Structure of the phytoplankton model BLOOM II
where:
- \( y_i \) is the present value of \( y_i \),
- \( y_{s_i} \) is the initial value of \( y_i \) at the beginning of a time-step,
- \( y_{e_i} \) is the equilibrium value of \( y_i \), calculated according to Eq. (1.2),
- \( \Delta t \) is the time-step of the model.

If the amount of detritus were always small compared to nutrients in living phytoplankton, we could simply use the steady state solution under all conditions, but this is not the case. Usually the size of both nutrient pools are of the same order of magnitude. Therefore it is necessary to discuss how closely \( y_i \) approaches equilibrium. According to Eq. (1.5) this depends on (1) the rate of approach \((u_i + s)\), multiplied by the time-step \(\Delta t\), and (2) the difference between equilibrium and initial value of the detritus concentration \((y_{s_i} - y_{e_i})\).

The characteristic time of Eq. (1.2) for the nominal version of the model (\( s \) is small or completely neglected) is \(1/u_i \) days, which is in the order of about a week in summer to a month under winter conditions. Thus the nominal one-week time-step of the model is rather short, at least in winter. Fortunately the difference between the initial and the equilibrium values of the detritus pools \((y_{s_i} - y_{e_i})\) tends to be rather small because all the important processes such as phytoplankton mortality, remineralization and changes in the total amounts of nutrients are usually rather smooth functions of time (actually of temperature). Thus errors from these steady state assumptions seem to be tolerable under ordinary conditions. Indeed it was shown in the BLOOM II report [Los, 1982] that for most natural lakes\(^4\) the results of BLOOM II using Eq. (1.4) are essentially similar to those obtained with the alternative solution Eq. (1.5). The largest deviations from steady state are found during spring diatom blooms, because (1) the initial detritus pool size is usually very small and (2) the mineralization rate of silicon is also small.

Also notice that both the steady state solution or any kind of dynamic solution depends on the ratio \(q_i M_i/(u_i + s)\). At low temperatures both the numerator and the denominator become rather small, hence the detritus pool size is very sensitive to the values of these coefficients; errors in any of these may well have a larger impact on the final results than the type of solution used in the model.

### 1.4.3 Formulation of nutrient constraints

Having obtained two solutions for the detritus equations, we can now easily eliminate \( y_i \) from the set of mass-balance equations (1.1) which can be rewritten with the phytoplankton concentrations \((x_k)\) as the only variables. For brevity, only the steady state sol-

\(^4\) The model's results for those Grote Rug reservoirs in which P is chemically reduced, show greater differences because the time-dependent simulation of detritus is strongly affected by the artificially induced variations for example in the sedimentation rates.
ution will be given here, but both are described by Los [1982] (and included in the model). Substituting (1.4) into Eq. (1.1) gives:

\[
\Sigma_k \left[ \frac{u_i + s + q_i \cdot M_k}{u_i + s} \right] a_{i,k} x_k + e_i = b_i - h_i \cdot z \quad (1.6)
\]

This set of equations will be used as constraints by BLOOM's linear programming algorithm to compute the phytoplankton biomass concentrations (See 'Solution algorithm' on page 27).

1.5 ENERGY REQUIREMENTS OF PHYTOPLANKTON SPECIES

The division into autotrophic and heterotrophic organisms is one of the most important ways to classify living creatures into functional units. Photoautotrophic organisms such as all the well known terrestrial plants, phytoplankton and many bacteria are essentially independent of other creatures for survival, because sunlight is their main source of energy. There is no doubt that life on earth depends almost completely on the photosynthetic products of the plants (energy rich compounds and oxygen), because these are required by all other organisms whether directly or indirectly (e.g. as fossil fuels).

Plant cells contain a number of light-sensitive pigments, such as chlorophyll-a and -b, which can absorb light quantum (photons). In a complicated sequence of photochemical reactions, the energy from the photons is transferred to and stored as readily available chemical energy (mainly in the form of carbohydrates). Notice that other important products of photosynthesis such as proteins are not explicitly incorporated in the model.

The rate of photosynthesis, which is one of the most complicated physiological processes, varies with circumstances and between species. There are great differences in the preferred light intensities (Figure 1.2 on page 19) and in the ability of species to adapt to new light or temperature conditions. As shown by Jørgensen [1969], van Liere [1979] and others, many adaptations involve a change in internal chlorophyll levels. This phenomenon is incorporated in BLOOM II as different types of each species have different dry weight to chlorophyll ratios.

It is difficult to compute the carbon fixation rate of phytoplankton because the light intensity varies with (1) the seasonal rhythm, (2) the daily rhythm, and (3) the water depth. The last factor is particularly important since light attenuates exponentially with depth, and because many Dutch waters have a high turbidity. Thus, the lowest light intensity at which photosynthesis is still possible (at the euphotic depth) is usually found far above the bottom. Meanwhile the light intensities in the surface layers are supersaturating to photosynthesis. An additional complication is the ability in some groups of species (blue-green algae, dinoflagellates) to avoid homogeneous mixing (See Los [1982], Sec. 5.7, and also 'The standard input of BLOOM II' on page 51). Superimposed on all variations are weather-induced light fluctuations (clouding).
As a final complication, carbon fixation and growth are not always synonymous. Photosynthesis is a prerequisite to but not a sufficient condition for growth\(^5\) because all cell materials, not just carbon must be assimilated before a cell division can occur. Under certain conditions carbon fixation may continue, although actual growth is prohibited by for example a nutrient limitation; in that case the surplus amount of fixed carbon must be excreted by the cell.

For simplicity most coefficients in the remainder of this chapter are written without a subscript \(k\), but actually they depend on the type and are treated as such by the model.

1.6 THE ENERGY BUDGET OF PHYTOPLANKTON

Phytoplankton blooms develop when, during a substantial period, the environmental conditions enable a population to fix more energy than required to compensate for all current losses. Energy gains are determined by the rate of production, which is a function of the solar intensity, the surface reflectance, the attenuation in the water (mixing depth, background extinction, contribution of living and dead phytoplankton to extinction), the day length, the water temperature, perhaps the spectral distribution, or the variations in light intensity\(^6\).

Energy is lost by several processes: respiration (to some extent the opposite of primary production), mainly for maintenance; mortality (old or unhealthy cells die); grazing by zooplankton or fish; and sedimentation (most phytoplankton species have positive sinking rates).

Define the following symbols:

- \(P_g\) is the depth and time averaged gross growth rate constant per day,
- \(R\) is the respiration rate constant per day,
- \(M\) is the natural mortality rate constant per day, including all death processes except grazing,
- \(G\) is the mortality rate constant per day due to grazing by zooplankton.

The net effect of these processes may be summarized in the well-known differential equation:

\[
\frac{dx}{dt} = (P_g - M - R - G) x
\]

(1.7)

---

\(^5\) Photosynthesis is here defined as photo-chemical carbon fixation, growth as increase in biomass (dry weight); it requires a balanced uptake of all essential elements in addition to carbon and is usually followed by a cell division.

\(^6\) Short periods with high intensities and long periods with moderate intensities might have a different impact, even if the total number of received quanta is the same.
Notice that no functional relations are indicated, which could suggest there are no feedback mechanisms to prevent unlimited growth. Actually the gross growth rate \( P_g \) depends on the light intensity and therefore on the attenuation caused by the phytoplankton cells themselves. Thus as the number of cells grows, the light intensity (and hence the energy available) at any depth is reduced, a phenomenon called self-shading. \( P_g \) also depends upon the nutrient availability and declines when nutrients are scarce. However, when the light intensity is optimal, nutrients are abundant and all other conditions are favorable, phytoplankton will achieve a maximum gross growth rate (\( P_{g\max} \)), which we can think of as an innate characteristic of each species of phytoplankton.

Logically our next section should discuss the maximum gross growth rate. Unfortunately, it is much easier to measure the maximum net growth rate, which is the maximum gross rate less respiration and perhaps mortality, depending on how the net growth rates were determined. It is the net rate we discuss next.

### 1.6.1 Maximum growth and temperature

The maximum gross growth rate, \( P_{g\max} \), cannot easily be observed directly, since phytoplankters always respire. But under favorable conditions, some forms of mortality (for example sedimentation and grazing) can be eliminated for all practical purposes, leaving a maximum net growth rate, \( P_{n\max} \).

In a classical paper on the relation between temperature and growth of phytoplankton species in the sea, Eppley [1972] plotted more than a hundred observations on the maximum number of doublings per day \( (U_{\max}) \) from continuous cultures of different species against \( T \) and then drew an envelope curve through the highest points at each temperature. This curve was described by the following equation:

\[
U_{\max}(T) = \exp(0.0639T - 0.16)
\]  

(1.8)

It is easy to demonstrate that one must multiply the doublings per day by the logarithm of 2 to obtain the daily growth rate. Thus the maximum net growth rate can be expressed as:

\[
P_{n\max}(T) = \log_2 \exp(0.0639T - 0.16)
\]  

(1.9)

This curve has become sort of a standard to which observations on individual species are often compared. Many of these data fall below Eppley's curve, but a similar temperature dependence was observed frequently for several groups of salt and fresh water species, for example by Foy et al. [1976], Goldman and Carpenter [1974], Sakshaug [1977] and Harris [1978], and was also computed for Grote Rug from measurements by the Environmental Division.

Until 1983 \( P_{n\max}(T) \) of small species in BLOOM II was computed according to Eppley's equation. For large species such as dinoflagellates and blue-green algae the same temperature dependence was used, but the value of \( P_{n\max}(T) \) was reduced by a volume dependent term of (approximately) 0.70 because \( P_{n\max}(T) \) tends to decrease with cell volume. These formulations are described by Los [1982].
In recent years a lot of data have been collected on growth rates of individual species for example during the WABASIM incubator research project at the Microbiological Department of the Amsterdam University. Unfortunately interpreting these data is not always straightforward. Several complicated processes influence growth rates, but the relations between these processes are not always apparent. Thus a significant part of this information cannot yet be used in a multi-species model [Los in prep.].

From recent experiments we can reach some conclusions, however:

- Maximum growth rates of individual species show rather large differences over the entire range of measured temperatures.
- Species differ strongly in their temperature dependence. For some of them \( P_{\text{max}}(T) \) rapidly approaches zero at moderate temperatures between 5 and 10°C.
- For some species \( P_{\text{max}}(T) \) is not a continuous function of temperature.

Using (1) data from the Incubator Research Project, (2) literature data, and (3) calibration, (in this order) we have established the set of maximum growth rates for the phytoplankton types in the model, which is shown in Table 1.2 on page 31.

Some final remarks:

1. We have used exponential and linear functions for different species.
2. Nutrient limited types not only have lower maximum growth rates but also a stronger temperature dependence. This is because the nutrient requirements of phytoplankton increases with decreasing temperatures.
3. There is little information on growth rates between 0 and 5°C [Reynolds, 1984] to which unfortunately the model is very sensitive.

More details will be given by Los [in prep.].

Assuming that observations on net maximum growth rates do not include respiration or natural mortality, we can finally calculate the maximum gross growth rate \( P_{\text{max}}(T) \), to be:

\[
P_{\text{max}}(T) = P_{\text{max}}(T) + R(T) + M(T)
\]

(1.10)

1.6.2 Respiration

To obtain the maximum gross growth rate, we must add the respiration rate to \( P_{\text{max}}(T) \). There are two different respiratory processes: photorespiration and dark respiration, of which only the latter will be discussed here, as, according to Harris [1978], the rate of photorespiration of phytoplankton is presumably small under conditions of a bloom. In the process known as dark respiration, substrates of the previous light period are used by the cell to synthesize energy-rich compounds and carbon skeletons. Unlike its name suggests, dark respiration proceeds during the light at approximately the same rate in many species.
Strickland [1960], Harris [1978] and many others have shown that instantaneous dark respiration rates are difficult to measure and moreover the usually employed $^{14}$C method to estimate primary production does not allow respiration to be measured as well. Thus there are comparatively few data on respiration in the literature.

In all previous versions of the model we have used considerably larger respiration rate constants for eukaryotic (non-blue-green algal) species than for prokaryotic (blue-green algal) species [Loo, 1982]. In accordance with this the respiration values obtained during the Incubator Research Project indicate a difference of about a factor 8 at 20°C. The respiration rate constant for the three eukaryotic species equals 0.192 per day, in contrast with a value of only 0.024 per day for the three prokaryotic species.

The Incubator Research Project gives no information on the temperature dependence of these rates, but Harris [1978] reports Q10 values between 2.3 and 2.9. For eukaryotic species we have chosen a value of 2.5. For blue-greens we have used a smaller Q10 value of 2.0, but since their rate of respiration is so small, the temperature coefficient is unimportant.

Thus we arrive at the following equations:

For eukaryotic species: $R(T) = 0.031 \times 1.096^T$  
(1.11)

For prokaryotic species: $R(T) = 0.006 \times 1.072^T$

1.6.3 Natural mortality

Although respiration and natural mortality are both loss terms, they are physiologically quite distinct. Respiration is mainly (though not entirely) a loss of carbonic substances (energy). But a healthy cell will avoid respiring proteins for example, unless conditions become extremely unfavorable. Thus respiration will not affect any of the nutrient cycles in the model.

After a number of divisions, particularly under unfavorable conditions, any phytoplankter stops producing new cells and will eventually die. Afterwards it disintegrates and releases its cell contents to the surrounding water. A certain fraction will be released in a form, that is available to living cells (our present estimate is 50 percent, (See 'Nutrient recycling' on page 6), but the rest will be lost temporarily to the detritus pool and has to be remineralized. Thus mortality has two effects on a phytoplankton population: like respiration, it is a loss of energy, but because the whole cell disintegrates, it is also a loss of cell material (available nutrients).

There is much less knowledge on mortality rates than on growth (production) or respiration rates. This is mainly because there are no direct ways to measure natural mortality. Even under a micro-

---

7 What We have called 'Respiration' here is called 'Maintenance Energy' (Ue) in the reports of the Incubator Research Project.
scope it is hardly possible to determine which cell is dead and which is living. Hence mortality rates are usually approximated by indirect methods, for example by subtracting observed net changes in biomass from measured rates of primary production. Employing this indirect method to Grote Rug data for WABASIM it was concluded that (1) both mortality and production rates are large (up to 1.0 per day) compared to the observed net rates of change in phytoplankton populations (usually less than 0.1 per day), (2) $M(T)$ is exponentially correlated with the water temperature but (3) exceptionally low values can occur at any temperatures.

As explained by Los [1982] we have determined a so called minimum estimate $M_{\text{min}}(T)$ from these Grote Rug data by drawing an envelope curve through the lowest points in a plot of mortality rates against temperature:

$$M_{\text{min}}(T) = \exp(0.0742T - 2.9678)$$  \hspace{1cm} (1.12)

Growth rates in the current model version are considerably below the values used in older model versions. In recent calibrations we have modified (1.12) only slightly, mainly because detritus production should not decrease dramatically in JSBACH. Thus we arrived at the following equation,

$$T^{-1} M(T) = 0.065 \times 1.07$$  \hspace{1cm} (1.13)

which according to the model results can no longer be regarded as a minimum estimate for the natural mortality.

With the introduction of phytoplankton types with their own (maximum) growth rate functions and stochiometric coefficients it would have been possible to introduce type-specific mortality functions as well. We have decided not to do this for the time being, mainly because we still had insufficient information. There are some strong indications, however, that knowledge on loss processes is currently progressing rather strongly. Reynolds [1984] for example devotes an entire chapter to this subject. Recent information will be considered within WABASIM at the end of 1985. It is expected that a more sophisticated approach will follow from this.

### 1.6.4 Grazing by zooplankton

Oligotrophic lakes are often characterized by a classical food-web, in which the primary producers such as phytoplankton are eaten by herbivores such as zooplankton. The herbivores get eaten by omnivores and carnivores etc. In these systems zooplankton seems to contribute significantly to the mortality rate of phytoplankton.

In many eutrophic lakes nutrients are so abundant that the phytoplankton species can achieve extremely high biomass levels. Once a bloom is present there is not enough zooplankton to affect the phytoplankton significantly and moreover the specific rate of increase of zooplankton is usually too low to make it a main source of phytoplankton mortality. Perhaps even more important: blooms in
eutrophic lakes are usually dominated by large-sized species such as blue-green algal colonies or *Ceratium*. There is no agreement in opinion whether these groups are eaten at all, but many zooplankton species either prefer other prey species, or they cannot even handle these large-sized preys.

Hence it was decided for the WABASIM project to start with a simple, pragmatic approach which would only be abandoned, if we were to find that zooplankton is crucial to BLOOM II's performance. In this initial approach we would not try to model the zooplankton biomass, but use it as input to the model, trying to estimate its effect on the computed phytoplankton levels.

Before we explain how we compute the grazing rate constant in the model, we must first consider some indirect effects of zooplankton. In their bodies zooplankters contain nutrients which are not directly available to phytoplankton cells. We can simply take this effect into account, however, by subtracting the amount of each nutrient in zooplankton from the total available amount as shown in Eq. (1.6). Also, like every other particle, zooplankton cells might contribute to the extinction of the water. As (1) the number of zooplankton cells is usually small in comparison with the total number of particles and (2) these cells do not contain large amounts of pigments, the effect of zooplankton on the extinction can be neglected.

To calculate the specific grazing rate constant $G_k$, we shall use an equation derived from Scavia and Eadie's [1976]:

$$G_k = \frac{ZG \cdot z \cdot (Xed - \sum_k PR_k \cdot xmin_k)}{(ZK + Xed - \sum_k PR_k \cdot xmin_k) \cdot Xed} \tag{1.14}$$

in which

- $G_k$ is the overall mortality rate constant due to grazing by zooplankton of phytoplankton type $k$ per day,
- $ZG$ is the filtration rate constant of zooplankton in mg dry weight of phytoplankton per mg dry weight of zooplankton per day,
- $z$ is the total amount of zooplankton in mg dry weight of zooplankton per m$^3$,
- $Xed$ is the total biomass of all phytoplankton types which are potentially grazed: edible types,
- $PR_k$ is the relative preference of $z$ for type $k$ (from 0.0 to 1.0),
- $xmin_k$ is the amount of type $k$ in mg dry weight per m$^3$ which escapes grazing,
- $ZK$ is a zooplankton constant, closely related to a half saturation constant in mg dry weight of phytoplankton per m$^3$.

To compute $G_k$ from Eq. (1.14) we must know $x_k$: the amount of each type $k$ during a time-step. Obviously, this number is known only at the end of each time-step but $G_k$ should be known in advance. Therefore, we have developed an iteration scheme in which we start with a computation of $x_k$, then compute $G$, re-compute $x_k$ etc., until certain stop criteria are met. This procedure is explained by Los [1982] in Chapter 7.
With the exception of Grote Rug, BLOOM II has never been applied to a lake for which an adequate set of zooplankton data was available. This, of course, makes it difficult to calibrate or validate its grazing computation. Also it proved to be extremely difficult, to find reliable values for the various zooplankton parameters. As nominal values we have taken:

\[ ZK = 2000 \text{ mg dry weight/m}^3 \]
\[ X_{\text{min}} = 250 \text{ mg dry weight/m}^3 \]
\[ ZG = 1.0/\text{day} \]

From an analysis of the results for Grote rug with the 1982 version of the model we concluded that the overall importance of zooplankton to its performance was limited in the cases that were considered. In over 80 percent of all time-periods (335 out of 416 weekly results) the initial solution of BLOOM II consisted exclusively of inedible species. When edible species were present before the grazing iterations (20 percent of all periods), they remained in the bloom in 12 percent of all periods and even increased in 8 percent of all periods. Grazing enhanced a switch in dominance from edible to inedible species in only 7 percent of all periods, but in terms of total biomass, the differences were not significant.

Of course, because zooplankton is an input to the model it cannot be applied to predict the importance of zooplankton in future situations, but we may investigate its potential role by running the model for different scenarios.

To investigate the potential role of zooplankton we have further included an option 'MAXGRA' in the program of BLOOM II to calculate the maximum potential grazing rate constant of type \( k \): \( G_{\text{max}} \). (See 'Assessment of maximum grazing rates (FT16F001)' on page 93). When \( G = G_{\text{max}} \), the steady state biomass of type \( k \) is zero. If we rewrite Eq. (1.14) to solve \( z \) and put \( G_{\text{max}} \) for \( G_k \) into the right hand side, we can also compute the corresponding zooplankton concentration. It came out that usually \( z \) had to be considerably larger than the maximum values presently observed in order to decrease the biomass of edible phytoplankton types to zero.

It should be noted that a systematic investigation on the role of zooplankton has not been repeated with the current version of the model. As currently the growth rates are smaller than previously, the importance of zooplankton might have increased. Thus the role of zooplankton will be re-evaluated in the near future within WABASIM.

1.6.5 Growth rates

The growth rates is determined by many environmental conditions, which moreover have a different impact on different types. Therefore each mathematical equation will always be a (rather simple) abstraction of reality. In the following the maximum gross growth rate constant, \( P_g \), will be written as the product of various terms, which presumably act independently.

Given sufficient nutrients, \( P_g \) is a function of temperature and light, which achieves a single maximum for any given temperature \( T \), called \( P_{g \text{max}}(T) \) at an optimal light intensity that we call \( I_{\text{opt}} \). If
only a small portion (e₁) of nutrient i is left available, Pg could decrease by a factor g(e₁), which is equal to or less than one according as e₁ is large or small. Similarly, if light energy is scarce, Pg will decrease by a factor of E(I,T), which depends on both temperature T and light intensity I:

- I is the total radiation in Joules/m²/hr of the photosynthetically active part of the light spectrum.

Pg may be calculated according to:

\[ Pg(I,T,e₁) = E(I,T) * Pg_{\text{max}}(T) * g(e₁) \]  \hspace{1cm} (1.15)

Various expressions (e.g. the so-called Monod and threshold approaches) have been proposed for g(e₁). But it was argued by Los [1982] that BLOOM II does not require an explicit functional form for g(e₁), but rather:

\[ g(e₁) = 1 \ \text{for each type limited by energy} \]  \hspace{1cm} (1.16)

\[ g(e₁) < 1 \ \text{for each type limited by a nutrient} \]  \hspace{1cm} (1.17)

Hence in the case of an energy limitation, nutrients do not affect growth rates; in the case of a nutrient limitation, the bloom will be calculated according to Eq. (1.6) as if the growth rate would remain sufficiently high for phytoplankton to achieve the nutrient limited bloom level within a time-step. This approach, of course, is consistent with the purpose of the model.

Notice, that the previous argument holds for phytoplankton types, not necessarily for phytoplankton species. When a nutrient becomes exhausted, the composition of the bloom might switch from an energy limited type of species x to a nutrient limited type of that same species. The latter has a lower maximum growth rate and a correspondingly smaller energy limited upperbound. In other words, the biomass level that may ultimately be achieved by a species x is indeed affected by the available amounts of nutrients.

1.6.6 Growth efficiency and light

Since the mid 1950s, when techniques to measure photosynthetic rates were greatly improved, many results have been obtained relating primary production to light intensity. Some of these earlier results, especially those of Ryther [1956], have become classical and are reproduced by Parsons and Takahashi [1973] and Bougis [1974] in their books on aquatic ecology. Usually production is an optimum-type function of light. The production rate increases with increasing intensity I, as long as I is below Iopt. For intensities above Iopt, the production rate decreases as intensities increase. This same type of curve is also found for the growth rates of phytoplankton species as a function of the light intensity. As we have
explained earlier, BLOOM II considers growth rather than primary production.

To compare the growth curves of different species, we standardize them (i.e. divide by $P_{\text{max}}(T)$). The standardized curves, which we call efficiency curves, range from 0.0 to 1.0. It is obvious from Eq. (1.15) that without nutrient limitation this scaling procedure exactly gives us the function $E(I,T)$ for BLOOM II at a particular temperature.

Because several species are included in BLOOM II, an efficiency curve should be specified for each one of them. Before 1983 no efficiency curves were available which had been measured under standardized, laboratory conditions. Instead we therefore used photosynthetic efficiency curves measured in an incubator by the Environmental Division with samples taken from Grote Rug. We made the assumption, that the shape of these photosynthetic efficiency curves would also hold for the growth efficiency of the phytoplankton species.

The results of the so-called WABASIM Incubator Research Project became available in 1983. During this project the growth rates of six phytoplankton species were determined at different temperatures, light intensities and day lengths at the Microbiological Laboratory of the Amsterdam University. The current efficiency curves of the model were determined from the results of these experiments.

---

8 Strictly speaking we should specify a different efficiency curve for each phytoplankton type. We have, however, assumed that all types of a species have the same efficiency curve.
Figure 1.2 Growth efficiency curves of BLOOM II. Data derived from measurements by the Microbiological Laboratory of the Amsterdam University.

Given sufficient nutrients, $g(e_i)=1.0$ for each nutrient $i$. Hence it follows from Eqs. (1.15) and (1.16) that:

$$P_g(I,T) = E(I,T) \times P_{g\text{max}}(T)$$

and:

Structure of the phytoplankton model BLOOM II 19
E(I,T) = Pg(I,T) / Pgmax(T)  

(1.18)

To compute the average efficiency in the model, E(I,T) must be integrated over I, as will be shown in 'Averaging the production.' As, however, Pgmax(T) is an exponential function of temperature, E(I,T) is a function of T too. Thus we shall find a different value for the average efficiency for each temperature, even if I is constant. This is a great computational disadvantage, which could force us to reintegrate E(I,T) for each temperature.

It was demonstrated in the BLOOM II report [Los, 1982] that for physiological reasons E(I,T) and Pgmax(T) should have exactly the same temperature dependence and that indeed this was the case in the photosynthetic measurements for Grote Rug. Thus it could be shown that it is possible to transform E(I,T) to a function of a single variable I', where

\[
E(I') = E(I,T)
\]

(1.19)

and:

\[
I' = I \times \frac{Pgmax(Tref)}{Pgmax(T)}
\]

(1.20)

where

- Tref is the temperature, at which the efficiency curves have been determined.

Notice that I'=I at T=Tref, which is 20°C in the incubator results (We might call I' an equivalent intensity). To compute the average efficiency at a given temperature T and light intensity I, the model (1) integrates the standard, 20°C efficiency curves and puts the results in a table, (2) transforms the light intensity I to the corresponding 20°C value I', and (3) looks up the average efficiency at this transformed intensity level I' (and not I).

1.6.7 Averaging the production

The light intensity encountered by, and hence the efficiency E(I,T) of, a phytoplankton cell is not constant but varies with the water depth, turbulence and time. To account for these variations, the model must compute the average efficiency EAVG in a certain period. Several biological and mathematical complication which are involved in the averaging procedures are discussed by Bigelow et al. [1977] and by Los [1982]. Introducing a light function F, a day length function L and defining the following symbols:

- DL is the average day length in hours,
- Is is the surface light intensity in joules/m²/hr,
- Zmax is the maximum mixing depth of the water,
- K is the total extinction of the water per m.

Structure of the phytoplankton model BLOOM II
- $K_I$ is the total extinction due to living phytoplankton per m,
- $K_d$ is the total extinction due to detritus per m,
- $K_b$ is the background extinction of the water per m,

$EAVG$ may ultimately be computed as:

$$EAVG = \frac{L_DL}{L(24)} + \frac{F(K, Z_{\text{max}}, I_s)}{K \cdot Z_{\text{max}}} \quad (1.21)$$

In this equation,

$$K = K_I + K_d + K_b \quad (1.22)$$

Functions $F$ and $L$ will be explained by Los [in prep.] in the second BLOOM report. Notice that

1. $EAVG$ is inversely related to the product of $K$ and $Z_{\text{max}}$ at normal light intensities. Thus when a bloom goes up and $K$ rises, $EAVG$ becomes progressively smaller (self-shading).
2. $EAVG$ is not assumed to be a linear function of the day length.

The latter means that the light intensity patterns is important in BLOOM II. Thus for example the growth rate at a surface light intensity $I_s$ during a period of 16 hours is not twice as large as the growth rate at that same surface light intensity ($I_s$), but during a period of 8 hours. For most species the difference is considerably less than a factor two, because growth as a function of day length saturates.

The results of Eqs. (1.21) and (1.18) could be substituted into the differential equation (1.7) for each type $k$:

$$\frac{dx_k}{dt} = [P_{\text{max}}(T)_k \cdot EAVG_k - M(T) - R(T)_k - G_k] \cdot x_k \quad (1.23)$$

Using the appropriate expressions for $P_{\text{max}}(T)$, $EAVG$, $M(T)$ and $R(T)$, it is now possible to solve Eq. (1.23) numerically. There is no analytical solution, however, because $EAVG$ is a function of $K$, hence a (complicated) function of $x$. As we shall see in the next section, this equation is not solved in the nominal version of the model. It is used, however, to compute the right hand sides of the growth constraints (See 'Formulation of growth constraints' on page 25).

### 1.6.8 Steady state assumption

Rather than solving Eq. (1.23) BLOOM II assumes a steady state with a nominal time-step of a week. This assumption is crucial to
the model and therefore deserves some explanation. Rapid fluctuations in phytoplankton populations have been observed frequently, under both laboratory and natural conditions. Usually, however, concentrations change more gradually, which could be caused by a slow approach to some steady state, but also by slowly altering environmental conditions.

To investigate, how rapid the approach to equilibrium could be under various conditions, Eq. (1.23) was solved numerically with coefficient values which are typical for various parts of the year and with several (small) initial biomass concentrations. It was concluded that in summer, when usually the worst blooms occur, phytoplankton populations respond rapidly to changes in environmental conditions, although a complete shift in species dominance usually takes more than one week.

In spring, two or three weeks may be required to approach equilibrium; large fluctuations in species composition generally take still longer periods. Under (severe) winter conditions, biomass can only change slowly, implying that equilibrium can only be approached in a week, if the initial value is already close to it. Changes in species composition are therefore unlikely.

Considering the purpose of the model (a management model to predict size and composition of objectionable blooms), we concluded that a steady state with a one-week time-step could reasonable be assumed in the stand-alone model version because the worst blooms occur in summer. But there are obviously periods, when this time-scale could be too short, particularly if there is a shift in dominance. Whether the model overpredicts during these periods is determined, however, not only by the growth rate, but also by the initial biomass. If the steady states of succeeding periods have about the same value and composition, the model's prediction may still be correct.

There are two basically different approaches to investigate (possible) deviations from equilibrium:

1. Use the alternative options for the time-scale, nutrient recycling and the maximum possible growth rate of individual species as was already mentioned in 'Purpose of BLOOM II' on page 1. A more detailed description of these options can be found in the BLOOM II report [Los, 1982].

2. Use the incremental version of BLOOM II. The default time-step of this version is one day.

Note: in JSBACH, the time-step is variable, but usually half a week. In the coupling of BLOOM II and DELWAQ, the time-step is variable, but usually one day. The steady state assumption in the incremental versions of the model therefore has a completely different meaning. Here equilibrium is assumed for production during a much shorter period of time than one week. Time dependence of the model will be investigated in the near future.

Under the assumption that each type k is at steady state, obviously

\[
\frac{dx}{dt} = 0 \quad (1.24)
\]

which can only be true if either:

Structure of the phytoplankton model BLOOM II
\[ \begin{align*}
\text{Pg} - M(T) - R(T) - G &= 0 \quad (1.25) \\
\text{or:} \\
x &= 0 \quad (1.26)
\end{align*} \]

For each type \( k \) either (1.25) or (1.26) should hold. Substituting (1.16) and (1.17) for \( g(e_i) \) into (1.15) it follows that

\[ \text{Pg} \leq E(I,T) \text{ Pgmax } (T) \quad (1.27) \]

which after substitution into (1.25) gives:

\[ E(I,T) \geq \frac{M(T) + R(T) + G}{\text{Pgmax}(T)} \quad (1.28) \]

Any type for which (1.28) does not hold, has a negative net growth rate and is unable to sustain itself; thus it is excluded from the bloom. The value of \( E(I,T) \) for which (1.28) is an equality is called the minimum efficiency requirement \( \text{Emin}(T) \). This is the lowest possible value of \( E(I,T) \) at which gains and losses are balanced.

Next EAVG is set equal to \( \text{Emin}(T) \) and then Eq. (1.21) is solved for \( K \). Due to the shape of the efficiency curves, there are usually two roots, one called \( U_{kmin} \), the other \( U_{kmax} \); the first is the limit where the light intensity becomes too high (photo-inhibition), the second where it becomes too low (energy limitation). Since these two roots only include the physiological responses of the model's types to the light regime as if there were no background extinction, the latter must be subtracted from both roots, hence:

\[ \begin{align*}
K_{\text{min}} &= U_{kmin} - K_b \\
K_{\text{max}} &= U_{kmax} - K_b
\end{align*} \quad (1.29) \]

A type can only sustain a positive net growth rate, if the total extinction is between its \( K_{\text{min}} \) and \( K_{\text{max}} \) value\(^9\):

\[ K_{\text{min}} \leq K \leq K_{\text{max}} \quad (1.30) \]

Under unfavorable conditions there may be no root for \( K \), implying that the average light intensity is too low (or, conceivably, too high) for production to compensate total losses.

\(^9\) In most eutrophic waters \( K_{\text{min}} \) of any type \( k \) is smaller than the background extinction, hence the average light intensity is too low to exclude a phytoplankter due to photo-inhibition.
1.6.9 Formulation of energy constraints

Besides absorption by background material for which Eq. (1.29) is corrected, the model must also take the extinction by dead phytoplankton particles into account. Defining:

- $v$ is the rate constant, by which the effect of dead algal material on the extinction diminishes,
- $K_k$ is the specific extinction of living and dead material of phytoplankton type $k$ in $m^2/mg$ dry weight,

and reintroducing subscripts, one may write in analogy to the detritus equations for nutrients described in 'Nutrient recycling' on page 6:

\[
\frac{dK_d}{dt} = \sum_k (q \cdot M_k \cdot x_k) - v \cdot K_d - s \cdot K_d \tag{1.31}
\]

As discussed in the BLOOM II report [Los, 1982] the model assumes that (1) $q = 0.5$: dead phytoplankton particles lose 50 percent of their light absorbing properties almost instantaneously and (2) $v$ can be calculated as an exponential function of temperature:

\[
v = \exp(0.0296T - 1.897) \tag{1.32}
\]

Also since this equation yields values between 0.15 and 0.30 for the normal temperature range, it is reasonable to assume $K_d$ to be at steady state. Thus:

\[
K_d = \frac{\sum_k (q \cdot M_k \cdot x_k)}{v + s} \tag{1.33}
\]

Using the temperature dependences for $M$ and $v$ and a value of 0.5 for $q$, we can easily verify that $K_d$ is in the order of 0.20 to 0.40 times the extinction of live phytoplankton $K_l$. Thus the computations are less sensitive to errors in $K_d$ than they are to errors in the steady state detritus pools of the nutrients.

There is one final complication for species with buoyancy control, because living and dead phytoplankton do not have the same depth profile. Assuming a homogeneous distribution for dead phytoplankton, we must multiply $K_d$ by the fraction of the depth ($R_{mix, k}$) over which each species is presumably mixed homogeneously. Hence

\[
K_d = \frac{\sum_k (q \cdot M \cdot R_{mix, k} \cdot K_k \cdot x_k)}{v + s} \tag{1.34}
\]

It has proved to be more difficult to model energy by linear constraints than to model nutrients. However, the transformation of light into extinction makes it possible to use energy and nutrient...
equations in the same mathematical framework. The energy constraints of type \( k \) follow logically from condition (1.30), substituting the appropriate expressions for living and dead phytoplankton for \( K \):

\[
K_{\text{min}_k} \leq \Sigma_k K_k x_k + K_d \leq K_{\text{max}_k} \tag{1.35}
\]

Substitution of Eq. (1.34) into (1.35) gives the final energy constraints:

\[
K_{\text{min}_k} \leq \Sigma_k \left[ \frac{v + s + q \cdot R_{\text{mix}_k} \cdot M}{K_k x_k} \right] \leq K_{\text{max}_k} \tag{1.36}
\]

1.7 FORMULATION OF GROWTH CONSTRAINTS

In the previous version of BLOOM II we had included an option to add a set of growth constraints to the normal set of nutrient and energy constraints. These growth constraints would set an upperbound to the biomass increase of each species in a time-step taking both its initial value and its growth rate into account. Hence we would determine one growth constraint for each variable (= phytoplankton species) of the model.

In the current version of the model phytoplankton types have replaced species as basic variables. This means that without any modification the model would determine one growth constraint for each type. This, however, does not make sense as it would often prevent switches between types of the same species. But these switches are essential to the concept of variable stochiometry in BLOOM II. There is no doubt that switches between types (hence adaptation) require some time. But if we accept the fact that individual cells can adapt or at least can produce adapted daughter cells, then we should use a single growth constraint for each species rather than for each type.

The right hand side of the growth constraint (the upperbound of each species' biomass) depends on its

1. Initial biomass level at time zero.
2. Growth rate constant during a time-step.

However, in the current version of the model we distinguish different types with among others different growth rate constants. So the question arises which growth rate should be used to compute the biomass upperbound:

- The (high) growth rate of the energy limited type?
- The (low) growth rate of one of the nutrient limited types?
- Something in between?

We have decided to use the highest growth rate (hence of the E-type) to compute the upperbound of the biomass of species \( j \). The rationale behind this is the following. Species which are growth limited, are increasing rapidly due to a change in environmental
conditions which is favorable to them. Hence it seems more logical to assign the highest possible growth rate to them than the much lower value of a type approaching a nutrient limited steady state value.

We shall now give the mathematical formulations. Define the following symbols:

- \( x_{j,k} \) is the biomass of phytoplankton type \( k \) of species \( j \),
- \( r_{j,k} \) is the net rate of increase of type \( k \) of species \( j \) at the beginning of a time-step,
- \( x_{s,j} \) is the initial total biomass of phytoplankton species \( j \),
- \( x_{p,j} \) is the total biomass of phytoplankton species \( j \) at the end of a time-step,
- \( x_{\text{lim},j} \) is the maximum level which can be attained by species \( j \) in one time-step of the model.

To establish the growth constraint for species \( j \) we determine

1. The total biomass of species \( j \) (\( x_{s,j} \)) at the beginning of the time-step as

\[
x_{s,j} = \sum_k (x_{j,k})
\]  
(1.37)

2. The maximum net rate of increase \( r_j \) of species \( j \) as

\[
r_j = \max_k (r_{j,k})
\]  
(1.38)

3. The upperbound \( x_{\text{lim},j} \) as

\[
x_{\text{lim},j} = x_{s,j} \exp(r_j \Delta t)
\]  
(1.39)

See also Eq. (1.41). If the total biomass at time zero equals zero, we put in some base level to prevent \( x_{\text{lim},j} \) from becoming zero, which would imply that no new species would ever enter the bloom.

4. The growth constraint of species \( j \) as

\[
\sum_k (x_{j,k}) \leq x_{\text{lim},j}
\]  
(1.40)

### 1.7.1 Net rate of increase

So far we have used the term 'net rate of increase' without indicating how it is determined by the model. To compute this rate we solve Eq. (1.23) the basic differential equation of the model giving
Putting $\Delta t$ to the time-step of the model and fixing $EAVG_j$ at its initial value, Eq. (1.41) gives us the upperbound $x_{lim,j}$ for species $j$.

As was shown earlier (See 'Averaging the production' on page 20) the average efficiency $EAVG_j$ depends on the light intensity in the water and thus on the attenuation by phytoplankton. Since $EAVG_j$ gets smaller as the extinction get larger, it can be very difficult if not impossible for new species to replace the ones that currently dominate. To give some indication: new species can according to Eq. (1.41) often increase by no more than 20 percent per week due to the high extinction level.

In natural waters changes in species dominance may not always occur in one or two weeks, but there are numerous examples of dramatic changes within periods of one or two months. Thus it seems obvious that simply using Eq. (1.41) would make our model too rigid. This rigidity has in fact always been one of the main objections against the first versions of JSBACH. We therefore compute $x_{lim,j}$ from Eq. (1.41) under the assumption that the mortality rate is zero. We think this is a reasonable assumption because species who just start growing rapidly usually consist of young and healthy cells so that they can grow exponentially for a period of time. Under these conditions their mortality rate constant is probably much lower than the nominal, temperature dependent value computed with the model.

Notice that the assumption $M_j = 0.0$ is only used for the construction of the growth constraints, but not in any of the other constraints. So as soon as a species reaches some steady state value, all its types die according to their normal mortality function. By that time the value of the right hand side of the growth constraint is, however, no longer of importance.

### 1.8 SOLUTION ALGORITHM

The energy constraints, which were set up in 'Formulation of energy constraints' on page 24 permit each species to grow within certain limits of the total extinction coefficient of the water. As shown in 'Steady state assumption' on page 21 usually $K_{min,k}$ is negative for each type $k$ and $K_{max,k}$ is positive for at least one phytoplankton type. Thus the feasible ranges of extinction limits of the different types generally overlap.

In the solution algorithm of BLOOM II, all $K_{min}$ and $K_{max}$ values are ordered, resulting in $2N-1$ or less extinction intervals, if there are $N$ types in the model. Below the smallest $K_{min}$ value, no type can grow and of course if $K$ exceeds the largest $K_{max}$ value, no type can grow either. In between these absolute limits, each interval has a different set of types associated with it, which could have positive growth rates. In general most types can grow in the lowest intervals, but as the extinction increases, only a few can maintain a positive net growth rate, as self-shading becomes limiting to an ever larger number of types.

Because the total extinction cannot possibly have two different values at the same time, a separate LP is set up and solved for each interval, always with the same three nutrient constraints, because these should hold regardless of the energy conditions. Also the (optional) growth constraints of the species, which put an upper-
bound \((x_{lim})\) to the biomass concentration that can be attained in a time-step are kept constant in each interval (See 'Formulation of growth constraints' on page 25). Finally the coefficients of the extinction rows are the same in each interval, because the light absorption per unit of living and dead phytoplankton does not vary between intervals. We have used these coefficient similarities to increase the efficiency of the solution algorithm (See 'Solution of the linear programs' on page 78). Thus BLOOM II solves the following LP for each interval \(l\), which is characterized by its extinction limits and its subset of types \(S_l\):

\[
\text{Find:} \quad x_k \geq 0 \quad \text{and} \quad e_1 \geq 0, \quad \text{(for subset } S_l) \\
\text{Maximizing:} \quad \Sigma_k x_k, \\
\text{Subject to:} \\
\Sigma_k \left[ \frac{u_i + s + q_i \cdot M_k}{u_i + s} \right] a_{i,k} \cdot x_k + e_i = b_i - h_i \cdot z \\
K_{min} \leq \Sigma_k \left[ \frac{v + s + q \cdot R \cdot M_k}{v + s} \right] K_k \cdot x_k \leq K_{max}
\]

\(x_k \leq x_{lim}\)

We then find the maximum total biomass \(B_{max}\), the type composition and limiting factors. After each LP for a time-step has been solved, we compare their \(B_{max}\) values; the interval with the highest \(B_{max}\) value contains the ultimate solution of BLOOM II, the others can be considered as local maxima. This procedure is repeated for each single time-step.

1.9 BEHAVIOR AND PERFORMANCE

BLOOM II has been set up in a way, that makes overpredictions more likely than underpredictions: it computes maximum equilibria rather than actual biomasses. Indeed results of previous model versions were often higher than observed during a substantial part of the year, but agreed well with the measurements during phytoplankton peaks. This pattern has changed gradually in recent years due to

1. A reduction in the maximum growth rates of the phytoplankters.
2. The introduction of variable stochiometry: types are now the basic variables of the model rather than species.

Consequently the appearance of the model output is now similar to the output pattern of a dynamic model, although the purpose of the model and the mathematical solution algorithm have not changed.

In fact the current version of the model even suffers from a structural underprediction in the final quarter of the year in many lakes, where energy is the main limitation during that period. This
problem will be investigated in the near future, together with several other subjects.

1.10 INPUTS TO THE MODEL

There are two kinds of inputs to the model: universal and lake-specific inputs. The universal inputs are the model coefficients. The initial values for these inputs were determined from

- Laboratory experiments. (Incubator Research Project).
- Literature sources.
- Observations in Grote Rug.
- Calibration.

The lake-specific inputs were directly or sometimes indirectly determined from measurements.

How all these data should be specified to the BLOOM II computer program is discussed in 'The standard input of BLOOM II' on page 51 and in 'The optional input of BLOOM II' on page 63.

1.10.1 Lake specific inputs

Contrary to the universal inputs, lake-specific inputs vary between lakes and years. These inputs are:

1. The average weekly water temperature.
2. Weekly concentrations of total available N, P and Si.
4. The background extinction Kb of the water.
5. The mixing depth Zmax.
6. The flushing rate constant or residence time.
7. Weekly zooplankton concentrations.

Already at a brief examination the reader will notice that frequently data for some of these variables will be missing for either some time-periods, or entirely. How significant that is to the model's performance, depends on which data are missing and how frequently. Obviously, we have to supply the model with reasonable values for the nutrient concentrations. But, for example, a run without zooplankton data is still valuable.

1.10.2 Universal inputs

Most inputs have already been discussed here, or are discussed in the BLOOM II report [Los, 1982]. Summarizing, BLOOM II requires the following universal inputs:

1. Minimum stochiometric coefficients (a_i,k) for the nutrients.
2. The remineralization rates (u_i) of the nutrients as a function of temperature.
3. The fraction of dead phytoplankton (q_i), which has to be remineralized.
4. The ratio of dryweight to chlorophyll of each type k.
5. The specific extinction coefficients (K_x) for each type k.
6. The coefficients to compute the respiration rate constant as a function of temperature.
7. The coefficients to compute the maximum production rate constant as a function of temperature.
8. The coefficients to compute the mortality rate constant as a function of temperature.
9. The efficiency curve E(I,T) of each type k.
10. The relative mixing depth RMIX_k of each type k to compute the impact of buoyancy regulation.
11. The disappearance rate of dead chlorophyll (v) as a function of temperature.
12. The relative preference of zooplankton for type k.
13. Several grazing coefficients (ZG, ZK, Xmin and PR_k).
14. The nutrient contents of zooplankton (h_i).
15. The average day length.

Values for some of the more important coefficients for phytoplankton species appear in Table 1.2 on page 31, and Table 1.3 on page 32.
Table 1.2

Dry weight - chlorophyll ratio, specific extinction coefficient and maximum growth rate constant $P_{\text{max}}(T)$ of phytoplankton types.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Diatom-E</td>
<td>75</td>
<td>7.50D-05</td>
<td>$0.50 \times 1.038$</td>
</tr>
<tr>
<td>Diatom-P/Si</td>
<td>100</td>
<td>6.50D-05</td>
<td>$0.42 \times 1.038$</td>
</tr>
<tr>
<td>Flagelate-E</td>
<td>75</td>
<td>7.50D-05</td>
<td>$0.33 \times 1.058$</td>
</tr>
<tr>
<td>Green-E</td>
<td>60</td>
<td>6.10D-05</td>
<td>$0.088 \times (T - 5.0)$</td>
</tr>
<tr>
<td>Green-N</td>
<td>80</td>
<td>6.10D-05</td>
<td>$0.053 \times (T - 10.0)$</td>
</tr>
<tr>
<td>Green-P</td>
<td>80</td>
<td>6.10D-05</td>
<td>$0.092 \times (T - 10.0)$</td>
</tr>
<tr>
<td>Aphanizomenon-E</td>
<td>75</td>
<td>1.50D-04</td>
<td>$0.18 \times 1.081$</td>
</tr>
<tr>
<td>Aphanizomenon-P</td>
<td>75</td>
<td>1.50D-04</td>
<td>$0.12 \times 1.095$</td>
</tr>
<tr>
<td>Microcystis-E</td>
<td>100</td>
<td>1.20D-04</td>
<td>$0.050 \times (T - 5.0)$</td>
</tr>
<tr>
<td>Microcystis-N</td>
<td>150</td>
<td>1.20D-04</td>
<td>$0.053 \times (T - 8.0)$</td>
</tr>
<tr>
<td>Microcystis-P</td>
<td>150</td>
<td>1.20D-04</td>
<td>$0.053 \times (T - 8.0)$</td>
</tr>
<tr>
<td>Oscillatoria-E</td>
<td>75</td>
<td>1.35D-04</td>
<td>$0.25 \times 1.062$</td>
</tr>
<tr>
<td>Oscillatoria-N</td>
<td>125</td>
<td>1.35D-04</td>
<td>$0.161 \times 1.077$</td>
</tr>
<tr>
<td>Oscillatoria-P</td>
<td>125</td>
<td>1.35D-04</td>
<td>$0.161 \times 1.077$</td>
</tr>
<tr>
<td>Type</td>
<td>Nitrogen</td>
<td>Phosphorus</td>
<td>Silicon</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>Diatom-E</td>
<td>0.050</td>
<td>0.0075</td>
<td>0.2200</td>
</tr>
<tr>
<td>Diatom-P/Si</td>
<td>0.050</td>
<td>0.0035</td>
<td>0.1500</td>
</tr>
<tr>
<td>Flagellate-E</td>
<td>0.105</td>
<td>0.0088</td>
<td>0.0007</td>
</tr>
<tr>
<td>Green-E</td>
<td>0.110</td>
<td>0.0088</td>
<td>0.0007</td>
</tr>
<tr>
<td>Green-N</td>
<td>0.059</td>
<td>0.0046</td>
<td>0.0007</td>
</tr>
<tr>
<td>Green-P</td>
<td>0.080</td>
<td>0.0046</td>
<td>0.0007</td>
</tr>
<tr>
<td>Aphanizomenon-E</td>
<td>0.088</td>
<td>0.0050</td>
<td>0.0007</td>
</tr>
<tr>
<td>Aphanizomenon-P</td>
<td>0.068</td>
<td>0.0030</td>
<td>0.0007</td>
</tr>
<tr>
<td>Microcystis-E</td>
<td>0.075</td>
<td>0.0100</td>
<td>0.0007</td>
</tr>
<tr>
<td>Microcystis-N</td>
<td>0.040</td>
<td>0.0080</td>
<td>0.0007</td>
</tr>
<tr>
<td>Microcystis-P</td>
<td>0.060</td>
<td>0.0065</td>
<td>0.0007</td>
</tr>
<tr>
<td>Oscillatoria-E</td>
<td>0.075</td>
<td>0.0075</td>
<td>0.0007</td>
</tr>
<tr>
<td>Oscillatoria-N</td>
<td>0.040</td>
<td>0.0060</td>
<td>0.0007</td>
</tr>
<tr>
<td>Oscillatoria-P</td>
<td>0.060</td>
<td>0.0035</td>
<td>0.0007</td>
</tr>
</tbody>
</table>
2.0 BLOOM II AND RELATED COMPUTER PROGRAMS

The BLOOM II model consists of several individual computer programs:

1. A pre-processing program called BLEFFPRO which integrates growth efficiency functions of phytoplankton species.
2. A pre-processing program called BLCOMBIN which re-orders the output of BLEFFPRO, or combines the results of several runs by BLEFFPRO into one input dataset for BLEFUNFO.
3. A pre-processing program called BLEFUNFO which writes the integrated efficiency dataset (output of BLEFFPRO or BLEFFFORM) unformatted to disk to save disk space and to reduce I/O time for the BLOOM program (See also BLEFFORM program).
4. A pre-processing program called BLEFFORM which writes the unformatted, integrated efficiency dataset (output of BLEFUNFO) formatted to disk. This way the efficiency file can be inspected and also used by BLCOMBIN.
5. The actual BLOOM II program, which reads the data, performs the computations and puts the results into tables and graphs.
6. A post-processing program called OXYMOD which computes the impacts of the phytoplankton blooms computed by BLOOM on the oxygen budget. This model was developed by J.G.C Smits of the Delft Hydraulics Laboratory [Los et al., 1982].
7. Post-processing programs using the output of BLOOM as input to the Statistical Analysis System (SAS) to compute certain statistics, or to produce graphical output.

This manual shall be confined to the pre-processing programs and the actual BLOOM program.
3.0 THE PRODUCTION EFFICIENCY PROGRAMS

3.1 PURPOSE

The production and hence growth rate of phytoplankton is determined by the light intensity among others. However, the latter varies in time and with the water depth. Thus to compute the growth rate in a specific time-interval, the model must take these variations into account. The most straightforward way to do this would be to compute the average efficiency of each phytoplankton species in each time-interval. This, however, proves to be very expensive in terms of computer time.

Fortunately Bigelow et al. [1977] demonstrated, that it is not necessary to re-integrate the growth efficiency in each time-step. Two important variations can be taken into account if the integration has been performed only once for a standard function under standard conditions:

1. A change in the number of daylight hours, but with the same standard intensity pattern.
2. A change in intensity at each instant by a constant fraction.

Any combination of the two can also be accommodated. Thus for all practical purposes we may use standard efficiency functions. The purpose of the four pre-processing programs: BLEFFPRO, BCOMBIN, BLEFUNFO and BLEFFFORM therefore is to generate, and manipulate the integrated efficiency functions and put them in a suitable format for the main BLOOM program.

Because operation of these programs is straightforward, they will be described much less elaborately than the main BLOOM program.

3.2 THE INTEGRATION PROGRAM BLEFFPRO

BLEFFPRO, which integrates the growth efficiency functions of the phytoplankton species, was developed by the Rand Corporation for the POLANO project [Bigelow et al., 1977]. The computational sections of the program have remained the same. The input was extended to except data in two rather than in only one form. For convenience the output is slightly modified.

3.2.1 Input of BLEFFPRO

The phytoplankton efficiencies can be entered in two different forms:

1. Tabulated.
2. As the three coefficients of the Eilers-Peeters production model of the Delta Department.
3.2.1.1 Tabulated input

Production curves are frequently published in graphical form obtained by some fit procedure through the measurement points. Many curves result from 'fitting by eye', but others are based upon formal mathematical fit procedures. In tabulated form each of these curves can be used as an input to BLEFFPRO. Figure 3.1 on page 36 and Figure 3.2 on page 37 show an example of an input dataset and the corresponding formats.

Line 1 specifies the number of functions (2 in the example) and the temperature, at which the functions have been measured.

Line 2 should contain the word 'TABLE' to indicate that the input is in the form of tabulated efficiency functions.

Line 3 indicates the number of tabulation points.

Lines 4 through 37 specify the actual efficiency data. The first ten columns of each line contain the light intensity in joules/m²/hr. The columns 11 though 20 contain the corresponding efficiency of species 1, columns 21 through 30 of species 2 etc., up to 10 different species.

Line 38 indicates the number of daily periods for which an intensity will be given.

Lines 39 through 61 specify the distribution of light intensities over a 24 hour period. This is usually done for an average day with 12 hours light and 12 hours darkness¹. Columns 1 through 11 contain the light intensity in Joules/m²/hr and columns 16 through 25 the corresponding time of the day when this intensity is encountered.

Lines 62 through 72 specify the relative growth rates of the phytoplankton species for day lengths between 7 and 17 hours. Columns 1 through 5 contain the day length in hours, the remaining columns contain the relative growth rates of the species.

¹ The implication is that BLOOM considers each day of the year to have the same intensity pattern, but as we have previously seen, the hourly intensity and day length may vary seasonally.
Figure 3.1 Example of tabulated input for BLEFFPRO. Two species of phytoplankton. The efficiencies are tabulated. The column line and the line numbers are not in the input file, but printed for convenience.

<table>
<thead>
<tr>
<th>2</th>
<th>20.0</th>
<th>1 = OSCILLATORIA, 2 = SCENEDESMUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00 D 04</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>2.03 D 04</td>
<td>0.88</td>
<td>0.74</td>
</tr>
<tr>
<td>4.05 D 04</td>
<td>0.97</td>
<td>0.84</td>
</tr>
<tr>
<td>8.11 D 04</td>
<td>1.00</td>
<td>0.93</td>
</tr>
<tr>
<td>1.22 D 05</td>
<td>1.00</td>
<td>0.98</td>
</tr>
<tr>
<td>1.62 D 05</td>
<td>0.95</td>
<td>1.00</td>
</tr>
<tr>
<td>2.03 D 05</td>
<td>0.80</td>
<td>1.00</td>
</tr>
<tr>
<td>2.43 D 05</td>
<td>0.65</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.70 D 06</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>1.75 D 06</td>
<td>0.00</td>
<td>0.0</td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000 D 00</td>
<td>6.9</td>
<td>39</td>
</tr>
<tr>
<td>1.381 D 05</td>
<td>7.5</td>
<td>40</td>
</tr>
<tr>
<td>2.637 D 05</td>
<td>8.0</td>
<td>41</td>
</tr>
<tr>
<td>3.767 D 05</td>
<td>8.5</td>
<td>42</td>
</tr>
<tr>
<td>4.771 D 05</td>
<td>9.0</td>
<td>43</td>
</tr>
<tr>
<td>6.025 D 05</td>
<td>9.5</td>
<td>44</td>
</tr>
<tr>
<td>7.030 D 05</td>
<td>10.0</td>
<td>45</td>
</tr>
<tr>
<td>7.660 D 05</td>
<td>10.5</td>
<td>46</td>
</tr>
<tr>
<td>8.285 D 05</td>
<td>11.0</td>
<td>47</td>
</tr>
<tr>
<td>8.535 D 05</td>
<td>11.5</td>
<td>48</td>
</tr>
<tr>
<td>8.665 D 05</td>
<td>12.0</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.020 D 04</td>
<td>17.5</td>
<td>60</td>
</tr>
<tr>
<td>0.000 D 00</td>
<td>17.8</td>
<td>61</td>
</tr>
<tr>
<td>7.</td>
<td>0.49</td>
<td>0.58</td>
</tr>
<tr>
<td>8.</td>
<td>0.55</td>
<td>0.65</td>
</tr>
<tr>
<td>9.</td>
<td>0.62</td>
<td>0.70</td>
</tr>
<tr>
<td>10.</td>
<td>0.68</td>
<td>0.76</td>
</tr>
<tr>
<td>11.</td>
<td>0.75</td>
<td>0.81</td>
</tr>
<tr>
<td>12.</td>
<td>0.82</td>
<td>0.86</td>
</tr>
<tr>
<td>13.</td>
<td>0.87</td>
<td>0.90</td>
</tr>
<tr>
<td>14.</td>
<td>0.91</td>
<td>0.93</td>
</tr>
<tr>
<td>15.</td>
<td>0.96</td>
<td>0.97</td>
</tr>
<tr>
<td>16.</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>17.</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

The production efficiency programs
Figure 3.2  Formats for tabulated input of BLEFFPRO.

<table>
<thead>
<tr>
<th>Line</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I5,10X,F5.2</td>
</tr>
<tr>
<td>2</td>
<td>A8</td>
</tr>
<tr>
<td>3</td>
<td>I5</td>
</tr>
<tr>
<td>4-37</td>
<td>D10.0,10F10.0</td>
</tr>
<tr>
<td>38</td>
<td>D11.4,4X,F10.0</td>
</tr>
<tr>
<td>39-61</td>
<td></td>
</tr>
<tr>
<td>62-72</td>
<td>11F5.0</td>
</tr>
</tbody>
</table>

3.2.1.2 Input Eilers-Peeters model

Regular measurements of primary production for example by the Environmental Division of the Delta Department and by the Microbiological Department of the Amsterdam University during the incubator research project have provided huge amounts of production curves for various species under various conditions. Two members of the Delta department, Paul Eilers and Kees Peeters, have developed a model to describe production data. They postulate that each set of production data can be described by a curve determined by three parameters: A, B, and C, according to the following equation:

\[
P = \frac{I}{AI^2 + BI + C}
\]  \hspace{1cm} (3.1)

in which

- \( P \) is the production rate constant in mg O2 per mg chlorophyll per hr,
- \( I \) is the light intensity in Watt per m²,
- A, B, and C are the specific coefficients of the model.

These are estimated by a least squares fit technique. Because the efficiency program BLEFFPRO uses a standardized efficiency curve, Eq. (3.1) must, according to Eq. (1.18), be divided by \( P_{\text{max}}(T) \). This is done in BLEFFPRO.

Figure 3.3 on page 39 and Figure 3.4 on page 40 show an example of an input dataset and the corresponding formats.

**Line 1** specifies the number of functions (7 in the example) and the temperature, at which the functions have been measured.

**Line 2** contains the word 'NOTABLE' to indicate that the input is not tabulated. The table of efficiency data will be generated from the Eilers-Peeters model.

**Line 3** indicates the number of tabulation points.

**Lines 4 through 37** contain the light intensity in joules/m²/hr for which the efficiency will be computed.
Lines 38 through 44 contain the coefficients for the Eilers-Peeters model for the 7 species of the example. Columns 1 through 10 contain values for A, columns 11 through 20 for B and columns 21 through 30 for C. There is one input line for each species.

Line 45 indicates the number of daily periods for which an intensity will be given.

Lines 46 through 68 specify the distribution of light intensities over a 24 hour period. This is usually done for an average day with 12 hours light and 12 hours darkness. Columns 1 through 11 contain the light intensity in Joules/m²/hr and columns 15 through 25 the corresponding time of the day when this intensity is encountered.

Lines 69 through 79 specify the relative growth rates of the phytoplankton species for day lengths between 7 and 17 hours. Columns 1 through 5 contain the day length in hours, the remaining columns contain the relative growth rates of the species.
Figure 3.3  Example of Eilers-Peeters input for BLEFFPRO. Seven species of phytoplankton. The efficiencies are represented by the Eilers and Peeters model. The column line and the line numbers are not in the input file, but printed for convenience.

```
+---+---+---+---+---+---+---+---+
  7  15.0
NOTABLE
  34
  0.00 D  04
  2.03 D  04
  4.05 D  04
  8.11 D  04

  1.70 D  06
  1.75 D  06
```

```
+---+---+---+---+---+---+---+---+
  1.70 D  06  15.21730
  0.001995  0.148408
  0.000926  0.130173
  0.000926  0.130173
  0.002337  0.318847
  0.002594  0.121344
  0.001037  0.312904
  0.003828  0.297496

  23
  0.000 D  00  6.9
  1.381 D  05  7.5
  2.637 D  05  8.0
  3.767 D  05  8.5
  4.771 D  05  9.0
  6.025 D  05  9.5
  7.030 D  05 10.0
  7.660 D  05 10.5
  8.285 D  05 11.0
  8.535 D  05 11.5

  5.020 D  04  17.5
  0.000 D  00  17.8
  7 0.67 0.58 0.49 0.59 0.71 0.58

  15 0.98 0.99 0.96 0.97 0.97 0.97
  16 1.00 1.00 1.00 1.00 1.00 1.00
  17 1.00 1.00 1.00 1.00 1.00 1.00
```

The production efficiency programs
Figure 3.4 Formats for Eilers-Peeters input of BLEFFPRO. The example is for 7 species.

<table>
<thead>
<tr>
<th>Line</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I5,10X,F5.2</td>
</tr>
<tr>
<td>2</td>
<td>A8</td>
</tr>
<tr>
<td>3</td>
<td>I5</td>
</tr>
<tr>
<td>4-37</td>
<td>D10.0</td>
</tr>
<tr>
<td>38-44</td>
<td>3F10.6</td>
</tr>
<tr>
<td>45</td>
<td>I5</td>
</tr>
<tr>
<td>46-68</td>
<td>D11.4,4X,F10.0</td>
</tr>
<tr>
<td>69-79</td>
<td>11F5.0</td>
</tr>
</tbody>
</table>

3.2.2 Output of BLEFFPRO

The output of BLEFFPRO consists of:

1. A line containing the number of points and the temperature of the functions.
   The format is: I5,10X,F5.2.
2. A vector containing the diurnal intensity distribution (the Z-vector).
   The format is: 10(D15.8,3X).
3. A line containing the number of points in the following vectors.
   The format is: I5.
4. N vectors containing the integrated production efficiencies of the n species (the FUN vector).
   The format is: 10(D15.8,3X).
5. N vectors containing the derivatives of the integrated production efficiencies of the n species (the DER vector).
   The format is: 10(D15.8,3X).
   The way it is set up the program first writes the first line of FUN for all species, followed by the first line of DER. Next comes the second line of FUN, followed by the second line of DER etc.
6. If the coefficients of the Eilers Peeters model were used as input, the program also writes the light intensities and the computed, tabulated efficiencies of each species into a second output file. This enables a simple comparison between various input datasets, or a combination into one new input dataset for another run.
   The format is: E10.3,10F10.2.

3.3 THE COMBINATION PROGRAM BLCOMBIN

Often the integrated efficiency functions, which are computed by BLEFFPRO, are incomplete or in the wrong order. For example
BLEFFPRO may have been applied to generate a new function for only one phytoplankton species, which should be used in combination with previously computed functions. Some species have more or less the same production functions, hence we may want to use one and the same integrated efficiency function as input to the model for different species.

For these purposes the program BLCOMBIN was written. This program reads two datasets with integrated efficiency functions, combines them as prescribed by the user, and writes the result into a new disk file. Obviously we can replace the efficiency data of a particular species by new data. If, however, we use two identical copies of the same efficiency datasets as input to BLCOMBIN, we can also re-arrange the order of the data in a single file or duplicate the data of a species in more than one position. Hence by making one or several successive runs with BLCOMBIN we can put any set of integrated efficiency data into any position.

The program checks for certain inconsistencies between the two efficiency input datasets and will abend if the number of data points, the temperatures, or the Z-vectors are different.

### 3.3.1 Input of BLCOMBIN

BLCOMBIN reads three different files: a control file, and the two efficiency files. The latter have been written by the BLEFFPRO program and are read here in the same format as shown in 'Output of BLEFFPRO' on page 40. The control file consists of only three lines of input:

- **Line 1** contains the number of efficiency functions (columns) in the first, respectively second input file.
  - The format is 2I5.
- **Line 2** contains information where to put each efficiency function (column) of the first file in the resulting output file. If the number $i$ is equal to $j$, this means that efficiency function $i$ of file 1 should be written in position $j$ of the resulting file. A zero number indicates that a function is deleted.
  - The format is 10I5.
- **Line 3** contains information where to put each efficiency function (column) of the second file in the resulting output file. If the number $i$ is equal to $j$, this means that efficiency function $i$ of file 2 should be written in position $j$ of the resulting file. A zero number indicates that a function is deleted.
  - The format is 10I5.

**Figure 3.5 Example of input file for BLCOMBIN.**

```
......+......1......+......2......+......3......+......4......+......5......+......6
  5    7
  0   2   5   6   7
  3   4   0   8   0   1   0
......+......1......+......2......+......3......+......4......+......5......+......6
```

In the example we combine a file with efficiency data for 5 species with a file for 7 species to create a new file for 8 species. Species 6 of file 2 will be the first in the output file, species 2 of file 1 comes in second place, species 1 of file 2 in third place, etc. Data of four species are ignored.

The production efficiency programs 41
The program does not check for inconsistencies in this input file such as duplicate numbers. The maximum number of species in the output file equals 10.

3.3.2 Output of BLCOMBIN

The output of BLCOMBIN is completely similar to the output of the integration program BLEFFPRO.

3.4 THE TRANSFORMATION PROGRAM BLEFUNFO

To save disk space and to reduce the CPU time required by BLOOM to read the (large) efficiency files, they are transformed from formatted to unformatted by the program BLEFUNFO. This program simply reads an entire efficiency file which has been created by BLEFFPRO and BLCOMBIN and writes it unformatted into a new disk file, which is an input for BLOOM.

3.5 THE TRANSFORMATION PROGRAM BLEFFORM

The unformatted output of BLEFUNFO can only be read by computer programs. To visualize the data in a particular file, or to create an input file for BLCOMBIN, it should be possible to (re)format this output. To this purpose the program BLEFFORM was written.
4.0 BLOOM II COMPUTER PROGRAM

4.1 GENERAL DESCRIPTION OF THE BLOOM II PROGRAM

Once the integrated efficiency functions of the phytoplankton species have been calculated and perhaps re-arranged, all further computations are performed by the main BLOOM program. Currently the program consists of more than 60 subroutines and uses 8 special functions in addition to the normal FORTRAN functions. It can be divided into four sections for:

1. Standard input,
2. Optional input,
3. Computations,
4. Output.

The standard input section reads lake-specific data, and model coefficients (universal inputs), including the integrated efficiency datasets.

The optional input section can read additional information to change nominal coefficient values, or to invoke certain options of the program. This section of the program is written conversationally: the user can retrieve information from the model, respond in a certain way, receive answers, warning messages, help etc. Although primarily intended to be used in an interactive job, almost every option can also be used in a batch job.

The computational section of the program performs the actual calculations as prescribed in the preceding two input sections.

The output section puts the results into several tables and (optionally) graphs. The minimum output consists of only two summary tables, but at request more than 5000 lines of detailed information can be produced.

4.2 INTERACTIVE AND BATCH PROCESSING

Computer programs can be divided into two general categories: batch and interactive programs. As a rule the latter are more 'user-friendly' than the first. In batch jobs there is no communication between the user and the computer program. The user submits a job, which reads input data from disk (or tape or punch cards), performs the computations and sends the output to a printer (or disk or tape). If the user decides to make a new computation he must adjust some of the input data, re-submit the job and once again wait until the computations are completed. Most computer programs still operate this way, even though the input of a deck of punch cards has often been replaced by a set of instructions entered from a display terminal.

Interactive programs allow at least some communication between the user and the program. In their infant stages interactive programs simply read some information from a terminal and send (part of) the output back to it. However, these programs can be made much more intelligent and effectively guide the (unexperienced) user. For example rather than just indicating that some number should be entered, the program could ask:
Enter the mineralization rate constant of NITROGEN per day:

Checks can be incorporated to signal certain errors and warn the user that he has entered a peculiar coefficient value, or even force him to enter a value within the range expected by the program.

To make them even more user-friendly, interactive programs can be extended with a tutorial (HELP) section to provide on-line information about program options (their names; how they work; their status: on/off), and about coefficients (their current values; how to enter new values).

For an interactive program, however, the burden of using strictly formatted input data as in FORTRAN, should be relieved to allow (some form of) free formatted input\(^1\). As a matter of fact this kind of input is a major improvement not only for interactive, but also for batch programs.

Whether a program should be written as an interactive, or as a batch program, is determined by many factors such as:

- The nature of the problem.
- The number of different pathways.
- The number of different program users and their experience.
- The required amount of computer time on the available computer.
- The available computer operating system.

Algae Bloom and the first versions of the BLOOM II computer program could only be processed in batch mode. However, the BLOOM II model leans itself extremely well for interactive processing because (1) it is basically a steady state model which means that computations can be started or stopped in any time-period, (2) there are many model coefficients and program options, and (3) the program runs fast enough to complete a whole year computation within a minute on any medium or large seized computer system. Thus the optional input section of the program was gradually rewritten into its current, highly interactive version with the following characteristics:

1. Great freedom of input formats: data can be entered on one line or extended over several lines. Also each type of variable (integer, real, and character string) can start in any column as long as there is at least one delimiting blank.
2. Correction possibilities for many (probable) errors (invalid instructions, suspected or erroneous coefficient values). In batch mode the same checks are performed, but in this case severe errors are fatal and cause termination of the program.
3. Substring matching of program instructions. This means that any program instruction can be abbreviated to a minimum number of characters. If, however, the abbreviation is ambiguous because it could refer to more than one program instruction, an error message is displayed to use a longer (=unique) abbreviation.
4. Logical way of prompting following each user instruction. The amount and contents of the messages depends on the user's responses (valid, invalid, expected continuation).

\(^1\) In FORTRAN all data are read (and written) according to certain strict rules related to (1) the type of variable: whole number (integer), decimal number (real), or character string, (2) its value, (3) its size, and (4) its position on an input line (the starting column for example).
5. Extensive help facilities for program options and coefficient values.
6. The ability to execute a great number of system commands within the program. For example tables with coefficient values can be written to disk by the program, then edited with the full-screen system editor, saved and then read back by the program. Also any existing file can be displayed, including the output files produced during the current session of the program.

4.3 SYSTEM DEPENDENT PROGRAM FEATURES

Most computer systems offer at least some special, system dependent features, which could extend the capabilities of a program. However, incorporation of these features makes it difficult if not impossible to implement that program on a computer system of another manufacturer or with another operating system. Therefore a programmer should carefully decide:

1. Whether he wants to use system dependent features,
2. Which features he is going to use,
3. And how he incorporates them in his program.

The current BLOOM II program makes use of a number of system dependent features to

- Perform free formatted input, and character string matching and manipulation,
- Execute system commands (time functions; display control; modify coefficient values with the editor).

These features are heavily used by the program. Still it should not be hard to implement the program on another computer system. First all system dependent features are build into separate subroutines so they are not mixed throughout the program code. Second most features can be replaced easily by others that perform (almost) exactly the same function. Third alternative (but less fancy) functions are already build into the program for those features, which are probably unique for computers with the VM/370 operating system.

4.3.1 Special interactive functions

To improve interactive programming in FORTRAN, Norman Katz has written a number of functions in IBM assembler language, which are the basis for the conversational section of the BLOOM II program. The procedure is as follows. First each line of input is read by the program as if it consisted of character strings (10A8 format). Second a special function breaks this line into elements called 'tokens' (numbers and words), which are separated by at least one blank. The position of a token on an input line is unimportant. Third several special functions convert each token into a form which can be understood by FORTRAN. For instance a token which should be treated as a whole number is converted from a character
into a whole number. After this procedure each token can be used properly by the program.

Other functions are used in the program for (partial) string matching, and to execute system commands, among others.

Except the function to perform system commands, which is of course unique for each computer system, there is a FORTRAN and/or PL/I version available of each of these special functions. So the program can still be implemented on other computer systems without loss of its conversational section. The current program makes use of the Assembler versions of the routines because these are more efficient in terms of computer time.

4.3.2 System commands

Under the VM/370 operating system it is possible to execute a large number of system commands from within a FORTRAN program. For compatibility with other operating systems this powerful option is, however, only used in a few well defined cases:

1. To retrieve the current date and the date when the most recent update of the program was installed.
2. To perform special display functions: clear the screen at the beginning of an output section, hold a particular screen image.
3. To modify tables of coefficient values with the full-screen system editor.

Date functions are available on many computer systems. Display functions improve user friendliness, but are not of vital importance. Use of the editor is not the only way to change coefficient values, because the value of each coefficient can also be altered by a special command language, incorporated in the BLOOM program. Conclusion: the program could rather easily be implemented on computers with other operating systems.

4.4 JOB CONTROL FOR BLOOM II

4.4.1 Standardized procedures

Job control can be defined as the set of instructions which is necessary to execute a computer program. The two essential steps are

1. Specification of the input and output files.
2. Loading of the program into storage and execution of the program.

The form of these instructions depends on the operating system of the computer. Basic job control under VM/370 is very simple. Moreover, this system stimulates the user to combine several tasks in a procedure program, called an 'EXEC'. A simple exec may be nothing more but a sequence of several commands, but it is rather easy to develop intelligent, highly inter-
active execs with various branches. For each of the efficiency programs and for the actual BLOOM program we have written an exec bearing the name of the program, for example BLEFFPRO or BLOOM. Once the program name has been typed, the exec will question the user for the minimum amount of information it needs to set up a job.

The main exec (BLOOM) questions the user by means of a 'MENU' in which the questions and the user's answers from the previous job are displayed on a fresh terminal screen. The BLOOM exec can also provide additional help and show the user a list of all data files which, according to their name and organization, could be used as input to the program. Thus as long as the user is satisfied with the logic and naming conventions build into the execs, he never has to worry about any form of job control.

The BLOOM exec starts an interactive job of the program. An exec called 'BATCHBLM' can be used for a batch job of the model.

4.4.2 File definitions for BLOOM II

As we have explained in the previous section, execs can set up a standard job for each of the bloom computer programs. However, at some stage the user may want to modify this set-up, for example because he wants to use different naming conventions than those expected by the execs. Or he may want to implement the program on a different computer system with different job control. Therefore we shall now describe the input and output files of BLOOM II.

Under VM/370 names and organization of files are specified by the 'FILEDEF' system command. As an example we have shown the file definitions for BLOOM II in the following figure.

---

2 Using menus is only possible on full-screen terminals. If a type-writer type terminal is used, the BLOOM exec will automatically change its mode of operation and question the user differently.
Figure 4.1  File definitions for BLOOM II. Interactive job. Some filenames are fictive.

<table>
<thead>
<tr>
<th>I Unit number</th>
<th>Devi ce</th>
<th>FN</th>
<th>FT</th>
<th>FM</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>FI</td>
<td>FT05F001</td>
<td>DISK</td>
<td>GR77</td>
<td>DATA05VS A</td>
<td>(LRECL 80 BLOCK 800 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT06F001</td>
<td>TERM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FI</td>
<td>FT09F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>OUT9 A</td>
<td>(LRECL 132 BLOCK 132 RECFM F</td>
</tr>
<tr>
<td>FI</td>
<td>FT10F001</td>
<td>TERM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FI</td>
<td>FT11F001</td>
<td>DISK</td>
<td>GR772</td>
<td>DATA11 A</td>
<td>(LRECL 80 BLOCK 800 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT12F001</td>
<td>DISK</td>
<td>EFFUNF</td>
<td>M078 A4</td>
<td>(LRECL 1020 BLOCK 1024 RECFM VS</td>
</tr>
<tr>
<td>FI</td>
<td>FT14F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>OUT14 A</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT15F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>OUT15 A</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT16F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>OUT16 A</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT17F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>OUT17 A</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT18F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>OUT18 A</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT19F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>OUT19 A</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT21F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>OUT21 A</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT30F001</td>
<td>TERM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FI</td>
<td>FT31F001</td>
<td>DISK</td>
<td>DUMMY</td>
<td>DATA</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT41F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>DUMMY1 A</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT42F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>DUMMY2 A</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT43F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>CFT14 A</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
<tr>
<td>FI</td>
<td>FT45F001</td>
<td>DISK</td>
<td>GRTEST</td>
<td>PLOT</td>
<td>(LRECL 132 BLOCK 132 RECFM FB</td>
</tr>
</tbody>
</table>

In Figure 4.1

FI FTnnF001 is the abbreviation of the system command FILEDEF. is the name of the FORTRAN file. Thus if nn=21 there is a read or write statement to unit 21 in the program, for example

WRITE (21,100) VAR

would write the value of variable VAR to a disk file called 'GRTEST OUT21 A'.

Device indicates to the program where it can find a particular file. Most files are read from or written to DISK. In an interactive job, however, some input is read from the user's terminal and part of the output is directed towards it. In principle each DISK file could also have the device TERM or visa versa, but it is not practical, of course, to direct large output files to the terminal let alone to read large input files from the terminal. Also notice that there is only one terminal, so if more output files are written to the terminal they may be intermixed.

FN, FT, FM are respectively filename, filetype and filemode of disk files. Name and type can both be eight or less characters, the filemode is one or two characters long. Together they form the name of a file in the VM/370 system, so we have for example a file called 'GR77 DATA05VS A' and one called 'GRTEST OUT21 A'. Input files, of course, should always exist. Output files, however, do not have to exist because they can
be created by the job. Normally output files automatically override any existing files with the same name, but in the BLOOM exec the user is warned at the start of the job so that he can still cancel it. Terminal files have no specific name.

Organization indicates the characteristics of a file. Most input files are fixed (RECFM FB) with a logical record length of 80 (LRECL 80) and a blocking size of 800 (BLOCK 800). The output files are fixed (RECFM F) with a logical record length of 132 (LRECL 132) and a blocking size of 132 (BLOCK 132). The input file with the integrated efficiency curves in the example called 'EFFUNF M078 A4' has a variable format (RECFM VS) and therefore a different blocking size and record length. Creation of this file has been explained in Sec. 'The transformation program BLE-FUNFO' on page 42.

As the BLOOM and BATCHBLM execs are currently set-up, the user specifies

1. The file name of the two input files with filetype DATA05VS and DATA11. In the example this name is GR772.
2. The filetype of the efficiency file with filename EFFUNFOR³. In the example this name is M078.
3. The file name of the main output files with filetypes OUTnn where nn has the value of the corresponding file number FTnnF001. In the example this name is GRTESIS.
4. The version of the program which must be loaded.

4.4.3 Creating a terminal log file

As we have explained in the previous section, a part of the output (and also of the input) of BLOOM II appears on the terminal. Without special measures this information would be lost so that it would be very difficult to reproduce an interactive program session. Fortunately the VM/370 operating system offers a simple possibility to save both the terminal inputs and outputs. If at a certain moment the user enters the instruction

spool console to # start

all terminal information will be kept until the instruction

spool console stop close

is entered⁴. The user now receives a message that a 'RDR' (reader) file has been created. This file can be handled further following the system instruction 'FULRDR'. Under control of this program the user can (1) BROWSE, or (2) PURGE (erase), or (3) PRINT the reader file, or READ it to create a normal disk file.

---

³ In Figure 4.1 on page 48 we have abbreviated this name to EFFUNF, but EFFUNFOR is the filename actually required by the current execs.
⁴ If the console file is not closed explicitly, it is automatically closed when the user logs off.

BLOOM II computer program 49
The BLOOM procedure to execute the BLOOM program automatically generates a console file with terminal input and output of a program session. In other words after a session the user can immediately type FULRDR to process this log file.

Note: this procedure is system dependent. If the program would be implemented on a computer system where it would be impossible to save terminal information, we would have to adjust the program and for example duplicate all terminal input and output statements to save them in an ordinary disk file.
5.3 INPUT FILE FT12F001

The third file is a variable format file containing the integrated growth efficiency curves of the phytoplankton species. How this file was created has already been explained (See 'The production efficiency programs' on page 34). Notice that the order by which the integrated efficiency curves are read should be consistent with the order of species names and species specific coefficients read from input file FT05F001.
6.0 THE OPTIONAL INPUT OF BLOOM II

When input file FT11F001 has been read (See 'The standard input of BLOOM II' on page 51), the program is put into the 'Option command mode' and resumes reading from input file FT05F001 until

1. It reads the program instruction 'INTERACT' (or any valid abbreviation). The interactive mode is switched on. No further lines are read from input file FT05F001. The rest of the input must be entered at a terminal.

2. It reads the program instruction 'RUN' followed by either 'CONTINUE' or 'STOP'. The program stops reading and starts computing. When it is finished, it starts to read new instructions from file FT05F001 if 'CONTINUE' was specified.

3. The program detects a severe error when it is operating in batch mode. It stops reading, prints an error message and terminates abnormally.

6.1 DIFFERENCES BETWEEN BATCH AND INTERACTIVE MODE

If the instruction 'INTERACT' is found in file FT05F001, the program mode is switched from batch to interactive. This instruction can be given anywhere in section 6 of file FT05F001. This file will now be closed and from now on the program reads from file FT30F001 which is associated to the terminal (See Figure 4.1 on page 48). Unlike the batch mode, the interactive mode is irreversible: once the program is operating interactively, it will not allow the user to return to batch processing.

Almost every program instruction can be entered regardless whether the program operates in interactive or batch mode. Moreover inputs are essentially free formatted in both cases (See 'Interactive and batch processing' on page 43). But as will be shown in Figure 6.1 on page 64 there are some differences too.
Figure 6.1 Differences between batch and interactive mode.

<table>
<thead>
<tr>
<th></th>
<th>Batch</th>
<th>Interactive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input messages</td>
<td>No intermediate prompting messages.</td>
<td>Intermediate prompting messages displayed.</td>
</tr>
<tr>
<td>(optional)</td>
<td>Red from disk file FT05F001.</td>
<td>Red from terminal.</td>
</tr>
<tr>
<td>Output</td>
<td>Directed to disk files.</td>
<td>Some output directed to terminal in compressed format. Same output as in batch run directed to disk files.</td>
</tr>
<tr>
<td>System commands</td>
<td>Usage limited to commands not requiring display terminal. Usage editor prohibited.</td>
<td>Usage limited by operating system only. Usage editor supported.</td>
</tr>
</tbody>
</table>

Unless stated otherwise, it will be assumed that the program operates in interactive mode in the rest of this manual.

6.2 HOW INSTRUCTIONS SHOULD BE ENTERED

Instructions are invoked by entering their full name or any valid abbreviation. If the program expects more information from the user for example a second instruction, or a coefficient value, it prompts him if it operates in interactive mode. However, most prompting messages are suppressed when an experienced user enters an entire sequence of instructions all at once.

Suppose that from now on the user wants to decrease the available phosphorus concentration to 0.75 times the nominal value. The following sample session shows how an unexperienced user could instruct the program to do this:
Begin Sample Session

Parameter cmd: Nutrc
Nutrient name: ?
? Is an invalid name for a nutrient
Valid nutrients are: NITROGEN PHOSPHOR SILICON
Nutrient name: P
Mult. factor: 0.75
Add. factor: 0.0
New concentration of PHOSPHOR is 0.75 * nominal conc. + 0.00
Parameter cmd:

End Sample Session

An experienced user would have entered this instruction much more compactly:

Begin Sample Session

nutrc p 0.75 0
New concentration of PHOSPHOR is 0.75 * nominal conc. + 0.00
Parameter cmd:

End Sample Session

In the latter example all information is entered on one line, the commands are abbreviated to the smallest valid number of characters and the decimal point following the number '0' is omitted. Notice that the program has automatically suppressed all intermediate messages. Only the final message is displayed which shows how the phosphorus concentration will be computed.

In case the user really does not know how to continue or when he gets stuck, he can always enter a question mark '?'. In most parts of the program a question mark is understood as a call for help, which is then provided. But even when a question mark is not recognized it will still be treated as an invalid response, which is usually followed by a message to indicate the kind of response expected by the program. This usage of the question mark was already illustrated in one of the two previous sample sessions.
6.3 CONTROL OF MODEL OPTIONS

As remarked at the beginning of this chapter, the program is automatically put into 'Option command mode' when all the standard input information has been read. In this mode the user has control over a large number of program options. He can for example request a printplot of certain output variables, or add growth constraints to the normal nutrient energy constraints (see 'Structure of the phytoplankton model BLOOM II' on page 1).

Each option defaults to a certain state in the program, usually this is 'off': it has no effect on the computations. At request an option is turned 'on': it now becomes active and affects the results of the model. Options can also be in the state 'Reset': the default state is restored, but the program remembers that this state had been modified previously. Hence the user can distinguish between options which were automatically turned off by the program and options reset by himself.

The user is notified that the program is in option command mode by the message:

Option cmd:

which is displayed on his terminal. In a batch run this message is suppressed. Each time he enters the name of a program option, it is turned 'on', a message is displayed (or printed). The program re-displays 'Option cmd:' to indicate that it awaits further instructions.

Some options require that at least one other option is also active. If the user request any of these options, the program automatically checks whether all prerequisite options are active and puts them 'on' if necessary. The user is notified by a message.

A listing of all currently active options is displayed (or printed in a batch run) following the command 'WHICH?'.

If the user enters 'RESET' the program is put into a special state in which options can only be reset. The user is notified by the message:

Reset Option cmd:

Each time the user enters the name of an option, the program responds with a message indicating it has been reset to its default value. Some options are automatically reset by the program and cannot be reset by the user. A listing of all options which were reset is displayed (or printed in a batch run) following the command 'WHICH?'. To return to the primary 'Option cmd:' status the user must enter 'END'.

If the user enters 'HELP' the program is put into a special state in which (only) help information is available about program options. The user is notified by the message:

Help Option cmd:

Each time the user enters the name of an option, the program responds with a short description of that option. To return to the primary 'Option cmd:' status the user must enter 'END'.
The option command mode can be terminated by several commands:

1. 'PARAM' puts the program into the 'Parameter command mode', in which the user can control coefficient values rather than program options (See 'Control of model coefficients').
2. 'RUN' followed by either 'CONTINUE' or 'STOP' instructs the program to start its computational and output section. Afterwards the program either stops, or awaits new instructions.
3. 'STOP' (not preceded by 'RUN') instructs the program to terminate immediately without performing any further calculations.

The (current) tutorial for options commands is shown in 'Appendix 1: Help for program options' on page 105. The following sample session is a short illustration.

```
Begin Sample Session

All options have been (re)set to default values
Option INTERACT has been put on
Option cmd:
help

.
.
.

Help Option cmd:
du
Option DUMP writes a detailed output to unit 6
Containing information for all extinction intervals,
for each iteration and time-step.

WARNING DUMP writes many output lines and should ONLY be
used in an interactive run for a limited number of periods
/about 5 or less).
Help Option cmd:
e
Option cmd:
diu
Option DIURNAL has been put on
Option PRODUC has been put on
Option cmd:

End Sample Session
```

Notice that activating option 'DIURNAL' has automatically activated option 'PRODUC' as well.

6.4 CONTROL OF MODEL COEFFICIENTS

The optional input of BLOOM II
### 6.4.1 General coefficients

In 'Control of model options' on page 66 we have described how the user can control program options. In addition the user may also want to change some coefficient values, for example in calibration or sensitivity computations with the model. To do so he should enter the instruction 'PARAM' which puts the program into 'Parameter command mode'. The user is notified by the message:

**Parameter cmd:**

which is displayed on his terminal. In a batch run this message is suppressed.

A short description of the 'Parameter command mode' including a list of valid coefficient names is displayed (or printed) following the instruction '?'. The user who does not know how to continue will be prompted by the program. A list of all current coefficient settings is displayed (or printed) following the 'PRINT' command. To return to the option command mode the user should enter 'OPTION'.

A parameter value can be modified according to one of the following equations:

\[
\begin{align*}
P &= A \\
P &= A \times \text{Input} + B \\
P &= F(A,B)
\end{align*}
\]

(6.1)

where

- \(P\) is the coefficient value to be modified,
- \(A\) and \(B\) are numbers which must be entered by the user,
- Input is the array of coefficient values which was read by the program from one of the standard input files (See 'The standard input of BLOOM II' on page 51),
- \(F\) is some function (linear; exponential).

The modification procedure is always initiated by the user typing the coefficient name, such as 'DEPTH', 'SOLARRAD', or 'MORTAL'.

If a parameter \(P\) can be computed according to several functions \(F\), the type of function should be specified as the second instruction. Finally the values of \(A\) and sometimes \(B\) should be entered to set or compute the new coefficient value of \(P\). A relatively complex example has already been shown in 'How instructions should be entered' on page 64. Another illustration is given in the next example.
Begin Sample Session

Parameter cmd: pr
Diepte 1.35M; achtergrond extinktie 2.5 per m;
Refraktair N = 0.075. Tentatieve berekening.
Temperature calculated as 1.00 * nominal Temp + 0.00
Solar radiation calculated as 1.00 * nominal rad + 0.00
Backgr. ext. calculated as 1.00 * nominal Kb + 0.00
Depth calculated as 1.00 * nominal depth + 0.00
Concentration of NITROGEN is 1.00 * nominal conc. + -0.75
Concentration of PHOSPHOR is 1.00 * nominal conc. + 0.00
Concentration of SILICON is 1.00 * nominal conc. + 0.00
Mineralization of NITROGEN = 0.0060 * Temperature
Mineralization of PHOSPHOR = 0.0060 * Temperature
Mineralization of SILICON = 0.0250
Blooms will be calculated for the following 1 periods:
First period: 1 Last period: 52 Increment: 1
Mortality rate calculated as exponential function of temp.
Sedimentation rate = 0.000
Flushing rate = 0.000
Mineralization rate of detritus = 0.006 * Temperature
Mineralization rate of chlorophyll is:
  EXP ( 0.0296 * Temperature - 1.8971 )
Nutrient fraction becoming detritus = 0.500
Parameter cmd: temp
Mult. factor:
1
Add. factor:
3
New temperature calculated as 1.00 * nominal Temp + 3.00
Parameter cmd:

End Sample Session

In most cases the programs checks for obvious errors in the input numbers. A description of all parameter commands is shown in 'Appendix 2: Help for model coefficients' on page 108.
Separate modes are available to modify coefficients of (1) phytoplankton stoichiometry, (2) phytoplankton growth coefficients or (3) zooplankton. These three modes are described in the following sections. Note: sections 'Phytoplankton species stoichiometric coefficients' and 'Phytoplankton species growth coefficients' on page 72 are more or less identical.

6.4.2 Phytoplankton species stoichiometric coefficients

As we have explained earlier, the actual variables of the model are phytoplankton types rather than species. Therefore the species stoichiometry consists of the coefficients all its types. The initial stoichiometry of the phytoplankton types is read from unit FT05F001 (See 'The standard input of BLOOM II' on page 51). To
change any of these coefficients for one or several types, the user should first activate the parameter command mode and then enter the instruction 'STOCHIOMETRY' to put the program into 'Stochiometry command mode'. The user is notified by the message:

Stochiometry cmd:

which is displayed on his terminal. In a batch run this message is suppressed.

A short description of the 'Stochiometry command mode' including a list of valid coefficient names is displayed (or printed) following the instruction '?'. The text is shown in 'Appendix 3: Help for stochiometric coefficients' on page 109. To display (or print) the current values of all species stochiometric coefficient the user should enter the 'PRINT' command. To return to the parameter command mode the user should enter 'END'.

Species stochiometric coefficients can be modified in two different ways:

1. With the full-screen system editor.
2. With a series of instructions.

Using the system editor (XEDIT in the case of the VM/370 operating system) is of course the easiest way to modify stochiometric coefficients. The editor is invoked by the instruction 'EDIT'. A table with all the current stochiometric coefficient values is written to disk and appears at the user's display. All the normal edit commands are available. The edit session is terminated in the usual way (by 'QUIT' or 'FILE') and the result is read by the program.

The system editor cannot be used

1. In a batch run.
2. Under many other operating systems.

Therefore a second method to modify stochiometric coefficients was implemented which is more cumbersome but works always. With this method the user must indicate which coefficient should be altered and the new value to be assigned to it. This procedure is illustrated by the following example.
Begin Sample Session

Parameter cmd:
s
Stoichiometry cmd:
pr

Present species stoichiometry:
Spec. name  Spec. ext  NITROGEN  PHOSPHOR  SILICON  CHLTOCAR  CARTODRY
DIATOMS   0.750D-04   0.05000   0.00750   0.22000   25.0000   3.00000
DIATOMS   0.650D-04   0.05000   0.00350   0.15000   40.0000   2.50000
FLAGELAT  0.750D-04   0.10500   0.00880   0.00070   30.0000   2.50000
GREENS    0.610D-04   0.11000   0.00880   0.00070   30.0000   2.00000
GREENS    0.610D-04   0.05900   0.00460   0.00070   40.0000   2.00000
GREENS    0.610D-04   0.08000   0.00460   0.00070   40.0000   2.00000
APHANIZO  0.150D-03   0.08800   0.00500   0.00070   30.0000   2.50000
APHANIZO  0.150D-03   0.00000   0.00500   0.00070   30.0000   2.50000
APHANIZO  0.150D-03   0.06800   0.00200   0.00070   30.0000   2.50000
MICROCYST 0.120D-03   0.07500   0.01000   0.00070   40.0000   2.50000
MICROCYST 0.120D-03   0.04000   0.00800   0.00070   60.0000   2.50000
MICROCYST 0.120D-03   0.06000   0.00650   0.00070   60.0000   2.50000
OSCILAT   0.135D-03   0.07500   0.00750   0.00070   30.0000   2.50000
OSCILAT   0.135D-03   0.04000   0.00600   0.00070   50.0000   2.50000
OSCILAT   0.135D-03   0.06000   0.00350   0.00070   50.0000   2.50000

Stoichiometry cmd:
nu
Species name:
dia
There are 2 types of species DIATOMS
Type number:
2
Nutrient name:
si
Stoichiometric coeff:
0.0173
New value of SILICON coefficient for type 2 in species DIATOMS = 0.01730
Stoichiometry cmd:
en

Parameter cmd:

End Sample Session

Notes:

1. As usually the same instruction could have been entered much more compactly on one line as: 'N DIA 2 S .173'.
2. If a species consists of more than one type, the user must include the relative type number, for example 'OSCI 2' for the second Oscillatoria type. Numbers are invalid for species consisting of only one type.
3. The program indicates correction measures for any invalid instruction in an interactive run. In a batch run, of course, errors are fatal and cause termination of the program.

The optional input of BLOOM II 71
6.4.3 Phytoplankton species growth coefficients

As we have explained earlier, the actual variables of the model are phytoplankton types rather than species. Therefore the species stochiometry consists of the coefficients all its types. The initial growth coefficients of the phytoplankton types is read from unit FT05F001 (See 'The standard input of BLOOM II' on page 51). To change any of these coefficients for one or several types, the user should first activate the parameter command mode and then enter the instruction 'GROWTH' to put the program into 'Growth command mode'. The user is notified by the message:

Growth cmd:

which is displayed on his terminal. In a batch run this message is suppressed.

A short description of the 'Growth command mode' including a list of valid coefficient names is displayed (or printed) following the instruction '?'. The text is shown in 'Appendix 4: Help for growth coefficients' on page 110. To display (or print) the current values of all growth coefficient the user should enter the 'PRINT' command. To return to the parameter command mode the user should enter 'END'.

Growth coefficients can be modified in two different ways:

1. With the full-screen system editor.
2. With a series of instructions.

Using the system editor (XEDIT in the case of the VM/370 operating system) is of course the easiest way to modify growth coefficients. The editor is invoked by the instruction 'EDIT'. A table with all the current growth coefficient values is written to disk and appears at the user's display. All the normal edit commands are available. The edit session is terminated in the usual way (by 'QUIT' or 'FILE') and the result is read by the program.

The system editor cannot be used

1. In a batch run.
2. Under many other operating systems.

Therefore a second method to modify growth coefficients was implemented which is more cumbersome but works always. With this method the user must indicate which coefficient should be altered and the new value to be assigned to it. This procedure is illustrated by the following example.
Parameter cmd:
gro
Growth cmd:
pr
Present species growth coefficients:
Species Pmax 1 Pmax 2 P-func. Mort1 Mort2 Resp1 Resp2 Relm Zpr
DIATOMS 0.500 1.038 EXPONENT 0.065 1.070 0.031 1.096 1.0 1.0
DIATOMS 0.420 1.038 EXPONENT 0.065 1.070 0.031 1.096 1.0 1.0
FLAGELAT 0.330 1.058 EXPONENT 0.065 1.070 0.031 1.096 1.0 1.0
GREENS 0.088 5.000 LINEAR 0.065 1.070 0.031 1.096 1.0 1.0
GREENS 0.053 10.000 LINEAR 0.065 1.070 0.031 1.096 1.0 1.0
GREENS 0.092 10.000 LINEAR 0.065 1.070 0.031 1.096 1.0 1.0
APHANIZO 0.180 1.081 EXPONENT 0.065 1.070 0.006 1.072 1.0 0.0
APHANIZO 0.033 25.000 LINEAR 0.065 1.070 0.006 1.072 1.0 0.0
APHANIZO 0.120 1.095 EXPONENT 0.065 1.070 0.006 1.072 1.0 0.0
MICROCYST 0.050 5.000 LINEAR 0.065 1.070 0.006 1.072 1.0 0.0
MICROCYST 0.053 8.000 LINEAR 0.065 1.070 0.006 1.072 1.0 0.0
MICROCYST 0.053 8.000 LINEAR 0.065 1.070 0.006 1.072 1.0 0.0
OSCILAT 0.250 1.062 EXPONENT 0.065 1.070 0.006 1.072 1.0 0.0
OSCILAT 0.161 1.077 EXPONENT 0.065 1.070 0.006 1.072 1.0 0.0
OSCILAT 0.161 1.077 EXPONENT 0.065 1.070 0.006 1.072 1.0 0.0
growth cmd:
resp
Species name:
gr
There are 3 types of species GREENS
Type number:
1
  Resp. temp. coeff:
  0.025
  Resp. const. coeff:
  1.093
New value of RESPIRAT for type 1 of species GREENS:

RESPIRAT = 0.0250 * 1.0930 ** T

Growth cmd:
end

End Sample Session

Notes:

1. As usually the same instruction could have been entered much more compactly on one line as: "RESP G 1.025 1.093".
2. The program indicates corrective actions for any invalid instruction in an interactive run. In a batch run, of course, errors are fatal and cause termination of the program.

The optional input of BLOOM II 73
6.4.4 Zooplankton coefficients

The primary input of the zooplankton coefficients such as the nutrient contents or the maximum number of grazing iterations has been described earlier (See 'The standard input of BLOOM II' on page 51). To change any of these coefficients the user should first activate the parameter command mode and then enter the instruction 'ZOOPLANK' to put the program into 'Zooplankton command mode'. The user is notified by the message:

Zooplankton cmd:

which is displayed on his terminal. In a batch run this message is suppressed.

A short description of the 'Zooplankton command mode' including a list of valid coefficient names is displayed (or printed) following the instruction '?' The text is shown in 'Appendix 5: Help for zooplankton coefficients' on page 111. To display (or print) the current values of all zooplankton coefficients the user should enter the 'PRINT' command. To return to the parameter command mode the user should enter 'END'.

Because presently there is only one species of zooplankton in the model, it was not felt necessary to use the system editor to alter zooplankton coefficient values. The following sample session demonstrates how zooplankton parameters can be modified.
Begin Sample Session

Parameter cmd:
zo
Zooplankton cmd:
p
Present zooplankton characteristics:
Nutrient coefficient for NITROGEN 0.0833
Nutrient coefficient for PHOSPHOR 0.0067
Nutrient coefficient for SILICON 0.0001
Half saturation constant ZOOK = 2000.0
Phytoplankton escaping grazing XMIN = 250.0
Grazing rate constant ZOOGR = 1.0000 per day
Grazing rate first iteration step = 0.0
Maximum number of iterations = 3
Zooplankton cmd:
ha
Half saturation constant:
1500
New value of HALFSAT = 1500.0000
Zooplankton cmd:
nut
Nutrient name:
?
Incorrect nutrient name?
Valid nutrients are: NITROGEN PHOSPHOR SILICON
Nutrient name:
ni
Stochiometric coeff:
0.07
New nutrient NITROGEN coefficient = 0.0700
Zooplankton cmd:
en
Parameter cmd:

End Sample Session

Notes:

1. As usually the same instructions could have been entered much more compactly on one line as: 'H 1500' respectively 'N N .07'.
2. The program indicates corrective actions for any invalid instruction in an interactive run. In a batch run, of course, errors are fatal and cause termination of the program.

The optional input of BLOOM II 75
7.0 THE COMPUTATIONS OF BLOOM II

In Chapter 'Structure of the phytoplankton model BLOOM II' on page 1 we have shown the general outline of the BLOOM II model. Input and operation of the computer program of the model have been described in Secs. 'The standard input of BLOOM II' on page 51 and 'The optional input of BLOOM II' on page 63. To some readers it may be interesting how the computational section of the BLOOM II program is written. On the other hand, of course, a user's manual is not the proper place for a thorough discussion on computational details about the structure and efficiency of the program. Therefore we have only included some of the highlights.

For a phytoplankton model with (presently) 6 species and 15 types, the total computation time of BLOOM II is remarkably short. For example the steady state runs for a one year, weekly computation of Lake IJssel only take about 11 seconds of CPU on DHL's IBM 4361, costing about 11 Dutch guilders at the 1985 price level. With additional growth constraints the costs increase to about 15 Dutch guilders. With a different set of environmental constraints, with zooplankton data and with the maximum number of program options turned 'on', the CPU time of the current model version can increase about threefold. But even such amounts of computer time are quite acceptable to an interactive user. The corresponding costs are in the order of 30 Dutch guilders.

In a simple steady state computation, more than 50 percent of the computer time is absorbed by two activities:

1. Computation, sorting and construction of the extinction intervals.
2. Solution of the linear programs.

Of these two, the latter is most time consuming. These activities have been explained in Secs. 'Steady state assumption' on page 21, and 'Solution algorithm' on page 27. With the addition of a growth constraint for each phytoplankton species, these two activities take up more than 75 percent of the total CPU time of the BLOOM II program.

7.1 EXTINCTION ROOTS COMPUTATIONS

As we have explained in Sec. 'Steady state assumption' on page 21 each phytoplankton type can only sustain a positive net growth rate within certain limits of the total extinction. To compute these limits, Bigelow et al. [1977] have set up an iterative procedure, which is still used in the current version of BLOOM II. As a result, Kmax and Kmin can be computed for any given value of the growth and loss terms, combined in the minimum efficiency requirement Emin(T), as shown in Eq. (1.28), with the solar intensity, the mixing depth, and the day length.

When we want to add the optional growth constraints, we need the rate of growth (hence the efficiency) given the extinction to compute the right hand sides of these constraints. In other words Eq. (1.21) must be solved for EAVG rather than for K. Because an algo-
rithm to compute $K$ given $EAVG$ is already available in the program, we have set up the following iteration procedure:

1. Compute the extinction roots $K_{min}$ and $K_{max}$ for each type $k$ as usual, which is necessary anyhow.
2. Specify or compute the extinction value against which each species should compete\(^1\).
3. Compute an initial estimate of the growth efficiency $EAVG(0)$. As (1) we know by experience that the denominator of Eq. (1.21) strongly dominates the final result and as (2) the depth and day length are constant, the following approximately holds:

\[
EAVG \times K = K_{max} \times E_{min}(T)
\]  

(7.1)

Hence given $K_{max}$, $E_{min}(T)$ and $K$, $EAVG(0)$ is estimated as

\[
EAVG(0) = \frac{K_{max} \times E_{min}(T)}{K}
\]  

(7.2)

4. Compute the extinction $K(0)$ corresponding to $EAVG(0)$ with the normal routines in the program and compare $K(0)$ with $K$. If the agreement is well enough, $EAVG(0)$ is a good estimator for the average efficiency of a species' $EAVG$ value.
5. Otherwise compute the next value $EAVG(1)$ as

\[
EAVG(1) = EAVG(0) \pm 0.025
\]  

(7.3)

according as $EAVG(0)$ is too high or too low. The (small) value 0.025 was chosen because experience has shown that under most conditions $EAVG(0)$ is very close to $EAVG$. If the agreement is well enough, $EAVG(1)$ is a good estimator for the average efficiency of a species. Otherwise repeat this step as many times as necessary.
6. The procedure stops after iteration $i$ if

\[
K(i) = K \pm 0.05
\]  

(7.4)

An extinction value of 0.05 corresponds to a phytoplankton biomass in the order of 1 to 10 mg chlorophyll/m$^3$, depending on the specific extinction and chlorophyll to biomass conversion.

\(^1\) This presents an interesting problem by itself, because there are at least two reasonable, but distinct assumptions. We could use (1) the background extinction and minimize the impact of previous solutions, (2) the total extinction. In the current version of the model we use the second assumption. Notice, however, that no mortality is taken into account when the growth constraint is computed (See 'Formulation of growth constraints' on page 25).
In most cases the stop criterion is already met in the first iteration for each phytoplankton species. Several iterations are only necessary in very shallow lakes with relatively high extinctions.

Certain additional checks and a modified iteration scheme are build into the program in case the previously described algorithm would fail.

The three subroutines which perform the bulk of these computations have been compiled with an optimizing compiler to increase their performance. This reduces the total required amount of CPU by about 5 percent.

7.2 SOLUTION OF THE LINEAR PROGRAMS

The linear programs of BLOOM II are solved by a subroutine based on the revised simplex algorithm [Danzig, 1963]. This subroutine is the heart of the BLOOM II program. It performs the most important computations and it consumes a considerable part of the total computation time. This is caused by

1. The number of linear programs to be solved. As we have explained in 'Solution algorithm' on page 27 one linear program should be solved for each extinction interval. In most applications the number of intervals varies between 3 and 10 per time-step, hence the total number of intervals for a one-year computation with a time-step of one week is in the order of 150 to about 500.

2. The dimension of the linear programs to be solved, if the optional growth constraints are included. Under steady state conditions each linear program consists of only 6 rows and N variables where N, the number of phytoplankton types, is usually 15 so far. The computation time for each interval is in the order of 0.1 seconds CPU on DHL's IBM 4361. But with the addition of one growth constraint for each species, the number of rows is significantly increased to (6+N). The computation time for each interval now rises with about a factor of two to approximately 0.2 seconds CPU.

To optimize this (important) part of the program, we have

1. Compared the performance of four different subroutines.
2. Chosen the best possible initial solution for the simplex algorithm.
3. Incorporated several checks to determine certain infeasibilities before the simplex routine is started.
4. Compiled the subroutine by an optimizing compiler.

Comparison of different subroutines

We have performed a large number of computations with initially four and later two different simplex subroutines: DSIMPL written by the Rand Corp. and used in BLOOM II until 1982, ZX3LP of the IMSL library, and DOSP and DOSD of IBM's mathematical library. In these computations we have both considered cases where the number of variables exceeds the number of constraints (growth constraints not
considered), as well as cases with more constraints than variables (growth constraints are included).

Of these four subroutines the IMSL routine proved to be much slower than the rest under all considered conditions. The difference was in the order of a factor 2 when the optimizing compiler had been used to even a factor 3 when all subroutines were compiled with the normal IBM FORTRAN compiler. Of the other three subroutines, DOSD was also slower, although the difference with the other two routines was relatively small when the number of rows was large.

Subroutines DSIMPL and DOSP were about equally fast in the test computations with a 6x10 matrix, but DOSP was somewhat faster with a 16x10 matrix. Therefore two program versions of BLOOM II were developed to compare the performance of these two subroutines in a number of representative cases. Initially there was no significant difference both with and without growth constraints. However, using the so-called re-start option of DOSP (see below) we could increase the efficiency of this subroutine considerably and make it faster than DSIMPL. The total CPU time of BLOOM II was thus reduced by 13 to 23 percent in the test cases. DOSP is therefore used in the current version of the model.

The start basis for the simplex algorithm

The simplex algorithm solves a linear program step by step. Usually it starts with all variables (in the case of BLOOM II phytoplankton types) put to zero. The algorithm then selects a promising variable and replaces one constraint by this variable: now we have one type limited by one factor. This procedure may have to be repeated several times to put more variables into the solution. Hence a solution with M phytoplankton types results from at least M simplex iterations. The actual number of iterations may even be larger, because the algorithm may sometimes select a promising type, which is rejected in later iteration steps anyway. In the case of BLOOM II the number of iterations can become quite large (between 5 and 10) when the growth constraints are included in the linear program.

Subroutine DOSP offers the possibility to specify for each variable whether it should be initiated at zero or not (the re-start option). This is highly useful when we know the solution of an earlier computation which resembles the computation to be started in terms of row coefficients and right hand sides. In that case we can reduce the number of simplex iterations considerably.

In BLOOM II we have used the re-start option of DOSP in the following way. At each time-step we solve the linear program of the first extinction interval with the normal starting assumption that all types are zero. As we have shown in 'Solution algorithm' on page 27, however, the other linear programs to be solved at this time-step share many characteristics with the first because each extinction interval has exactly the same nutrient and (optional) growth constraints. The only difference is an increase in the permissible extinction levels and a decrease in the number of permitted phytoplankton types. Hence at the start of interval i we

---

2 It has been shown that the simplex algorithm will always convert to the same solution, regardless of the intermediate iteration steps. Hence starting with the wrong variables in the solution downgrades convergence, but does not result in an incorrect final answer.
include those types in the initial solution which satisfy two conditions:

1. They were in the optimal LP solution of the previous interval i-1.
2. They are permitted to grow (have a positive extinction range) in interval i.

In case a type was optimal in interval i-1, but the extinction of interval i is too high, it is not included in the starting solution. Moreover an additional action is taken to prevent that it could be considered as a promising type at some iteration step of the simplex algorithm: this type is no longer maximized.

Use of the re-start option for successive extinction intervals has reduced to total computation time of BLOOM II by 13 to 23 percent. The greater the similarity between the solutions of intervals, the larger the reduction in computation time.

Checks to determine infeasibilities without simplex

In two cases it is not necessary to call the simplex subroutine because there cannot be a feasible solution for interval i:

1. Both extinction roots are negative.
2. Interval i-1 was infeasible and no type are allowed to grow in interval i which were excluded from interval i-1.

The first is obvious: no type can grow at negative extinction levels. The latter perhaps requires some explanation. When the available concentration of some nutrient \(N_i\) is low compared with the availability of solar energy, \(N_i\) will be completely exhausted by a phytoplankton biomass with a relatively low total extinction. Thus the lower limit of a specific extinction interval i-1 can already be so high that no type can both satisfy this extinction constraint and the constraint of nutrient \(N_i\). This infeasibility is determined by the simplex subroutine. Obviously, none of the higher intervals i, i+1, ..., can be feasible because the extinction limits become higher and higher. Therefore the program checks whether this situation occurs before calling the simplex subroutine.

Only if a new type would be permitted in some interval i, which means it was limited by photoinhibition previously, we have to use the simplex algorithm again to determine the feasibility. But this situation has never occurred in the cases considered so far.

Use of optimizing compiler

Each of the four simplex subroutines which we have considered have been compiled with and without compiler optimization. The results of similar LP problems indicated that compiler optimization could decrease the CPU requirement by approximately 25 percent in three out of four cases. With the IMSL subroutine ZX3LP the reduction was even substantially larger: about 40 percent!

Obviously, we use the optimized version of DOSP in BLOOM II.

The computations of BLOOM II 80
8.0 THE OUTPUTS OF BLOOM II

As we have explained in Sec. 'Job control for BLOOM II' on page 46, BLOOM II writes its outputs to several different files. The minimum number of output files is three for a batch, and four for an interactive job. Several options, which by default are 'off', each write their output to a different file when they are activated by the user. Thus the specifications of the job determine how many output files are actually used. In addition the program will sometimes create intermediate output files, for example to transfer data between different program sections.

The procedure with which the BLOOM program is executed under VM/370 questions the user at the end of an interactive job (1) if he wants to print the output of the program and (2) if he wants to keep a copy of the output on disk. The procedure automatically sorts the output files in the most logical order. Existing output files can be printed by a special procedure called 'PRTBLOOM'.

With the exception of the (optional) print-plot file, the (optional) input file for OXYMOD and the so-called 'DUMP' file, all the output files of BLOOM II contain carriage control symbols to start certain output sections on a new page. The print-plot file should always be printed with a fixed page length of 62 lines per page.

8.1 THE STANDARD OUTPUT OF BLOOM II

When BLOOM II terminates normally, it always creates output for files FT09F001, FT10F001 and FT14F001 in both a batch and an interactive job, and for FT21F001 only in an interactive job. If the program performs zooplankton iterations, additional output is written to FT15F001. We shall describe these outputs in the following sections and show a (sometimes small) example for each of them.

8.1.1 Title page (FT09F001)

The output of the program starts with a title page containing the name BLOOM II in large characters. In addition this page contains the name Delft Hydraulics Laboratory, the name of the programmer, and the date on which the version of the program was created which was used by the job.

---

1 Carriage control (CC) symbols are special characters in one or several reserved columns of a file, which instruct the printer to perform certain actions such as a page skip. CC symbols themselves are not printed.

2 The 'BLOOM EXEC' adds time information to this file after a run is completed. So this extra output, which is shown in Figure 8.1 on page 82, is not produced by the actual BLOOM program.
Figure 8.1 Example of output file FT09F001. The title page.
8.1.2 Summarized solution (FT10F001 and FT21F001)

BLOOM's most essential output is a summary table with biomasses and limiting factors. The normal output begins with the title of the run and then gives

1. The period number.
2. The limiting factors abbreviated as NIT = nitrogen, PHO = phosphor, SIL = silicon, KMI = Kmin: the lower limit of the extinction interval, KMA = Kmax: the upper limit of the extinction interval (Energy limitation), and GRO = growth (optional).
3. The input zooplankton concentration in mg dry weight per m³.
4. The computed phytoplankton concentration of each species in mg dry weight per m³.
5. The computed total phytoplankton concentration in mg dry weight per m³.
6. The computed total phytoplankton concentration in mg chlorophyll per m³.

In a batch job these outputs are written on disk file unit 10, in an interactive job on disk file unit 21. However, as we may see in Figure 8.2 on page 84, this output file is more than eighty characters wide, so it will not fit on to a terminal display. Therefore a condensed copy of these outputs is written to the display terminal in an interactive job. The normal uncondensed output is send to FT21F001 to be printed afterwards. In the condensed output (FT10F001), as we may see in Figure 8.3 on page 85, limiting factors are abbreviated to one character and the phytoplankton biomass concentrations are converted to grams dry weight per m³.

If a program option called 'DOMINANCE' has been activated, both output files are extended with a short table showing

1. The total and average biomass (mg dry weight per m³) of each type over all periods considered, and its percentage of the total phytoplankton biomass.
2. The average total phytoplankton biomass (g dry weight per m³).
3. The average total chlorophyll concentration of all species (mg chlorophyll per m³).

If this table is created it is written to a separate page, but we have not done that here to reduce white space in this manual.

Because FT10F001 and FT21F001 are the standard output files, eventual warning and error messages are also written to these files. These messages are self explanatory and in the case of an error, the last actions performed by the program.

In addition all prompting messages in an interactive job are written to unit FT10F001.
YEAR 1977  GROTE RUG, RING2--WABASIM CASES -- FEBRUARY 1983
DDMI CURVES; BUOYANCY VIA EXTINCTION; GROWTH CHECK; NO GRAZING.
Variable Stoichiometry; Incubator Results.

30 MAY 1985

Summary of final solutions for this run:

<table>
<thead>
<tr>
<th>DATE</th>
<th>LIMITING FACTORS</th>
<th>ZOOD</th>
<th>DIATOMS</th>
<th>FLAGELAT GREENS</th>
<th>APHANIZO MICROCYC OSCIAT</th>
<th>TOTAL</th>
<th>CHL-PRED</th>
</tr>
</thead>
<tbody>
<tr>
<td>WK 4</td>
<td>PHO</td>
<td>379.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3348.1</td>
<td>3348.1</td>
</tr>
<tr>
<td>WK 8</td>
<td>PHO KMA</td>
<td>196.</td>
<td>2942.9</td>
<td>0.0</td>
<td>392.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>WK12</td>
<td>PHD SIL</td>
<td>287.</td>
<td>1683.9</td>
<td>0.0</td>
<td>0.0</td>
<td>622.4</td>
<td>2306.3</td>
</tr>
<tr>
<td>WK16</td>
<td>PHO KMA</td>
<td>719.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3193.6</td>
<td>3193.6</td>
</tr>
<tr>
<td>WK20</td>
<td>PHO KMA</td>
<td>763.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5771.8</td>
<td>0.0</td>
</tr>
<tr>
<td>WK24</td>
<td>PHO</td>
<td>159.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>28600.0</td>
<td>0.0</td>
</tr>
<tr>
<td>WK28</td>
<td>PHO KMA</td>
<td>972.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>479.3</td>
<td>2307.7</td>
</tr>
<tr>
<td>WK32</td>
<td>PHO KMA</td>
<td>976.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1309.7</td>
<td>481.1</td>
</tr>
<tr>
<td>WK36</td>
<td>PHO KMA</td>
<td>854.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>342.3</td>
<td>2559.7</td>
</tr>
<tr>
<td>WK40</td>
<td>PHO</td>
<td>399.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1095.7</td>
</tr>
<tr>
<td>WK44</td>
<td>PHO SIL KMA</td>
<td>189.</td>
<td>323.3</td>
<td>1370.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>WK48</td>
<td>PHO</td>
<td>267.</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>WK52</td>
<td>SIL KMA</td>
<td>119.</td>
<td>6172.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1731.9</td>
</tr>
</tbody>
</table>

Summary of output for 13 periods:

<table>
<thead>
<tr>
<th>Species name</th>
<th>Total</th>
<th>Average</th>
<th>Dominance</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIATOMS</td>
<td>323.3</td>
<td>24.9</td>
<td>0.5%</td>
</tr>
<tr>
<td>DIATOMS</td>
<td>10799.6</td>
<td>830.7</td>
<td>15.3%</td>
</tr>
<tr>
<td>FLAGELAT</td>
<td>1370.4</td>
<td>105.4</td>
<td>1.9%</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0%</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0%</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>5771.8</td>
<td>444.0</td>
<td>8.2%</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0%</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>28792.1</td>
<td>2214.8</td>
<td>40.9%</td>
</tr>
<tr>
<td>MICROCYC</td>
<td>821.7</td>
<td>63.2</td>
<td>1.2%</td>
</tr>
<tr>
<td>MICROCYC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0%</td>
</tr>
<tr>
<td>MICROCYC</td>
<td>1309.7</td>
<td>100.7</td>
<td>1.9%</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>8760.7</td>
<td>673.9</td>
<td>12.4%</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0%</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>12533.0</td>
<td>964.1</td>
<td>17.8%</td>
</tr>
</tbody>
</table>

Average total biomass = 5.42 grams dry weight per cubic meter.
Average chlorophyll = 63.5 mg per cubic meter.
Figure 8.3 Example of output file FT10F001 (interactive). This file contains the summarized solutions per week and (optionally) a table with average biomass values of all computation periods in condensed format. See also Figure 8.2 on page 84.

YEAR 1977  GROTE RUG, RING2--WABASIM CASES -- FEBRUARY 1983  
DDMI KURVES; BUOYANCY VIA EXTINCTION; GROWTH CHECK; NO GRAZING.  
Variable Stoichiometry; Incubator Results.

Summary of final solutions for this run:

<table>
<thead>
<tr>
<th>DATE LIMIT FACT</th>
<th>DIAT-</th>
<th>FLAG-</th>
<th>GREE-</th>
<th>APHA-</th>
<th>MICR-</th>
<th>OSCI-</th>
<th>TOTA-</th>
<th>CHL-</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DMS</td>
<td>ELAT</td>
<td>NS</td>
<td>NIZO</td>
<td>OCYS</td>
<td>LAT</td>
<td>L</td>
<td>PRED</td>
</tr>
<tr>
<td>4 P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.3</td>
<td>3.3</td>
<td>26.8</td>
</tr>
<tr>
<td>8 P K</td>
<td>2.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>0.0</td>
<td>0.0</td>
<td>3.3</td>
<td>34.7</td>
</tr>
<tr>
<td>12 P S</td>
<td>1.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.6</td>
<td>2.3</td>
<td>21.8</td>
</tr>
<tr>
<td>16 P K</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3.2</td>
<td>3.2</td>
<td>30.1</td>
</tr>
<tr>
<td>20 P K</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5.8</td>
<td>0.0</td>
<td>0.2</td>
<td>5.9</td>
<td>79.2</td>
</tr>
<tr>
<td>24 P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>28.4</td>
<td>0.0</td>
<td>0.0</td>
<td>28.4</td>
<td>378.7</td>
</tr>
<tr>
<td>28 P K</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
<td>2.3</td>
<td>2.8</td>
<td>23.3</td>
</tr>
<tr>
<td>32 P K</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.3</td>
<td>0.5</td>
<td>1.8</td>
<td>12.6</td>
</tr>
<tr>
<td>36 P K</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.3</td>
<td>2.6</td>
<td>2.9</td>
<td>23.9</td>
</tr>
<tr>
<td>40 P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.1</td>
<td>1.1</td>
<td>14.6</td>
</tr>
<tr>
<td>44 P S K</td>
<td>0.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.9</td>
<td>2.6</td>
<td>29.5</td>
</tr>
<tr>
<td>48 P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.9</td>
<td>4.9</td>
<td>65.6</td>
</tr>
<tr>
<td>52 S K</td>
<td>6.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.7</td>
<td>7.9</td>
<td>84.8</td>
</tr>
</tbody>
</table>

Summary of output for 13 periods.

<table>
<thead>
<tr>
<th>Species name</th>
<th>Total</th>
<th>Average</th>
<th>Dominance</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIATOMS</td>
<td>323.3</td>
<td>24.9</td>
<td>0.5 %</td>
</tr>
<tr>
<td>DIATOMS</td>
<td>10799</td>
<td>830.7</td>
<td>15.3 %</td>
</tr>
<tr>
<td>FLAGELAT</td>
<td>1370</td>
<td>105.4</td>
<td>1.9 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>5771</td>
<td>444.0</td>
<td>8.2 %</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>28792.1</td>
<td>2214.8</td>
<td>40.9 %</td>
</tr>
<tr>
<td>MICROCYCS</td>
<td>821.7</td>
<td>63.2</td>
<td>1.2 %</td>
</tr>
<tr>
<td>MICROCYCS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>MICROCYCS</td>
<td>1309.7</td>
<td>100.7</td>
<td>1.9 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>8760</td>
<td>673.9</td>
<td>12.4 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>12533</td>
<td>964.1</td>
<td>17.8 %</td>
</tr>
</tbody>
</table>

Average total biomass = 5.42 grams dry weight per cubic meter.  
Average chlorophyll = 63.5 mg per cubic meter.
8.1.3 Zooplankton iterations (FT15F001)

When zooplankton is included in the input of the model (See 'The standard input of BLOOM II' on page 51) it is able to compute the impacts of zooplankton grazing on the potential phytoplankton bloom levels. This is done by an iterative procedure explained in the BLOOM II report by Los [1982]. The results of the various iteration steps are written to FT15F001 in an almost identical format as was used for FT10F001 or FT21F001. One column is added containing the iteration step number and of course several lines of output can now be produced for each time-step.

This output will be completely suppressed by the program if (1) all input zooplankton concentrations are 0.0, or if (2) the maximum number of permissible grazing iterations is less than two.
<table>
<thead>
<tr>
<th>Date</th>
<th>Limiting Factors</th>
<th>Iter Zood</th>
<th>Diatoms</th>
<th>Flagelat Greens</th>
<th>Aphanizo Microcyx</th>
<th>Oscilat</th>
<th>Total</th>
<th>CHL-Pred</th>
</tr>
</thead>
<tbody>
<tr>
<td>WK 4</td>
<td>Pho</td>
<td>1</td>
<td>379.1</td>
<td>3348.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3348.1</td>
</tr>
<tr>
<td>WK 4</td>
<td>Pho</td>
<td>2</td>
<td>379.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3348.1</td>
</tr>
<tr>
<td>WK 8</td>
<td>Pho KMA</td>
<td>1</td>
<td>196.2</td>
<td>2942.9</td>
<td>0.0</td>
<td>392.1</td>
<td>0.0</td>
<td>3335.0</td>
</tr>
<tr>
<td>WK 8</td>
<td>Pho KMA</td>
<td>2</td>
<td>196.3</td>
<td>2942.9</td>
<td>0.0</td>
<td>392.1</td>
<td>0.0</td>
<td>3335.0</td>
</tr>
<tr>
<td>WK 12</td>
<td>Pho Sil</td>
<td>1</td>
<td>287.1</td>
<td>1683.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2306.3</td>
</tr>
<tr>
<td>WK 12</td>
<td>Pho Sil</td>
<td>2</td>
<td>287.2</td>
<td>1683.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2306.3</td>
</tr>
<tr>
<td>WK 16</td>
<td>Pho Sil KMA</td>
<td>1</td>
<td>719.1</td>
<td>1332.5</td>
<td>141.1</td>
<td>0.0</td>
<td>0.0</td>
<td>3967.2</td>
</tr>
<tr>
<td>WK 16</td>
<td>Pho Sil KMA</td>
<td>2</td>
<td>719.2</td>
<td>1332.5</td>
<td>141.1</td>
<td>0.0</td>
<td>0.0</td>
<td>3967.2</td>
</tr>
<tr>
<td>WK 20</td>
<td>Pho Sil KMA</td>
<td>1</td>
<td>763.1</td>
<td>177.9</td>
<td>1483.1</td>
<td>0.0</td>
<td>0.0</td>
<td>3193.6</td>
</tr>
<tr>
<td>WK 20</td>
<td>Pho KMA</td>
<td>2</td>
<td>763.2</td>
<td>0.0</td>
<td>0.0</td>
<td>5771.8</td>
<td>0.0</td>
<td>5940.6</td>
</tr>
<tr>
<td>WK 24</td>
<td>Pho</td>
<td>1</td>
<td>159.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>28400.0</td>
<td>0.0</td>
</tr>
<tr>
<td>WK 28</td>
<td>Pho KMA</td>
<td>1</td>
<td>972.1</td>
<td>0.0</td>
<td>0.0</td>
<td>457.4</td>
<td>0.0</td>
<td>2984.5</td>
</tr>
<tr>
<td>WK 28</td>
<td>Pho KMA</td>
<td>2</td>
<td>972.2</td>
<td>0.0</td>
<td>0.0</td>
<td>457.4</td>
<td>0.0</td>
<td>2984.5</td>
</tr>
<tr>
<td>WK 32</td>
<td>Pho KMA</td>
<td>1</td>
<td>976.1</td>
<td>0.0</td>
<td>0.0</td>
<td>502.5</td>
<td>0.0</td>
<td>2000.9</td>
</tr>
<tr>
<td>WK 32</td>
<td>Pho KMA</td>
<td>2</td>
<td>976.2</td>
<td>0.0</td>
<td>0.0</td>
<td>502.5</td>
<td>0.0</td>
<td>2000.9</td>
</tr>
<tr>
<td>WK 36</td>
<td>Pho KMA</td>
<td>1</td>
<td>854.1</td>
<td>0.0</td>
<td>0.0</td>
<td>326.7</td>
<td>0.0</td>
<td>3043.1</td>
</tr>
<tr>
<td>WK 36</td>
<td>Pho KMA</td>
<td>2</td>
<td>854.2</td>
<td>0.0</td>
<td>0.0</td>
<td>326.7</td>
<td>0.0</td>
<td>3043.1</td>
</tr>
<tr>
<td>WK 40</td>
<td>Pho Sil KMA</td>
<td>1</td>
<td>399.1</td>
<td>131.1</td>
<td>0.0</td>
<td>324.9</td>
<td>0.0</td>
<td>1098.8</td>
</tr>
<tr>
<td>WK 40</td>
<td>Pho</td>
<td>2</td>
<td>399.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1095.7</td>
<td>0.0</td>
</tr>
<tr>
<td>WK 44</td>
<td>Pho Sil KMA</td>
<td>1</td>
<td>189.1</td>
<td>472.3</td>
<td>1255.8</td>
<td>0.0</td>
<td>0.0</td>
<td>3103.4</td>
</tr>
<tr>
<td>WK 44</td>
<td>Pho Sil KMA</td>
<td>2</td>
<td>189.2</td>
<td>472.3</td>
<td>1255.8</td>
<td>0.0</td>
<td>0.0</td>
<td>3103.4</td>
</tr>
<tr>
<td>WK 48</td>
<td>Pho Sil KMA</td>
<td>1</td>
<td>287.1</td>
<td>2314.8</td>
<td>2818.3</td>
<td>0.0</td>
<td>0.0</td>
<td>6270.7</td>
</tr>
<tr>
<td>WK 48</td>
<td>Pho</td>
<td>2</td>
<td>287.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>6917.9</td>
<td>0.0</td>
</tr>
<tr>
<td>WK 52</td>
<td>Sil KMA</td>
<td>1</td>
<td>119.1</td>
<td>6165.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9499.4</td>
</tr>
<tr>
<td>WK 52</td>
<td>Sil KMA</td>
<td>2</td>
<td>119.2</td>
<td>6172.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7903.5</td>
</tr>
<tr>
<td>WK 52</td>
<td>Sil KMA</td>
<td>3</td>
<td>119.3</td>
<td>6172.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7904.7</td>
</tr>
</tbody>
</table>

Figure 8.4: Example of output file FT15001. This file contains the summarized solutions per week for each coexistence experiment iteration step. See also Figure 8.2 on page 84.
8.1.4 Nutrient fractions (FT14F001)

File FT14F001 contains the remaining standard outputs. Following the title, one line is written for each period showing

1. The period number.
2. The partitioning of each of the nutrients (usually nitrogen, phosphor, and silicon) in a planktonic and a dissolved fraction. A value of 0.0 for the dissolved fraction of a nutrient indicates that it is limiting (See also 'Summarized solution (FT10F001 and FT21F001)' on page 83).
3. The total extinction per m.
4. The computed total chlorophyll concentration in mg per m$^3$ (See also 'Summarized solution (FT10F001 and FT21F001)' on page 83).
5. The observed total chlorophyll concentration in mg per m$^3$.

---

3 BLOOM divides the total available nutrients over the following four pools: (1) live phytoplankton, (2) dead phytoplankton (detritus), (3) zooplankton and (4) remaining. The sum of the first three pools is called planktonic, the last pool is called dissolved.
<table>
<thead>
<tr>
<th>DATE</th>
<th>PLANK NITROGEN</th>
<th>DISS NITROGEN</th>
<th>PLANK PHOSPHOR</th>
<th>DISS PHOSPHOR</th>
<th>PLANK SILICON</th>
<th>DISS SILICON</th>
<th>TOT EXT</th>
<th>CHL-PRED</th>
<th>CHL-OBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>WK 4</td>
<td>776.05</td>
<td>3100.45</td>
<td>45.50</td>
<td>0.00</td>
<td>5.52</td>
<td>1776.98</td>
<td>1.20</td>
<td>26.78</td>
<td>8.25</td>
</tr>
<tr>
<td>WK 8</td>
<td>529.76</td>
<td>3577.74</td>
<td>35.25</td>
<td>0.00</td>
<td>1369.33</td>
<td>203.17</td>
<td>0.98</td>
<td>34.66</td>
<td>22.00</td>
</tr>
<tr>
<td>WK12</td>
<td>430.45</td>
<td>3535.30</td>
<td>29.00</td>
<td>0.00</td>
<td>1252.50</td>
<td>0.00</td>
<td>0.95</td>
<td>21.82</td>
<td>34.50</td>
</tr>
<tr>
<td>WK16</td>
<td>644.20</td>
<td>3182.30</td>
<td>46.50</td>
<td>0.00</td>
<td>9.60</td>
<td>667.70</td>
<td>1.27</td>
<td>30.06</td>
<td>47.50</td>
</tr>
<tr>
<td>WK20</td>
<td>1105.96</td>
<td>2176.79</td>
<td>65.25</td>
<td>0.00</td>
<td>15.66</td>
<td>146.64</td>
<td>1.63</td>
<td>37.91</td>
<td>73.25</td>
</tr>
<tr>
<td>WK24</td>
<td>2167.68</td>
<td>812.02</td>
<td>95.75</td>
<td>0.00</td>
<td>28.53</td>
<td>261.17</td>
<td>5.11</td>
<td>378.67</td>
<td>288.00</td>
</tr>
<tr>
<td>WK28</td>
<td>617.09</td>
<td>2187.91</td>
<td>46.75</td>
<td>0.00</td>
<td>15.91</td>
<td>409.09</td>
<td>1.27</td>
<td>23.25</td>
<td>63.25</td>
</tr>
<tr>
<td>WK32</td>
<td>356.10</td>
<td>2641.60</td>
<td>33.50</td>
<td>0.00</td>
<td>12.58</td>
<td>557.42</td>
<td>1.06</td>
<td>12.58</td>
<td>14.00</td>
</tr>
<tr>
<td>WK36</td>
<td>479.91</td>
<td>2502.59</td>
<td>33.50</td>
<td>0.00</td>
<td>15.78</td>
<td>506.72</td>
<td>1.29</td>
<td>23.90</td>
<td>12.50</td>
</tr>
<tr>
<td>WK40</td>
<td>353.05</td>
<td>2375.45</td>
<td>34.75</td>
<td>0.00</td>
<td>8.89</td>
<td>226.11</td>
<td>0.97</td>
<td>14.61</td>
<td>12.75</td>
</tr>
<tr>
<td>WK44</td>
<td>566.17</td>
<td>2601.58</td>
<td>46.75</td>
<td>0.00</td>
<td>432.50</td>
<td>100.00</td>
<td>1.03</td>
<td>29.52</td>
<td>53.00</td>
</tr>
<tr>
<td>WK48</td>
<td>760.26</td>
<td>2946.22</td>
<td>75.75</td>
<td>0.00</td>
<td>10.00</td>
<td>1005.00</td>
<td>1.48</td>
<td>65.57</td>
<td>126.75</td>
</tr>
<tr>
<td>WK52</td>
<td>804.60</td>
<td>3228.40</td>
<td>63.51</td>
<td>0.24</td>
<td>1622.50</td>
<td>0.00</td>
<td>1.36</td>
<td>84.82</td>
<td>76.75</td>
</tr>
</tbody>
</table>
8.2 THE OPTIONAL OUTPUT OF BLOOM II

In addition to the standard output, BLOOM II can create output for a number of additional files when certain program options are put 'on'. Notice, however, that some options do affect the numbers coming out of the program but do not affect the way they are written. In other words not every option creates a separate output file. With the exception of the 'DUMP' output, all output files are sent to DISK files both in an interactive and in a batch job. We shall describe these outputs in the following sections and show an example for each of them.

8.2.1 The dump output (FT06F001)

On several occasions the user of BLOOM II may want to obtain a more detailed output than the summarized output described previously. He may, for example, want to know the extent to which the maximum LP solution differs from the solutions of other extinction intervals (See 'Solution algorithm' on page 27). Or he may be interested in the values of certain coefficients such as Pnmax(T) at each time-step. For these purposes we have installed the 'DUMP' option in the program. When this option is active, the program writes the following information to file FT06F001 for each time-step and for each zooplankton iteration:

1. The title of the run and the period number.
2. The values of a number of coefficients:
   • Temperature in °C.
   • Solar radiation in Joules/m²/week of photosynthetically active radiation.
   • Total depth in m.
   • The maximum net production rate constant (Pnmax(T)), the respiration rate constant (R(T)), and the mortality rate constant (M(T)) of each type per day.
   • The temperature correction for the efficiency curves (See Eq. (1.20) for an explanation).
3. The zooplankton iteration number.
4. The following information for each extinction interval:
   • The interval number.
   • Lower and upper extinction limit.
   • The numbers of the types with a positive net growth rate.
   • The slacks⁴ of the nutrients, energy and (optional) growth constraints.
   • The biomass of each phytoplankton type in mg dry weight per m³.
   • The total phytoplankton biomass in mg dry weight per m³.
5. A table with the maximum solution for the time-step showing biomass of each type, total biomass, chlorophyll, total available and slacks of nutrients.

⁴ In the present context it is sufficient to know that a value of 0.0 for slack i indicates that constraint i is limiting. A positive value indicates it is not limiting. The order of the constraints is usually nitrogen, phosphor, silicon, Kmin and Kmax, but this can be changed by the user.
6. A message concerning the number and status of the extinction intervals.
7. The total biomass of edible types and the grazing rate constant per day computed for this edible biomass.

Normally this file is written to the terminal in an interactive job and to a disk file in a batch job (See Figure 4.1 on page 48). Notice, however, that 'DUMP' writes as many as 5000 output lines for a whole-year computation. Sending all this information to the terminal is not practical. Thus in an interactive job the user should either use 'DUMP' for a limited number of periods (See 'Control of model coefficients' on page 67), or he must change the FILEDEF for FT06F001 (See 'Job control for BLOOM II' on page 46). For the latter he must edit the file 'BLOOM EXEC' in which the file definitions are set and replace the line

FILEDEF FT06F001 TERM

by

FILEDEF FT06F001 DISK Fn OUT6 A (LRECL 132 BLOCK 132 RECFM F

where Fn stands for filename as usually and OUT6 is the filetype and A the filemode. (Other names are of course permitted as well).

Note: in an interactive run under VM/370 it is still possible to view the dumped output even it is being sent to a disk file. As we have mentioned earlier in 'Interactive and batch processing' on page 43 it is possible in BLOOM II to view every disk file with the standard BROWSE program. Thus if the user types the following instruction in 'Option cmd' or 'Parameter cmd' mode:

cms browse fn out6

he can browse the file to which the dump information was written. The advantage of using BROWSE is that it offers many possibilities to walk quickly through the output using for example find, forward and backward commands. In other words inspection of the dumped output can be restricted to the currently interesting sections.
### Figure 8.6 Example of output file FT06F001. This file contains one output page of the 'DUMP' option of the program with intermediate results and many coefficient values.

**YEAR 1977**  
**DATE 08 AO. KINDE--HARASIM CASES -- FEBRUARY 1985**  
**DIOMI KURSESI SUDANYO VIA EXTINCTION GROWTH CHECK NO GRAZING.**  
**Variables: Stochiometry, Incubator Results.**

<table>
<thead>
<tr>
<th>START Time Period</th>
<th>WEBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Important parameter values for this week:</td>
<td></td>
</tr>
<tr>
<td>Temperature: 4.5</td>
<td>Solar radiation: 606.0</td>
</tr>
<tr>
<td>Total depth: 4.57</td>
<td></td>
</tr>
<tr>
<td>Input natural mortality rates are used</td>
<td></td>
</tr>
<tr>
<td>Zooplankton biomass: 178.2</td>
<td></td>
</tr>
<tr>
<td>Penct(T,%): 0.67 0.57 0.50 0.01 0.01 0.01 0.01 0.01 0.01 0.01</td>
<td></td>
</tr>
<tr>
<td>Show(T,%): 0.05 0.05 0.05 0.05 0.05 0.05 0.01 0.01 0.01 0.01</td>
<td></td>
</tr>
<tr>
<td>Penct(T,%): 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03</td>
<td></td>
</tr>
<tr>
<td>Show(T,%): 0.05 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03</td>
<td></td>
</tr>
<tr>
<td>Terr(T,%): 2.26 2.32 2.91 9.79 3.84 3.08 4.62</td>
<td></td>
</tr>
<tr>
<td>1.01 91.01 3.02 3.74 1.76</td>
<td></td>
</tr>
<tr>
<td>Zooplankton Iteration: 1</td>
<td></td>
</tr>
</tbody>
</table>

### Interval 1

| Extinction limits: 1 2 3 4 5 6 7 8 9 10 |
|-------------------|------|
| Species permitted: 1 2 3 4 5 6 7 8 9 10 |
| Nutrient Slacks: 2765.0042 0.0000 0.0000 |
| Energy Slacks: 1.0579 0.0000 |
| Species: 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 |
| Total biomass: 11760.3498 |

### Interval 2

| Extinction limits: 1 2 3 4 5 6 7 8 9 10 |
|-------------------|------|
| Species permitted: 1 2 3 4 5 6 7 8 9 10 |
| Nutrient Slacks: 2856.6757 11.0091 0.0000 |
| Energy Slacks: 0.0079 0.0000 |
| Species: 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 |
| Total biomass: 11592.2664 |

### Interval 3

| Extinction limits: 1 2 3 4 5 6 7 8 9 10 |
|-------------------|------|
| Species permitted: 1 2 3 4 5 6 7 8 9 10 |
| Nutrient Slacks: 2722.1910 3.1627 0.0000 |
| Energy Slacks: 0.2184 0.0000 |
| Species: 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 |
| Total biomass: 12815.5995 |

### Interval 4

| Extinction limits: 1 2 3 4 5 6 7 8 9 10 |
|-------------------|------|
| Species permitted: 1 2 3 4 5 6 7 8 9 10 |
| Nutrient Slacks: 2867.4625 0.0000 1054.1795 0.0000 |
| Energy Slacks: 0.3183 0.0000 |
| Species: 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 |
| Total biomass: 12245.1959 |

### Interval 5

| Extinction limits: 1 2 3 4 5 6 7 8 9 10 |
|-------------------|------|
| Species permitted: 1 2 3 4 5 6 7 8 9 10 |
| Nutrient Slacks: 2686.7075 6.7159 1334.2700 |
| Energy Slacks: 0.0072 0.0000 |
| Species: 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 |
| Total biomass: 12242.4793 |

### Interval 6

| Extinction limits: 1 2 3 4 5 6 7 8 9 10 |
|-------------------|------|
| Species permitted: 1 2 3 4 5 6 7 8 9 10 |
| Nutrient Slacks: 2677.1530 9.4033 1334.1319 |
| Energy Slacks: 0.3183 0.0000 |
| Species: 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 |
| Total biomass: 12346.4994 |

### Interval 7

| Extinction limits: 1 2 3 4 5 6 7 8 9 10 |
|-------------------|------|
| Species permitted: 1 2 3 4 5 6 7 8 9 10 |
| Error message issued for interval 7: |
| A feasible solution does not exist |

**Summary of final solutions for this run:**

**DATE LIMIT FACT**  
DIATET: AEGRE- GREE- APHA- MICH- OSCI- TOTA- OMS ELAT HS MIZO GCC L PRD

<table>
<thead>
<tr>
<th>groups</th>
<th>maximum solution</th>
<th>species</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIATOMS</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>FLAGELL</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>GREE</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>APHA</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>MICH</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>OSCI</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>TOTA</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>OMS</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>ELAT</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>HS</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>MIZO</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>GCC</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>L</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>PRD</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

**Nutrient Total**  
Slacks: 12815.5995  
Chlorophyll: 12242.4793  
Total biomass: 11547.7072  
Diatom: 5991.2129  
Flagell: 30.0000  
Gree: 2722.1910  
Apha: 76.0000  
Mich: 3.0582  
Osci: 1150.0000  
Tota: 0.0000  
Oms: 0.0000  
Elat: 0.0000  
Hs: 0.0000  
Mizo: 0.0000  
GCC: 0.0000  
L: 0.0000  
PRD: 0.0000

**Number of intervals:** 7  
**Feasible:**: 1  
**Maximum interval:** 3  
**Total biomass edible species:** 5991.21  
**Grazing death rate constant:** 0.02

---

The outputs of BLOOM II
8.2.2 Assessment of maximum grazing rates (FT16F001)

To assess the potential impacts of zooplankton grazing, there is an option in the program called 'MAXGRA'. This option computes:

1. The highest possible grazing rate constant $G_{\text{max}_k}$ of each phytoplankton species. When the grazing rate constant $G = G_{\text{max}_k}$, species $k$ has a net growth rate of 0.0.

2. The zooplankton concentration $Z_{\text{Dmax}}$ which would be necessary to produce a grazing rate constant $G_{\text{max}_k}$. In other words $Z_{\text{Dmax}}$ is the amount of zooplankton, which would makes the net growth rate of a phytoplankton type equal to 0.0.

Figure 8.7 on page 94 shows an example of the output of 'MAXGRA'. The stars for several phytoplankton types indicate that these are not grazed because the zooplankton preference factor equals 0.0 (See 'The standard input of BLOOM II' on page 51). Types with the same values for $G_{\text{max}_k}$ have the same efficiency curve, maximum net growth rate, respiration and mortality rate constant.
Maximum possible grazing rate computation.
Determine: 1. $G_{max}$ = the maximum grazing rate constant at which a phytoplankton species has a non-zero net growth rate.
2. $Z_{max}$ = the corresponding zooplankton concentration.
When more zooplankton is present than $Z_{max}$, the growth rate of species $j$ might (but does not necessarily) become negative.

<table>
<thead>
<tr>
<th>Date</th>
<th>ZOOD</th>
<th>DIATOM</th>
<th>DIATOM</th>
<th>FLAGEL</th>
<th>GREENS</th>
<th>GREENS</th>
<th>GREENS</th>
<th>APHANI</th>
<th>APHANI</th>
<th>APHANI</th>
<th>MICROC</th>
<th>MICROC</th>
<th>MICROC</th>
<th>OSCILA</th>
<th>OSCILA</th>
<th>OSCILA</th>
</tr>
</thead>
<tbody>
<tr>
<td>WK4</td>
<td>378.5</td>
<td>0.17</td>
<td>0.12</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK8</td>
<td>195.7</td>
<td>0.25</td>
<td>0.20</td>
<td>0.18</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK12</td>
<td>287.2</td>
<td>0.27</td>
<td>0.21</td>
<td>0.19</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK16</td>
<td>719.2</td>
<td>0.31</td>
<td>0.23</td>
<td>0.22</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK20</td>
<td>762.7</td>
<td>0.33</td>
<td>0.29</td>
<td>0.29</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK24</td>
<td>159.0</td>
<td>0.41</td>
<td>0.39</td>
<td>0.40</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK28</td>
<td>972.2</td>
<td>0.28</td>
<td>0.26</td>
<td>0.27</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK32</td>
<td>972.7</td>
<td>0.29</td>
<td>0.17</td>
<td>0.16</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK36</td>
<td>854.2</td>
<td>0.32</td>
<td>0.21</td>
<td>0.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK40</td>
<td>396.5</td>
<td>0.12</td>
<td>0.15</td>
<td>0.19</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK44</td>
<td>189.2</td>
<td>0.27</td>
<td>0.21</td>
<td>0.24</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK48</td>
<td>287.0</td>
<td>0.25</td>
<td>0.16</td>
<td>0.15</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>WK52</td>
<td>118.5</td>
<td>0.19</td>
<td>0.15</td>
<td>0.13</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Figure 8.7**: Example of output file F16001. This file contains the output of the MoxSA option of the program to assess the potential impacts of zooplankton grazing.
8.2.3 Steady state carbon and oxygen fluxes (FT17F001)

According to the equilibrium assumption of BLOOM II, the mass transports between various compartments are constant. The mass transports to and from the phytoplankton compartment can be computed by the model when the biomass of a phytoplankton type is multiplied by the transport rate constant per unit of time. Thus for example the (constant) transport rate to the detritus compartment equals the biomass of phytoplankton type \( k \) times the mortality rate constant \( M(T) \times x_k \).

We can make similar computations for the mass transports due to production, respiration, flushing and grazing. Without any conversion these flows would all have the unit mg dry weight per m\(^3\) per day. However, measurements of these flows are usually expressed in units mg C or mg O\(_2\) per m\(^2\) per day. Therefore the program converts its biomass output from dry weight to carbon and oxygen, and multiplies the fluxes by the depth to obtain a conventional output unit. The conversion factors have been shown in Section 5 of Figure 5.1 on page 58.

In addition BLOOM II computes the average gross production rate in mg C and mg O\(_2\) per m\(^2\) per day for all periods considered.

These computations are activated by the option 'PRODUC'. An example of an output file is shown in Figure 8.8 on page 96. Notice that all the loss rates for each species (respiration, mortality, flushing and grazing) sum up to the gross production rate constant.
Figure 8.8  Example of output file FT17F001. This file contains the mass transports at equilibrium to and from the biomass compartment of each phytoplankton species.

<table>
<thead>
<tr>
<th>Date</th>
<th>DIATOMS</th>
<th>/ORANGE</th>
<th>APHANIZA MICROCYSTOS</th>
<th>OSCILAT</th>
<th>Total-C Total-02</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEEK</td>
<td>Production</td>
<td>Respiration</td>
<td>Flushing</td>
<td>Grazing</td>
<td>Production</td>
</tr>
<tr>
<td>36</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>40</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>44</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>48</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>52</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Units are mg C/m² day or mg O2/m² day.

Average gross production rate = 1.38 grams C per m² per day.
Average gross production rate = 4.41 grams O2 per m² per day.

The outputs of BLOOM II 96
8.2.4 Diurnal oxygen budget (FT18F001)

Phytoplankton cells produce (and consume) oxygen in the light, but they only consume oxygen in the dark. Because in the light production usually exceeds consumption, the oxygen concentration will fluctuate diurnally with a maximum in the daytime and a minimum at the end of the night. The amplitude of these variations itself may be harmful to other creatures, because they are adapted to a specific, slowly changing oxygen concentration. Moreover when there is a large phytoplankton bloom, the size of the amplitude may even become so large, that the minimum value during the night becomes critical to other organisms. Thus these daily fluctuations are an important characteristic of the ecosystem.

An additional complication is the observation by many investigators that the rate of primary production is not symmetrical during the day time, but peaks already in the morning. Therefore the total production in the afternoon is smaller even if the light intensity is perfectly symmetrical. As a result the number of consecutive hours when respiration exceeds production is even further increased.

To investigate the oxygen production and consumption of phytoplankton, BLOOM II was extended with an option called 'DIURNAL' to compute the gross production, the respiration, and the net production rate for six daily periods and for the night. The length of these periods varies with season and is computed by the model from the day length, assuming that the highest light intensity is exactly at noon. The results are computed for two different diurnal production distributions. In distribution one the production is symmetrical during the day, in distribution two it is assumed that approximately 75 percent of the daily production takes place in the morning, with a peak between 10 and 12 o'clock.

An example of the output to file FT18F001 is shown in Figure 8.9 on page 98.

Note: option 'PRODUC' is a prerequisite to option 'DIURNAL'. If 'PRODUC' has not been activated explicitly, the program will automatically put it 'on'.
Diurnal production and respiration rates at equilibrium. All data are in mg O2/m^3 / hour except "Night", which is in mg O2/m^3 / night, and "Total", which is in mg O2/m^3 / day. Two different diurnal patterns are assumed:
1. Distribution 1 assumes an equal production in morning and afternoon.
2. Distribution 2 assumes that 75% of gross production and 25% of photorespiration takes place in the morning but the reverse in the afternoon.

<table>
<thead>
<tr>
<th>Date: WK 4</th>
<th>Time:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.90</td>
</tr>
<tr>
<td></td>
<td>9.78</td>
</tr>
<tr>
<td></td>
<td>11.26</td>
</tr>
<tr>
<td></td>
<td>12.00</td>
</tr>
<tr>
<td></td>
<td>12.74</td>
</tr>
<tr>
<td></td>
<td>14.22</td>
</tr>
<tr>
<td></td>
<td>16.10</td>
</tr>
<tr>
<td>Distribution 1</td>
<td></td>
</tr>
<tr>
<td>Production</td>
<td>114.3</td>
</tr>
<tr>
<td>Respiration</td>
<td>4.3</td>
</tr>
<tr>
<td>Net-Production</td>
<td>110.0</td>
</tr>
<tr>
<td>Distribution 2</td>
<td></td>
</tr>
<tr>
<td>Production</td>
<td>259.4</td>
</tr>
<tr>
<td>Respiration</td>
<td>0.8</td>
</tr>
<tr>
<td>Net-Production</td>
<td>258.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date: WK 8</th>
<th>Time:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.20</td>
</tr>
<tr>
<td></td>
<td>9.41</td>
</tr>
<tr>
<td></td>
<td>11.14</td>
</tr>
<tr>
<td></td>
<td>12.00</td>
</tr>
<tr>
<td></td>
<td>12.86</td>
</tr>
<tr>
<td></td>
<td>14.59</td>
</tr>
<tr>
<td></td>
<td>16.80</td>
</tr>
<tr>
<td>Distribution 1</td>
<td></td>
</tr>
<tr>
<td>Production</td>
<td>236.8</td>
</tr>
<tr>
<td>Respiration</td>
<td>26.5</td>
</tr>
<tr>
<td>Net-Production</td>
<td>210.3</td>
</tr>
<tr>
<td>Distribution 2</td>
<td></td>
</tr>
<tr>
<td>Production</td>
<td>537.3</td>
</tr>
<tr>
<td>Respiration</td>
<td>4.6</td>
</tr>
<tr>
<td>Net-Production</td>
<td>532.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date: WK 12</th>
<th>Time:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6.32</td>
</tr>
<tr>
<td></td>
<td>8.93</td>
</tr>
<tr>
<td></td>
<td>10.98</td>
</tr>
<tr>
<td></td>
<td>12.00</td>
</tr>
<tr>
<td></td>
<td>13.02</td>
</tr>
<tr>
<td></td>
<td>15.07</td>
</tr>
<tr>
<td></td>
<td>17.68</td>
</tr>
<tr>
<td>Distribution 1</td>
<td></td>
</tr>
<tr>
<td>Production</td>
<td>225.2</td>
</tr>
<tr>
<td>Respiration</td>
<td>19.9</td>
</tr>
<tr>
<td>Net-Production</td>
<td>205.3</td>
</tr>
<tr>
<td>Distribution 2</td>
<td></td>
</tr>
<tr>
<td>Production</td>
<td>511.0</td>
</tr>
<tr>
<td>Respiration</td>
<td>3.5</td>
</tr>
<tr>
<td>Net-Production</td>
<td>507.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date: WK 16</th>
<th>Time:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5.37</td>
</tr>
<tr>
<td></td>
<td>8.42</td>
</tr>
<tr>
<td></td>
<td>10.81</td>
</tr>
<tr>
<td></td>
<td>12.00</td>
</tr>
<tr>
<td></td>
<td>13.19</td>
</tr>
<tr>
<td></td>
<td>15.58</td>
</tr>
<tr>
<td></td>
<td>18.63</td>
</tr>
<tr>
<td>Distribution 1</td>
<td></td>
</tr>
<tr>
<td>Production</td>
<td>168.9</td>
</tr>
<tr>
<td>Respiration</td>
<td>6.4</td>
</tr>
<tr>
<td>Net-Production</td>
<td>162.6</td>
</tr>
<tr>
<td>Distribution 2</td>
<td></td>
</tr>
<tr>
<td>Production</td>
<td>383.2</td>
</tr>
<tr>
<td>Respiration</td>
<td>1.1</td>
</tr>
<tr>
<td>Net-Production</td>
<td>382.1</td>
</tr>
</tbody>
</table>
8.2.5 Output for oxygen model OXYMOD (FT19F001)

The oxygen budget of a body of water is determined by more factors than consumption and production by phytoplankton alone. To investigate the entire oxygen budget, J.G.C. Smits of the Delft Hydraulics Laboratory developed a model called OXYMOD in the PAWN project reported by Los et al. [1982]. This model can communicate with BLOOM II because it can use the results of BLOOM's option 'DIURNAL' as an input. For reasons of efficiency and convenience the same output results as shown in Figure 8.9 on page 98, are, however, condensed in a second output file. This file is created when the program option 'OXMODOUT' is invoked. This file, for which an example is shown in Figure 8.10 on page 100, writes its output with exactly the same format as the input format of OXYMOD.

Note: options 'DIURNAL' and 'PRODUC' are prerequisite to option 'OXMODOUT'. If 'PRODUC' and 'DIURNAL' have not been activated explicitly, the program will automatically put them 'on'.

---

PAWN stands for 'Policy Analysis of Water Management for the Netherlands'. This project was executed by the Rand Corporation and the Delft Hydraulics Laboratory for the Dutch Rijkswaterstaat.
Figure 8.10 Example of output file FT19F001. This file shows the oxygen production and consumption of phytoplankton during six daily periods and during the night written in a special format to be used by the oxygen model OXY-MOD. See also Figure 8.9 on page 98.

<table>
<thead>
<tr>
<th>Week</th>
<th>Dayperiod</th>
<th>0.230</th>
<th>0.410</th>
<th>0.500</th>
</tr>
</thead>
<tbody>
<tr>
<td>WK4</td>
<td>8.20</td>
<td>110.00</td>
<td>160.03</td>
<td>287.87</td>
</tr>
<tr>
<td>WK5</td>
<td>8.20</td>
<td>258.66</td>
<td>253.22</td>
<td>218.33</td>
</tr>
<tr>
<td>WK6</td>
<td>9.59</td>
<td>210.34</td>
<td>313.96</td>
<td>578.76</td>
</tr>
<tr>
<td>WK7</td>
<td>9.59</td>
<td>532.72</td>
<td>516.58</td>
<td>434.13</td>
</tr>
<tr>
<td>WK10</td>
<td>11.36</td>
<td>205.30</td>
<td>303.84</td>
<td>555.66</td>
</tr>
<tr>
<td>WK11</td>
<td>11.36</td>
<td>507.54</td>
<td>492.35</td>
<td>418.29</td>
</tr>
<tr>
<td>WK12</td>
<td>13.27</td>
<td>162.56</td>
<td>236.46</td>
<td>425.33</td>
</tr>
<tr>
<td>WK13</td>
<td>13.27</td>
<td>382.14</td>
<td>374.12</td>
<td>322.59</td>
</tr>
<tr>
<td>WK14</td>
<td>15.02</td>
<td>242.48</td>
<td>355.96</td>
<td>645.98</td>
</tr>
<tr>
<td>WK15</td>
<td>15.02</td>
<td>585.55</td>
<td>570.44</td>
<td>487.98</td>
</tr>
<tr>
<td>WK16</td>
<td>16.29</td>
<td>203.01</td>
<td>341.35</td>
<td>694.89</td>
</tr>
<tr>
<td>WK17</td>
<td>16.29</td>
<td>697.71</td>
<td>642.93</td>
<td>499.20</td>
</tr>
<tr>
<td>WK18</td>
<td>16.61</td>
<td>267.06</td>
<td>389.86</td>
<td>703.67</td>
</tr>
<tr>
<td>WK19</td>
<td>16.61</td>
<td>634.41</td>
<td>619.90</td>
<td>532.86</td>
</tr>
<tr>
<td>WK20</td>
<td>15.73</td>
<td>231.59</td>
<td>336.76</td>
<td>605.52</td>
</tr>
<tr>
<td>WK21</td>
<td>15.73</td>
<td>543.83</td>
<td>532.54</td>
<td>459.32</td>
</tr>
<tr>
<td>WK22</td>
<td>14.01</td>
<td>296.05</td>
<td>431.31</td>
<td>776.99</td>
</tr>
<tr>
<td>WK23</td>
<td>14.01</td>
<td>699.17</td>
<td>683.92</td>
<td>588.90</td>
</tr>
<tr>
<td>WK24</td>
<td>12.19</td>
<td>172.96</td>
<td>250.10</td>
<td>447.22</td>
</tr>
<tr>
<td>WK25</td>
<td>12.19</td>
<td>399.41</td>
<td>392.35</td>
<td>340.10</td>
</tr>
<tr>
<td>WK26</td>
<td>10.39</td>
<td>265.09</td>
<td>394.68</td>
<td>725.83</td>
</tr>
<tr>
<td>WK27</td>
<td>10.39</td>
<td>666.56</td>
<td>644.69</td>
<td>545.02</td>
</tr>
<tr>
<td>WK28</td>
<td>8.72</td>
<td>217.68</td>
<td>317.15</td>
<td>571.34</td>
</tr>
<tr>
<td>WK29</td>
<td>8.72</td>
<td>514.13</td>
<td>502.91</td>
<td>433.03</td>
</tr>
<tr>
<td>WK30</td>
<td>7.84</td>
<td>299.58</td>
<td>457.01</td>
<td>859.33</td>
</tr>
<tr>
<td>WK31</td>
<td>7.84</td>
<td>805.91</td>
<td>770.49</td>
<td>638.92</td>
</tr>
</tbody>
</table>

The outputs of BLOOM II

100
8.2.6 Live and dead phytoplankton pools (FT20F001)

The partitioning of biomass between live and dead phytoplankton at equilibrium can also be computed with BLOOM II. For this computation the model first converts the biomass of each type from mg dry weight per m$^3$ into mg C per m$^3$. Second it computes the ratio between the pool sizes of detritus and of living phytoplankton at steady state based upon Eq. (1.4) for nutrients. Multiplication of the live phytoplankton biomass with this ratio gives the detritus pool size in mg C per m$^3$.

From the detritus pool the model also computes the sedimentation rate in mg C per m$^3$ per day. In the model this rate is usually zero, however, for reasons explained in 'Nutrient mass-balances' on page 4.

This file is created when the program option 'CARBON' is invoked. An example of this file is shown in Figure 8.11 on page 102.
Figure 8.11 Example of output file FT20F001. This file shows the phytoplankton and detritus pool sizes in mg C per m³ at equilibrium, and the sedimentation rate in mg C per m³ per day.

<table>
<thead>
<tr>
<th>Date</th>
<th>Temp</th>
<th>Living Algae Mg C/m³</th>
<th>Dead Algae Mg C/m³</th>
<th>Sedimentation Rate Mg C/m³/Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>NK4</td>
<td>2.0</td>
<td>56.5</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td>NK8</td>
<td>4.5</td>
<td>56.5</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td>NK12</td>
<td>7.1</td>
<td>38.8</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td>NK16</td>
<td>11.2</td>
<td>1.62</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td>NK20</td>
<td>15.5</td>
<td>1.62</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td>NK24</td>
<td>19.8</td>
<td>1.62</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td>NK28</td>
<td>18.6</td>
<td>1.62</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td>NK32</td>
<td>18.6</td>
<td>1.62</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td>NK36</td>
<td>14.9</td>
<td>1.62</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td>NK40</td>
<td>14.9</td>
<td>1.62</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td>NK44</td>
<td>12.8</td>
<td>1.62</td>
<td>1.9</td>
<td>0.0</td>
</tr>
<tr>
<td>NK48</td>
<td>7.7</td>
<td>1.62</td>
<td>1.9</td>
<td>0.0</td>
</tr>
</tbody>
</table>

The outputs of BLOOM II
8.2.7 File used by the editor (FT31F001)

When the user requests usage of the system editor to modify coefficient values or the run title (See 'Control of model coefficients' on page 67), BLOOM II creates an additional output file FT31F001 which is written to disk under the name 'DUMMY DATA'. This file is subsequently used by the editor. When the edit session is terminated, BLOOM II reads this file replacing all former information by the current contents of 'DUMMY DATA'.

Note: if a file 'DUMMY DATA' already exists from an earlier job, or from an earlier part of the session, it is replaced.

8.2.8 Print-plot of results (FT45F001)

For graphical output of a number of variables a set of print-plot routines was incorporated in BLOOM II. The print-plot option is invoked by the 'PRTPLOT' command. It creates the following plots, each with the time on the horizontal axis and the concentration of a variable on the vertical axis:

1. Computed and observed total chlorophyll.
2. Planktonic and dissolved fraction of each of the nutrients (normally nitrogen, phosphorus, silicon).

Hence usually four plots are produced. An example of the total chlorophyll plot is shown in Figure 8.12 on page 104.

Notes:

1. As mentioned earlier, the plot file has no carriage control, but a fixed page length of 62 lines. It should be printed consequently.
2. To simplify communication between BLOOM II and the plot subroutines, BLOOM II writes part of the information required by the plot routines to disk files which are subsequently read by the plot routines. The plot specifications are written to FT41F001 and FT42F001, and the actual data are written to file FT43F001. This file contains a (slightly reformatted) copy of the information written to FT14F001 during the last run. In other words file FT43F001 is replaced between the separate runs.
3. The symbols used in the plots are set by BLOOM II.
4. The plot routines can also be used as a stand-alone program to create print-plots (or graphical plots) of existing output files. However, in most cases we use SAS routines for graphical plots because these offer greater flexibility.

Note: there is a second program option, 'SCRPlot' to create a printplot of computed versus measured total chlorophyll. This option is designed for interactive runs, but can be used in batch jobs as well. In this case the plot is only 24 lines long and 80 columns wide so that it fits onto an ordinary terminal screen. This print output is written to the (standard) output files FT10F001 (interactive job) or FT21F001 (batch job).
Figure 8.12 Example of output file FT45F001. This file shows a print-plot of computed and measured total chlorophyll versus time. The nutrient plots which are not shown here have a similar appearance.
This appendix contains all help information on program options which is available in the BLOOM II program. To display (or print) this information, the user must activate the 'Option command mode' and enter 'HELP' to put the program into a special state in which (only) help information is available about program options. The user is notified by the message:

Help Option cmd:

Each time the user enters the name of an option, the program responds with the short description of that option included in here. To return to the primary 'Option cmd:' status the user must enter 'END'.

Notes:

1. The program instructions to invoke the following information are not shown here.
2. The text has been slightly reformatted for cosmetic purposes.

Option DUMP writes a detailed output to unit 6 containing information for all extinction intervals, for each iteration and time-step.

WARNING DUMP writes many output lines and should ONLY be used in an interactive run for a limited number of periods (about 5 or less).

Option MAXGRA calculates the highest feasible grazing pressure of edible species under the prevailing conditions. It finds the grazing rate constant for which the net-production is zero.

WARNING MAXGRA writes its output to a separate output unit. To avoid intermixing with normal output to unit 10, you better not send this output to the terminal.

Option PRODUC calculates production, respiration mortality, grazing and flushing rates for each phytoplankton species in each period.

WARNING PRODUC writes its output to a separate output unit. To avoid intermixing with normal output to unit 10, you better not send this output to the terminal.

Option DIURNAL calculates the diurnal production and respiration rates for each time-period.

WARNING DIURNAL writes its output to a separate output unit. To avoid intermixing with normal output to unit 10, you better not send this output to the terminal.
Option CARBON calculates the concentrations of living and
dead algae, and the sedimentation rate in each time-period.

WARNING CARBON writes its output to a separate output unit.
To avoid intermixing with normal output to unit 10,
you better not send this output to the terminal.

Option OXMODOUT rewrites the output of 'DIURNAL' into a
specially formatted dataset for the oxygen model 'OXYMOD'.

WARNING OXMODOUT writes its output to a separate output unit.
To avoid intermixing with normal output to unit 10,
you better not send this output to the terminal.

Option DYNADEAD calculates potential phytoplankton blooms
using a dynamic computation scheme for the detritus pools.

Option DYNAEEXT calculates potential phytoplankton blooms
using the reduced total extinction of the previous time-step
in the energy constraints for all but those species,
who were in the bloom at the previous time-step.

Option INTERACT tells the program that you want to make an
interactive run. It is the LAST option read from unit 5
and may not be reset during an interactive session.

A copy of the summarized output of the program
written to the terminal will be written to unit 21.

Option ? gives you some information
how the Option mode of the program works.

Option HELP gives you the detailed information
you are presently watching.
To return to Option cmd. mode enter 'END'.

Option RESET gives you the possibility to put one or more
options 'OFF' that had been turned 'ON' previously.

ALL operates under 'RESET' and resets ALL options
to their initial default values.

Option RUN tells the program to execute and solve your
problem. This command should ALWAYS be followed by 'CONTINUE'
OR 'STOP'.

Option CONTINUE informs the program that you want to continue
this session after it has solved the currently specified
bloom problem.

Appendix 1: Help for program options 106
Option STOP informs the program that you want to quit after it has solved the currently specified bloom problem.

Command PARAM returns the program control to its Parameter command mode to modify input values.

Option GROCHECK calculates steady state phytoplankton blooms with additional constraints on growth rates, determined at the start of the time-step. The initial value equals the biomass level at the previous time-step, or a constant baselevel, or a fraction of the equilibrium value.

Command CMS allows you to enter all CMS and CP commands valid under CMS SUBSET.

Option PRTPLOT creates a print-plot of predicted and observed chlorophyll and of planktonic and slack nutrients. If several subsequent runs are made, plots will be produced for each of them. However, if you want to change the title, or any of the upper limits, you should put this option 'ON' again to enter new title and limits.

Option DOMINANC creates a table with the total biomass and the relative species composition. It also computes the average total dry weight and chlorophyll concentration.

Option WHICH? gives you the names of all options which have been put 'ON' or 'OFF'.

Option PFKEYS resets the normal settings of the PF-keys to frequently used BLOOM commands.

Option SCR PLOT displays a plot of predicted and observed chlorophyll when a run is finished.
10.0 APPENDIX 2: HELP FOR MODEL COEFFICIENTS

This appendix contains help information on model coefficients. It is displayed (or printed) when the user activates the 'Parameter command mode' and enters '?'. After the command is executed, the parameter command mode is automatically restored. In an interactive run the user is notified by the message:

Parameter cmd:

Note: the text has been slightly reformatted for cosmetic purposes.

Begin Sample Session

Parameter cmd:
?
You are in parameter command mode; select one of the following parameters:
TEMPTURE
SOLARRAD
BACKEXT
DEPTH
NUTRCNC
NUTRMINH
CMS
MORTAR
SEDIMENT
FLUSH
ORG-MINE
CHL-MINE
AUTOLYSE
STOCHIOM
GROWTH
TITLE
PERIODS
CALCULAT
PRINT
ZOOPLANK

Or exit to option command mode specifying 'OPTION' command.
For most parameters the modification formula is of the type:

Multiplier * Present value + Increment = New value

To show column numbers, you may enter 'COLS'.
To show all current parameter settings, you may enter 'PRINT'.

Parameter cmd:

End Sample Session

Appendix 2: Help for model coefficients 108
This appendix contains help information on the stochiometry of phytoplankton types. It is displayed (or printed) when the user activates the 'Stochiometry command mode' and enters '?'. After the command is executed, the stochiometry command mode is automatically restored. In an interactive run the user is notified by the message:

Stochiometry cmd:

<table>
<thead>
<tr>
<th>Begin Sample Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stochiometry cmd: ?</td>
</tr>
<tr>
<td>You are in Stochiometry command mode.</td>
</tr>
<tr>
<td>Select one of the following specific inputs:</td>
</tr>
<tr>
<td>SPECEXT</td>
</tr>
<tr>
<td>NUTCoeff</td>
</tr>
<tr>
<td>CHLTocar</td>
</tr>
<tr>
<td>CARTODRY</td>
</tr>
<tr>
<td>EDIT</td>
</tr>
<tr>
<td>Then specify one of the following species names:</td>
</tr>
<tr>
<td>DIATOMS</td>
</tr>
<tr>
<td>FLAGELAT</td>
</tr>
<tr>
<td>GREENS</td>
</tr>
<tr>
<td>APHANIZO</td>
</tr>
<tr>
<td>MICROCYS</td>
</tr>
<tr>
<td>OSCILAT</td>
</tr>
<tr>
<td>If several types are in the same species, add the relative type number.</td>
</tr>
<tr>
<td>Or exit to parameter command mode entering 'END'.</td>
</tr>
<tr>
<td>You may enter 'COLS' to show column numbers</td>
</tr>
<tr>
<td>or use 'PRINT' command to show present dataset.</td>
</tr>
<tr>
<td>A global change using the editor can be made with the 'EDIT' command.</td>
</tr>
</tbody>
</table>

Stochiometry cmd:

| End Sample Session |
12.0 APPENDIX 4: HELP FOR GROWTH COEFFICIENTS

This appendix contains help information on the growth coefficients of phytoplankton types. 'Growth command mode' and enters '?' . When the command is executed, the growth command mode is automatically restored. In an interactive run the user is notified by the message:

Growth cmd:

<table>
<thead>
<tr>
<th>Begin Sample Session</th>
</tr>
</thead>
</table>

Growth cmd:

? 
You are in Growth command mode.
Select on of the following specific inputs:
PMAX
MORTAL
RESPIRAT
RELMIX
ZOOPOPREF
EDIT
Then specify one of the following species names:
DIATOMS
FLAGELAT
GREENS
APHANIZO
MICROCYS
OSCILAT

If several types are in the same species, add the relative type number.

Or exit to parameter command mode entering 'END'.
You may enter 'COLS' to show column numbers or use 'PRINT' command to show present dataset.
A global change using the editor can be made with the 'EDIT' command.

Growth cmd:

<table>
<thead>
<tr>
<th>End Sample Session</th>
</tr>
</thead>
</table>

Appendix 4: Help for growth coefficients
13.0 APPENDIX 5: HELP FOR ZOOPLANKTON COEFFICIENTS

This appendix contains help information on zooplankton coefficients. It is displayed (or printed) when the user activates the 'Zooplankton command mode' and enters '?'. When the command is executed, the zooplankton command mode is automatically restored. In an interactive run the user is notified by the message:

Zooplankton cmd:

<table>
<thead>
<tr>
<th>Begin Sample Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zooplankton cmd:</td>
</tr>
<tr>
<td>?</td>
</tr>
<tr>
<td>You are in zooplankton command mode.</td>
</tr>
<tr>
<td>Select one of the following characteristics:</td>
</tr>
<tr>
<td>NUTCDEFF</td>
</tr>
<tr>
<td>HALFSAT</td>
</tr>
<tr>
<td>ESCAPE</td>
</tr>
<tr>
<td>GRAZRATE</td>
</tr>
<tr>
<td>GRAZINIT</td>
</tr>
<tr>
<td>MAXITER</td>
</tr>
</tbody>
</table>

To exit to parameter command mode enter 'END'. You may enter 'COLS' to show column numbers or use 'PRINT' command to show present dataset.

Zooplankton cmd:

| End Sample Session |
APPENDIX 6: WORKSESSION OF BLOOM II

This chapter contains a (complete) example of a worksession with the BLOOM II model. The model has been loaded following the command 'BLOOM'. In this case the user makes several subsequent runs for Lake Veluwe, 1981, changing the background extinction, and the stoichiometry of one phytoplankton type. A probable error is followed by a warning message, and corrected afterwards. Three model options are put 'on' for some or all runs: 'SCR PLOT', 'DOMINANCE' and 'GRO-CHECK'. These options have been explained in 'Appendix 1: Help for program options' on page 105.

Begin Sample Session

All options have been (re)set to default values
Option INTERACT has been put on
Option cmd:

pa
Parameter cmd:

per 1 1 52 4
Bloom will be calculated for the following 1 periods:
First period: 1 Last period: 52 Increment: 4
Parameter cmd:

op
Option cmd:

do
Option DOMINANCE has been put on
Option cmd:

sc
Enter upper limit (mg /m3) for plot of CHLOROPHYLL
400
Option cmd:

ru
Option cmd:

co
Summary of solutions for this run:

<table>
<thead>
<tr>
<th>DATE LIMIT FACT</th>
<th>DIAT-</th>
<th>FLAG-</th>
<th>GREE-</th>
<th>APHA-</th>
<th>MICR-</th>
<th>OSCI-</th>
<th>TOTA-</th>
<th>CHL--</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OMS</td>
<td>ELAT</td>
<td>NS</td>
<td>NIZO</td>
<td>OCYS</td>
<td>LAT</td>
<td>L</td>
<td>PRED</td>
</tr>
<tr>
<td>4 P S</td>
<td>12.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
<td>13.0</td>
<td>129.2</td>
</tr>
<tr>
<td>4 P K</td>
<td>6.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.1</td>
<td>13.0</td>
<td>116.1</td>
</tr>
<tr>
<td>8 P K</td>
<td>9.8</td>
<td>0.0</td>
<td>0.0</td>
<td>2.8</td>
<td>0.0</td>
<td>0.0</td>
<td>12.6</td>
<td>135.6</td>
</tr>
<tr>
<td>12 P S</td>
<td>11.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>6.1</td>
<td>17.1</td>
<td>158.8</td>
</tr>
<tr>
<td>12 P S</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>17.1</td>
<td>17.1</td>
<td>136.8</td>
</tr>
<tr>
<td>16 P S</td>
<td>4.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>16.0</td>
<td>20.1</td>
<td>169.4</td>
</tr>
<tr>
<td>16 P S</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>20.1</td>
<td>20.1</td>
<td>161.1</td>
</tr>
<tr>
<td>20 N P S</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>18.2</td>
<td>18.4</td>
<td>147.4</td>
</tr>
<tr>
<td>24 N P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>15.1</td>
<td>15.1</td>
<td>120.4</td>
</tr>
<tr>
<td>28 N P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>19.3</td>
<td>19.3</td>
<td>154.3</td>
</tr>
<tr>
<td>32 N P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>20.8</td>
<td>20.8</td>
<td>166.4</td>
</tr>
<tr>
<td>36 N P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>21.0</td>
<td>21.0</td>
<td>168.3</td>
</tr>
<tr>
<td>40 N P K</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.4</td>
<td>17.3</td>
<td>19.7</td>
<td>162.8</td>
</tr>
<tr>
<td>44 N P S K</td>
<td>7.2</td>
<td>1.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>9.4</td>
<td>18.1</td>
<td>166.6</td>
</tr>
<tr>
<td>48 P S</td>
<td>10.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>6.2</td>
<td>16.7</td>
<td>154.2</td>
</tr>
<tr>
<td>52 P K</td>
<td>2.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.8</td>
<td>10.5</td>
<td>89.9</td>
</tr>
</tbody>
</table>

Summary of output for 13 periods:

<table>
<thead>
<tr>
<th>Species name</th>
<th>Total</th>
<th>Average</th>
<th>Dominance</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIATOMS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>DIATOMS</td>
<td>65855.0</td>
<td>5065.8</td>
<td>29.6 %</td>
</tr>
<tr>
<td>FLAGELAT</td>
<td>1463.9</td>
<td>112.6</td>
<td>0.7 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>2800.5</td>
<td>215.4</td>
<td>1.3 %</td>
</tr>
<tr>
<td>MICROCY S</td>
<td>2405.1</td>
<td>185.0</td>
<td>1.1 %</td>
</tr>
<tr>
<td>MICROCY S</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>MICROCY S</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>63638.9</td>
<td>4895.3</td>
<td>28.6 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>86292.4</td>
<td>6637.9</td>
<td>38.8 %</td>
</tr>
</tbody>
</table>

Average total biomass = 17.11 grams dry weight per cubic meter.
Average chlorophyll = 149.1 mg per cubic meter.
Warning message: the following species have a minimum reduced cost = 0.0 in the following period(s):

<table>
<thead>
<tr>
<th>Period</th>
<th>Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>48</td>
<td>2</td>
</tr>
<tr>
<td>52</td>
<td>15</td>
</tr>
</tbody>
</table>

There may be non-unique solutions. Details can be obtained using option 'DUMP'.

Parameter or Option?

pa
Parameter cmd:
ba
Mult. factor:
4
Add. factor:
0

*** WARNING MESSAGE ***

Unlikely multiplier 4.00000 for computation of BACKEXT

New backgr. ext. calculated as 4.00 * nominal Kb + 0.00
Parameter cmd:
ba 0 4
New backgr. ext. calculated as 0.00 * nominal Kb + 4.00
Parameter cmd:
op
Option cmd:
r
Option cmd:
co

Appendix 6: Worksession of BLOOD II
### Summary of solutions for this run:

<table>
<thead>
<tr>
<th>DATE LIMIT FACT</th>
<th>DIAT-</th>
<th>FLAG-</th>
<th>GREE-</th>
<th>APHA-</th>
<th>MICR-</th>
<th>OSCI-</th>
<th>TOTA-</th>
<th>CHL-</th>
<th>OMS</th>
<th>ELAT</th>
<th>NS</th>
<th>NIZO</th>
<th>OCYS</th>
<th>LAT</th>
<th>L</th>
<th>PRED</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 K</td>
<td>10.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>10.3</td>
<td>103.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 P</td>
<td>12.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>12.2</td>
<td>122.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 P S K</td>
<td>11.0</td>
<td>1.3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.8</td>
<td>15.1</td>
<td>149.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 P S K</td>
<td>4.1</td>
<td>3.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.5</td>
<td>15.0</td>
<td>146.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 P K</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>16.2</td>
<td>16.2</td>
<td>129.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 N P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>15.1</td>
<td>15.1</td>
<td>120.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28 N P K</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.8</td>
<td>14.3</td>
<td>17.1</td>
<td>142.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 N P K</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4.7</td>
<td>12.5</td>
<td>17.2</td>
<td>17.2</td>
<td>147.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36 N P K</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>7.5</td>
<td>7.9</td>
<td>15.4</td>
<td>137.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>13.7</td>
<td>13.7</td>
<td>183.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44 P S</td>
<td>4.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>6.2</td>
<td>11.0</td>
<td>147.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44 P</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>11.0</td>
<td>11.0</td>
<td>147.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48 K</td>
<td>8.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>8.5</td>
<td>84.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52 P</td>
<td>10.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>10.5</td>
<td>105.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Summary of output for 13 periods.

<table>
<thead>
<tr>
<th>Species name</th>
<th>Total</th>
<th>Average</th>
<th>Dominance</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIATOMS</td>
<td>4893.5</td>
<td>376.4</td>
<td>2.8 %</td>
</tr>
<tr>
<td>DIATOMS</td>
<td>56729.6</td>
<td>4363.8</td>
<td>32.0 %</td>
</tr>
<tr>
<td>FLAGELAT</td>
<td>4672.2</td>
<td>359.4</td>
<td>2.6 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>MICROCYC</td>
<td>14987.7</td>
<td>1152.9</td>
<td>8.4 %</td>
</tr>
<tr>
<td>MICROCYC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>MICROCYC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>19882.2</td>
<td>1529.4</td>
<td>11.2 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>31674.5</td>
<td>2436.5</td>
<td>17.8 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>44613.3</td>
<td>3431.8</td>
<td>25.1 %</td>
</tr>
</tbody>
</table>

Average total biomass = 13.65 grams dry weight per cubic meter.
Average chlorophyll = 132.3 mg per cubic meter.

Warning message: the following species have a minimum reduced cost = 0.0 in the following period(s):
- Period 20 Species 13 14
- Period 44 Species 1

There may be non-unique solutions.
Details can be obtained using option 'DUMP'.

Appendix 6: Works session of BLOOM II
Parameter or Option?

pa
Parameter cmd:
ba 1 0
New backgr. ext. calculated as 1.00 * nominal Kb + 0.00
Parameter cmd:
st
Stochiometry cmd:
nu
Species name:
apha
There are 3 types of species APHANIZO
Type number:
3
Nutrient name:
p
Stochiometric coeff:
0.002
New value of PHOSPHOR coefficient for type 3 in species APHANIZO = 0.002
Stochiometry cmd:
pr
Present species stochiometry:
Spec.name Spec ext NITROGEN PHOSPHOR SILICON CHLTOCAR CARTODRY
DIATOMS 0.750D-04 0.05000 0.00750 0.22000 25.0000 3.00000
DIATOMS 0.650D-04 0.05000 0.00350 0.15000 40.0000 2.50000
FLAGELAT 0.750D-04 0.10500 0.00880 0.00070 30.0000 2.50000
GREENS 0.610D-04 0.11000 0.00880 0.00070 30.0000 2.00000
GREENS 0.610D-04 0.05900 0.00460 0.00070 40.0000 2.00000
GREENS 0.610D-04 0.08000 0.00460 0.00070 40.0000 2.00000
APHANIZO 0.150D-03 0.08800 0.00500 0.00070 30.0000 2.50000
APHANIZO 0.150D-03 0.00000 0.00500 0.00070 30.0000 2.50000
APHANIZO 0.150D-03 0.06800 0.00200 0.00070 30.0000 2.50000
MICROCYS 0.120D-03 0.07500 0.01000 0.00070 40.0000 2.50000
MICROCYS 0.120D-03 0.04000 0.00800 0.00070 60.0000 2.50000
MICROCYS 0.120D-03 0.06000 0.00650 0.00070 60.0000 2.50000
OSCILAT 0.135D-03 0.07500 0.00750 0.00070 30.0000 2.50000
OSCILAT 0.135D-03 0.04000 0.00600 0.00070 50.0000 2.50000
OSCILAT 0.135D-03 0.06000 0.00350 0.00070 50.0000 2.50000

Appendix 6: Worksession of BLOOM II
YEAR 1981 VELUWEMEER--WAKWAL--Randmeren projekt. 14 JUNE 1985
Diepte 1.35m; Refraktair N = 0.75.
Tentatieve berekening.

Summary of solutions for this run:

<table>
<thead>
<tr>
<th>DATE LIMIT FACT</th>
<th>DIAT-</th>
<th>FLAG-</th>
<th>GREE-</th>
<th>APHA-</th>
<th>MICR-</th>
<th>OSCI-</th>
<th>TOTA-</th>
<th>CHL--</th>
<th>OMS</th>
<th>ELAT</th>
<th>NS</th>
<th>NIZO</th>
<th>OCYS</th>
<th>LAT</th>
<th>L</th>
<th>PRED</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 P K</td>
<td>9.5</td>
<td>0.0</td>
<td>0.0</td>
<td>6.2</td>
<td>0.0</td>
<td>0.0</td>
<td>15.7</td>
<td>177.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 N P</td>
<td>6.5</td>
<td>0.0</td>
<td>0.0</td>
<td>9.9</td>
<td>0.0</td>
<td>0.0</td>
<td>16.5</td>
<td>198.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 N P</td>
<td>10.7</td>
<td>0.0</td>
<td>0.0</td>
<td>11.2</td>
<td>0.0</td>
<td>0.0</td>
<td>21.9</td>
<td>256.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 N P S</td>
<td>4.1</td>
<td>0.0</td>
<td>0.0</td>
<td>15.8</td>
<td>0.0</td>
<td>0.0</td>
<td>4.1</td>
<td>24.0</td>
<td>284.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 N P S</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>6.7</td>
<td>0.0</td>
<td>11.8</td>
<td>18.7</td>
<td>185.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 N P S</td>
<td>1.6</td>
<td>0.0</td>
<td>0.0</td>
<td>3.5</td>
<td>0.0</td>
<td>10.3</td>
<td>15.4</td>
<td>145.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28 N P S</td>
<td>3.7</td>
<td>0.0</td>
<td>0.0</td>
<td>4.2</td>
<td>0.0</td>
<td>12.0</td>
<td>19.9</td>
<td>189.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 N P S K</td>
<td>4.1</td>
<td>0.0</td>
<td>0.0</td>
<td>3.3</td>
<td>0.0</td>
<td>14.1</td>
<td>21.4</td>
<td>197.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36 N P S K</td>
<td>4.5</td>
<td>0.0</td>
<td>0.0</td>
<td>2.2</td>
<td>0.0</td>
<td>14.9</td>
<td>21.7</td>
<td>194.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 N P S</td>
<td>5.2</td>
<td>0.0</td>
<td>0.0</td>
<td>4.6</td>
<td>0.0</td>
<td>12.6</td>
<td>22.4</td>
<td>213.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44 N P S</td>
<td>7.1</td>
<td>0.0</td>
<td>0.0</td>
<td>4.4</td>
<td>0.0</td>
<td>8.2</td>
<td>19.7</td>
<td>195.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48 P S K</td>
<td>10.5</td>
<td>0.9</td>
<td>0.0</td>
<td>7.0</td>
<td>0.0</td>
<td>0.0</td>
<td>18.3</td>
<td>209.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52 N P</td>
<td>6.6</td>
<td>0.0</td>
<td>0.0</td>
<td>6.9</td>
<td>0.0</td>
<td>0.0</td>
<td>13.5</td>
<td>158.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Summary of output for 13 periods.

<table>
<thead>
<tr>
<th>Species name</th>
<th>Total</th>
<th>Average</th>
<th>Dominance</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIATOMS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>DIATOMS</td>
<td>74337.4</td>
<td>5718.3</td>
<td>29.9 %</td>
</tr>
<tr>
<td>FLAGELAT</td>
<td>884.5</td>
<td>68.0</td>
<td>0.4 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>GREENS</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>APHANIZO</td>
<td>85942.7</td>
<td>6611.0</td>
<td>34.5 %</td>
</tr>
<tr>
<td>MICROCYC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>MICROCYC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>MICROCYC</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>84943.9</td>
<td>6534.1</td>
<td>34.1 %</td>
</tr>
<tr>
<td>OSCILAT</td>
<td>2906.1</td>
<td>223.5</td>
<td>1.2 %</td>
</tr>
</tbody>
</table>

Average total biomass = 19.15 grams dry weight per cubic meter.
Average chlorophyll = 200.3 mg per cubic meter.

Appendix 6: Worksession of BLOOM II 117
Parameter or Option?

Option cmd:

Do you want to print the output files, yes or no?

Time information this job:
Time elapsed: 483 seconds
Virtual cpu: 20.18 seconds
Total cpu: 25.34 seconds

End Sample Session
This appendix contains the subroutine calling sequence of the BLOOM II program. In addition to all (FORTRAN) subroutines, the names of the special interactive function are included.

Figure 15.1 Subroutine diagram of BLOOM II. Included are all subroutines and functions.

Level 1  Level 2  Level 3  Level 4  Level 5  Level 6

MAIN ==> INPUT1
      ==> INPUT2 ==> SPINDE
      ==> CALEND ==> CMS
      ==> INPUT3
      ==> OPTION ==> CLRSCR ==> CMS
      ==> INPTNM ==> PROMPT
          ==> STOD
          ==> STOR
          ==> STOI
      ==> INPTDT ==> GETS
          ==> PROMPT
      ==> MATCH
      ==> STOD
      ==> STOS
      ==> CMS
      ==> OPHELP ==> CLRSCR
          ==> INPTDT ==> GETS
          ==> PROMPT
      ==> MATCH
      ==> CHANGE ==> CLRSCR ==> CMS
      ==> INPTNM ==> PROMPT
          ==> STOD
          ==> STOR
          ==> STOI
      ==> INPTDT ==> GETS
          ==> PROMPT
      ==> MATCH
      ==> STOS
      ==> CMS
      ==> CALC
      ==> PROMPT
Figure 15.1 (con't)

Flow diagram of BLOOM program, showing all subroutines and functions.

Level 1  Level 2  Level 3  Level 4  Level 5  Level 6

CHANGE  ==>  CSPGRO  ==>  CLRSCR  ==>  CMS
        ==>  INPTNM  ==>  PROMPT
        ==>  STOD
        ==>  STOR
        ==>  STOI
        ==>  INPTDT  ==>  GETS
        ==>  PROMPT
        ==>  MATCH
        ==>  EDIT  ==>  CMS

        ==>  CSPSTO  ==>  CLRSCR  ==>  CMS
        ==>  INPTNM  ==>  PROMPT
        ==>  STOR
        ==>  STOD
        ==>  STOI
        ==>  INPTDT  ==>  GETS
        ==>  PROMPT
        ==>  MATCH
        ==>  EDIT  ==>  CMS

        ==>  EDIT  ==>  CMS
        ==>  CHHELP  ==>  CLRSCR  ==>  CMS
        ==>  MORESC  ==>  CMS

        ==>  CZOOPL  ==>  CLRSCR  ==>  CMS
        ==>  INPTNM  ==>  PROMPT
        ==>  STOR
        ==>  STOD
        ==>  STOI
        ==>  INPTDT  ==>  GETS
        ==>  PROMPT
        ==>  MATCH

        ==>  OPTION  ==>  CLRSCR  ==>  CMS
        ==>  INPTNM  ==>  PROMPT
        ==>  STOR
        ==>  STOD
        ==>  STOI
        ==>  INPTDT  ==>  GETS
        ==>  PROMPT
        ==>  MATCH
        ==>  STOD
        ==>  STOS
        ==>  CMS

        ==>  OPHelp  ==>  CLRSCR
        ==>  INPTDT  ==>  GETS
        ==>  PROMPT
        ==>  MATCH
Figure 15.1 (con't)

Flow diagram of BLOOM program, showing all subroutines and functions.

Level 1  Level 2  Level 3  Level 4  Level 5  Level 6

MAIN ===> RUN ===> PAGE1 ===> CMS
       ===> INTPOL
       ===> BLOOM ===> MAXPRD
              ===> NATMOR
              ===> SETMAT ===> DYEXTI
              ===> NUTFEA
              ===> CONSTR ===> EBCALC
              ===> MAXGRO ===> CONSTR ===> EBCALC
              ===> MAXGRA ===> CONSTR ===> EBCALC
              ===> SPCSD
              ===> EXCLUD
              ===> SOLVLP ===> DOSP
              ===> PRINT6
              ===> PRINT ===> HEADIN ===> STOS
              ===> PRINMA
              ===> GRAZIN
              ===> PRODUC ===> DIEL
                           ===> POOLS
              ===> NUSOL

              ===> DYNADE
              ===> DYEXTI
              ===> RECORD
              ===> DOMINA
              ===> MORESC ===> CMS
              ===> PRODUC ===> DIEL
                           ===> POOLS
              ===> PRPLOT ===> GRAPH ===> PLOTGR
### 16.0 APPENDIX 8: SUBROUTINE DESCRIPTION

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **BLOOM**  | Subroutine to set up bloom problem for one time-step and to handle the output.  
Language: FORTRAN  
System dependence: none  
Caller: RUN  
Calls: MAXPRD NATMOR SETMAT NUTFEA CONSTR MAXGRO MAXGRA  
Calls: SPCSD EXCLUD SOLVLP PRINT6 PRINT PRINMA GRAZIN  
Calls: PRODUC NUSOL  
- Call MAXPRD, NATMOR, and SETMAT to set up A-matrix, B-vectors for nutrient constraints.  
- Compute minimum efficiency requirements and call CONSTR to compute extinction roots.  
- Set up energy constraints. Optionally set growth constraints and compute maximum permissible grazing rate constants.  
- Check for valid extinction intervals.  
- Loop through extinction intervals, each time calling SOLVLP to obtain the Linear Programming solution.  
- Call output routines PRINT6, PRINT and PRINMA.  
- Determine grazing rate constant; if necessary reinitialize problem with new grazing rates.  
- Compute total extinction.  
- Write summary output.  
- Optionally call PRODUC to compute production rates.  |

| **CALEND**  | Subroutine to obtain the current date.  
Language: FORTRAN  
System dependence: Calls ASSEMBLER function(s) in INTERACT TALLIB.  
Caller: INPUT2  
Calls: CMS  
- Execute CMS command to obtain date by means of an EXEC.  
- Return output in three element array.  |

Appendix 8: Subroutine description 122
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| CHANGE | Subroutine to modify nominal coefficient values of the model. This routine is called by the 'PARAM' command of the program and terminated by the 'OPTION' command.  
Language: FORTRAN  
System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
Caller: MAIN  
Calls : CLRSCN INPTNM INPTDT MATCH STOS CMS CALC  
Calls : PROMPT CSPGRO CSPSTO EDIT CHHELP CZOOPL OPTION  
• Read input line and match with known control words. Reading is performed by integer functions INPTNM and INPTDT. Prompting messages are written by subroutine PROMPT.  
• Check for unknown command names. Upon detection, exit in a batch run. In an interactive run, try whether unknown BLOOM command is valid CMS or CP command. If not have error corrected.  
• Print available options, print current coefficient values calling subroutine CHHELP.  
• Modify coefficient values or multipliers of model coefficients such as temperature, background extinction, nutrient concentrations, or the output heading.  
• Call subroutines CSPGRO, CSPSPEC, and CZOOPL to modify respectively phytoplankton growth and stoichiometry coefficients, and zooplankton characteristics. |
| CHHELP | Subroutine to print current coefficient values.  
Language: FORTRAN  
System dependence: none.  
Caller: CHANGE  
Calls : CLRSCN MORESC  
• Print the value of all coefficient values that can be altered in subroutine CHANGE. |
| CLRSCR | Subroutine to refresh the terminal screen.  
Language: FORTRAN  
System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
Caller: OPTION CHANGE CSPGRO CSPSTO CHHELP CZOOPL  
Calls : CMS  
• Execute CMS command to refresh the terminal screen. |
| CONSTR | Subroutine to find minimum and maximum extinction roots.  
Language: FORTRAN  
System dependence: none.  
Caller: BLOOM MAXGRO MAXGRA  
Calls : EBCALC  
• Set both roots to -1.0 if energy level is too low.  
• Find the two roots. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| CSPGRO | **Subroutine to modify growth characteristics of phytoplankton types.**  
Language: FORTRAN  
System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
Caller: CHANGE  
Calls: CLRSCN INPTNM INPTDT MATCH EDIT  
- Read input line and match with known control words. Reading is performed by integer functions INPTNM and INPTDT. Prompting messages are written by subroutine PROMPT.  
- Print available options, print current coefficient values.  
- Make a global change using the system editor XEDIT. Option is only available in an interactive run under the VM/CMS operating system. Otherwise perform change per coefficient.  
- Change growth coefficient values for types. Check for errors. These are fatal in a batch job, but can be corrected in an interactive run. |
| CSPSTO | **Subroutine to modify stochiometric characteristics of phytoplankton types.**  
Language: FORTRAN  
System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
Caller: CHANGE  
Calls: CLRSCN INPTNM INPTDT MATCH EDIT  
- Read input line and match with known control words. Reading is performed by integer functions INPTNM and INPTDT. Prompting messages are written by subroutine PROMPT.  
- Print available options, print current coefficient values.  
- Make a global change using the system editor XEDIT. Option is only available in an interactive run under the VM/CMS operating system. Otherwise perform change per coefficient.  
- Change stochiometric coefficient values for types. Check for errors. These are fatal in a batch job, but can be corrected in an interactive run. |
| CZOOPL | **Subroutine to modify characteristics of zooplankton.**  
Language: FORTRAN  
System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
Caller: CHANGE  
Calls: CLRSCN INPTNM INPTDT MATCH  
- Read input line and match with known control words.  
- Change certain coefficient values of zooplankton. Check for errors. These are fatal in a batch job, but can be corrected in an interactive run.  
- Print available options, print current coefficient values. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **DIEL** | Subroutine to compute diurnal production and respiration rates.  
Language: FORTRAN  
System dependence: none  
Caller: PRODUC  
- Write heading for output unit 18.  
- Write heading for output unit 19 if option 'OXMODOUT' is 'ON'. This output is formatted for OXYMOD program.  
- Compute beginning and end of time periods.  
- Compute gross production, net production and respiration rate constants per hour for each daily period according to two distributions; one is symmetrical, one is skewed.  
- Compute total rates per day.  
- Write output to unit 18 and (optionally) to unit 19. |
| **DOMINA** | Subroutine to compute total and averaged biomasses.  
Language: FORTRAN  
System dependence: none.  
Caller: RUN  
Calls: MORESC  
- Increase counters for chlorophyll and dry weight of types between bloom computations for two time-steps.  
- For each type compute total dry weight, and average dry weight concentration.  
- For each type compute percentage of total dry weight of all phytoplankton types.  
- Compute average total dry weight concentration of all types.  
- Compute average total chlorophyll concentration of all types.  
- Print output on unit 10 and (in an interactive run) on unit 21.  
- Call MORESC to put screen into MORE... status. |
| **DOSP** | Subroutine to solve linear program by primal revised simplex method.  
Language: FORTRAN  
System dependence: none.  
Caller: SOLVLP  
- Subroutine from IBM mathematical library.  
- Subroutine described in SH12-5300-1.  
- 5736-XM7 COPYRIGHT IBM CORP. 1971. |
| **DYEXTI** | Subroutine for dynamic extinction calculation.  
Language: FORTRAN  
System dependence: none.  
Caller: SETMAT RUN  
- Initialize. If previous solution was infeasible, make limiting extinction equal to background extinction and exit.  
- Record the numbers of the species in the bloom.  
- Compute limiting extinction as the total extinction of the previous time-step minus the expected turnover due to mortality. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **DYNADE** | Subroutine for dynamic nutrient calculation.  
Language: FORTRAN  
System dependence: none.  
Caller: RUN  
* Compute detritus pool sizes according to dynamic solution of homogeneous differential equation of nutrient recycling. |
| **EBCALC** | Subroutine to perform special interpolation.  
Language: FORTRAN  
System dependence: none.  
Caller: CONSTR  
* Interpolate phytoplankton growth efficiencies. |
| **EDIT** | Subroutine to use editor to modify program inputs.  
Language: FORTRAN  
System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
Caller: CHANGE CSPGRO CSPSTO  
Calls : CMS  
* Execute CMS command XEDIT fn <ft <fm>> << profile prof> for file 'fn ft fm' using profile 'prof'. Ft, fm and prof are optional. |
| **EXCLUD** | Subroutine to determine permitted types in extinction interval.  
Language: FORTRAN  
System dependence: none.  
Caller: BLOOM  
* Check whether any type is permitted in current extinction interval.  
* Record types permitted in current extinction interval in appropriate row of A-matrix.  
* If previous (lower) extinction interval was infeasible set flag for current interval to 'infeasible', unless a type is permitted now, which was not permitted in the previous interval. |
| **GRAPH** | Subroutine to prepare input for print plot routine PLOTGR.  
Language: FORTRAN  
System dependence: none.  
Caller: PRPLOT  
Calls : PLOTGR  
* Initialize arrays and constants.  
* Read plot control parameters from disk file.  
* Read input from (temporary) output files written by subroutine BLOOM.  
* Call PLOTGR. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **GRAZIN** | Subroutine to compute grazing rate constants by zooplankton.  
                          Language: FORTRAN  
                          System dependence: none.  
                          Caller: BLOOM  
                          • Initialize counters.  
                          • Compute the total dry weight biomass of edible species.  
                          • Compute the grazing rate constant using measured zooplankton data and computed phytoplankton concentrations. |
| **HEADIN** | Subroutine to write heading for several output files.  
                          Language: FORTRAN  
                          System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
                          Caller: PRINT  
                          Calls: STOS  
                          • Write heading for outputs on units 10, 14 and 21.  
                          • Set print output array indices depending on the number of phytoplankton species, nutrients etc.  
                          • Use ASSEMBLER function STOS to split the full 8 character species names into two parts of 4 characters each. |
| **INPTDT** | Integer function to read tokenized character string.  
                          Language: FORTRAN  
                          System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
                          Caller: OPTION CHANGE CSPGRO CSPSTO CZOOPL  
                          Calls: GETS PROMPT  
                          • Read a single line with 10A8 format.  
                          • Call function GETS to obtain next token.  
                          • In an interactive run call PROMPT to write prompting messages. |
| **INPTNM** | Integer function to read and transform variables.  
                          Language: FORTRAN  
                          System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
                          Caller: OPTION CHANGE CSPGRO CSPSTO CZOOPL  
                          Calls: STOD STOI STOR PROMPT  
                          • Read a single line with 10A8 format.  
                          • Transform character string to double precision real (STOD), to real (STOR) or to integer (STOI).  
                          • In an interactive run call PROMPT to write prompting messages. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| INPUT1 | Subroutine to read input data.  
Language: FORTRAN  
System dependence: none.  
Caller: MAIN  
  - Read title for output.  
  - Read number of periods in input files.  
  - Read day lengths, mortality rates (optional), zooplankton concentrations (optional) and depths. |
| INPUT2 | Subroutine to read input data for system definition.  
Language: FORTRAN  
System dependence: none.  
Caller: MAIN  
Calls: SPINDI CALEND  
  - Read various control integers such as the number of species, the number of nutrient constraints etc.  
  - Compute dependent control integers.  
  - Read multipliers for input values of nutrients, temperature, solar intensities, background extinction and depth.  
  - Read coefficients for mineralization.  
  - Read instructions how determine the mortality rate constants.  
  - Read zooplankton coefficients.  
  - Read fraction for autolysis and sedimentation rate constant.  
  - Read stochiometry of types.  
  - Read growth coefficients of types.  
  - Call subroutine SPINDI to determine which types belong to which species.  
  - Read integrated efficiency curves for growth of phytoplankton species.  
  - Call subroutine CALEND to obtain the current date. |
| INPUT3 | Subroutine to read input data.  
Language: FORTRAN  
System dependence: none.  
Caller: MAIN  
  - Read weekly values for week name, week number, water temperature, solar intensity, chlorophyll, nutrients, and background extinction. |
| INTPOL | Subroutine to perform a linear interpolation.  
Language: FORTRAN  
System dependence: none.  
Caller: RUN  
  - Read an array of data and perform a linear interpolation between a specified begin and end point for all data satisfying some specific test value such as '0', '.', or '-1.0'.  
  - Missing values for begin and end point are taken equal to the first, respectively last value not equal to the test value.  
  - The input array is replaced by the interpolated array. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **MAIN** | Main section of BLOOM program.  
Language: FORTRAN  
System dependence: none.  
Calls: INPUT1 INPUT2 INPUT3 OPTION CHANGE RUN  
• Call the three input subroutines INPUT1, INPUT2 and INPUT3.  
• Call subroutine OPTION to read coefficients for program control.  
• Optionally call subroutine CHANGE to modify system definition.  
• Increase counter for number of runs.  
• Call subroutine RUN to start bloom computations.  
• Optionally call subroutine CHANGE to modify system definition.  
• Optionally write error message before termination of the program. |
| **MAXGRA** | Subroutine to compute maximum permissible grazing rates.  
Language: FORTRAN  
System dependence: none.  
Caller: BLOOM  
Calls: CONSTR  
• Write heading for output.  
• Compute maximum possible value for grazing rate constant of each phytoplankton type at which the net growth rate is not negative.  
• Compute the corresponding zooplankton biomass, which is required to bring about such a grazing rate.  
• Write output. |
| **MAXGRO** | Subroutine to determine the right hand sides of the growth constraints.  
Language: FORTRAN  
System dependence: none.  
Caller: BLOOM  
Calls: CONSTR  
• Check if species has non-negative net growth rate at current extinction level. If not, set base level for right hand side of growth constraint.  
• Compute at which growth efficiency the net growth rate of the species is zero, given a specific extinction level in addition to the normal factors such as temperature and solar intensity.  
• Compute the right hand side of the growth constraint as the biomass level of the previous period multiplied by the (maximum) value for the increment just determined. See also subroutine RECORD. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| MAXPRD | Subroutine to compute the maximum production and respiration rate constants of phytoplankton types.  
Language: FORTRAN  
System dependence: none.  
Caller: BLOOM  
• Compute the respiration rate constants.  
• Compute the maximum gross growth rate constants.  
• Optionally write values to output file. |
| MORESC | Subroutine to put the terminal screen into MORE... status.  
Language: FORTRAN  
System dependence: Calls ASSEMBLER function(s) in INTER-  
ACT TXTLIB.  
Caller: CHHELP RUN  
Calls: CMS  
• Execute CMS command to put the terminal screen into MORE... status. This command allows various output sections such as tables to start at the top of the screen, without automatically refreshing the previous screen. |
| NATMOR | Subroutine to determine the mortality rate constants for a time-step and to put them in the appropriate program variables.  
Language: FORTRAN  
System dependence: none.  
Caller: BLOOM  
• Find out how the mortality rate constants must be determined.  
• Set or compute mortality rate constants.  
• Optionally write output messages. |
| NUSOL | Subroutine to write a warning message for possible non-unique solutions. This subroutine is only called if option 'DUMP' is 'ON'.  
Language: FORTRAN  
System dependence: none.  
Caller: BLOOM  
• Write warning message indicating that certain types have a zero reduced cost factor in the present time-step, indicating they might be able to replace one of the types in the optimum bloom solution. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUTFEA</td>
<td>Subroutine to check feasibility of nutrient constraints. Currently this subroutine is only called when nutrients can indeed become negative as for example in a dynamic nutrient computation. See also subroutine DYNADE.</td>
</tr>
<tr>
<td></td>
<td>Language: FORTRAN</td>
</tr>
<tr>
<td></td>
<td>System dependence: none.</td>
</tr>
<tr>
<td></td>
<td>Caller: RUN BLOOM                                                                  • Check whether there are negative right hand sides for any of the nutrient constraints. If this is the case, set program flag to infeasible, unless some type does not require this particular nutrient.</td>
</tr>
<tr>
<td></td>
<td>• Replace negative value by a small positive number.</td>
</tr>
<tr>
<td>OPHELP</td>
<td>Subroutine to provide assistance on using program options.</td>
</tr>
<tr>
<td></td>
<td>Language: FORTRAN</td>
</tr>
<tr>
<td></td>
<td>System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.</td>
</tr>
<tr>
<td></td>
<td>Caller: OPTION                                                                    • Call CLRSCR to refresh the screen.</td>
</tr>
<tr>
<td></td>
<td>Calls: CLRSCR INPTDT MATCH                                                          • Call INPTDT to read and tokenize input line.</td>
</tr>
<tr>
<td></td>
<td>• Match each word with known program options and write required information to output unit.</td>
</tr>
<tr>
<td></td>
<td>• Set return code upon exit if an error is detected.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| OPTION | Subroutine to modify options of the model. This routine is called by the 'OPTION' command of the program and terminated by the 'PARAM', 'STOP', 'CONTINUE' commands.  
Language: FORTRAN  
System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
Caller: MAIN CHANGE  
Calls: CLRSCN INPTNM INPTDT MATCH STOS CMS STOD  
Calls: OPHELP  
- Set default values for program options (usually: 'OFF').  
- Read input line and match with known control words. Reading is performed by integer functions INPTNM and INPTDT. Prompting messages are written by subroutine PROMPT.  
- Check for unknown command names. Upon detection, exit in a batch run. In an interactive run, try whether unknown BLOOM command is valid CMS or CP command. If not have error corrected.  
- Set control parameters for selected options. In some cases automatically change status of prerequisite options to 'ON' as well.  
- For option 'GROCHECK' adjust array indices and get base level.  
- For option 'PRTPLOT' get title and plot boundaries, and manipulate plot input files.  
- For option 'SCRPLOT' get upper limit for chlorophyll plot. Manipulate plot input files.  
- For option 'RESET' perform the same actions as for 'OPTION', only this time change status of selected or all options to 'OFF'.  
- Call subroutine OPHELP to write help information on selected options. |
| PAGE1 | Subroutine to write first page of output.  
Language: FORTRAN  
System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
Caller: RUN  
Calls: CMS  
- Write BLOOM II logo.  
- Execute CMS command to obtain the date when the current version of the program was created.  
- Write additional information. |
| PLOTGR | Subroutine to create and write print plot of program output.  
Language: FORTRAN  
System dependence: none.  
Caller: GRAPH  
- Initialize symbols for axis.  
- Get or compute plot boundaries.  
- Write plot to output unit. |

Appendix B: Subroutine description
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| POOLS | Subroutine to compute pool sizes of living and dead phytoplankton at equilibrium and the sedimentation rate of detritus.  
Language: FORTRAN  
System dependence: Calls ASSEMBLER function(s) in INTERACT TXTLIB.  
Caller: PRODUC  
• Perform computations.  
• Write output to unit 20. |
| PRINMA | Subroutine to write detailed output for 'DUMP' option.  
Language: FORTRAN  
System dependence: none.  
Caller: BLOOM  
• Write phytoplankton type concentrations.  
• Write nutrient concentrations and slacks.  
• Write information on the number and feasibility of the extinction intervals. |
| PRINT | Subroutine to create output vector for summarized solutions. The actual write instructions are in the calling subroutine BLOOM.  
Language: FORTRAN  
System dependence: none.  
Caller: BLOOM  
Calls : HEADIN  
• If called for the first time, call HEADIN to write heading.  
• Determine limiting factors and record their names.  
• Compute and record total dry weight concentrations for each phytoplankton species.  
• Compute and record total chlorophyll concentration of phytoplankton.  
• In a run with grazing write output for current iteration step to unit 15.  
• Record planktonic and slack fractions of nutrients. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT6</td>
<td>Subroutine to write detailed output for 'DUMP' option and to determine the extinction interval with the maximum phytoplankton biomass. Language: FORTRAN System dependence: none. Caller: BLOOM</td>
</tr>
<tr>
<td></td>
<td>• If option 'DUMP' is 'ON' write current extinction interval limits and the numbers of the types whose net growth rate is not negative.</td>
</tr>
<tr>
<td></td>
<td>• If the extinction interval is infeasible, increase the infeasibility counter by 1. If DUMP is 'ON', write error message set by the simplex subroutine DOSP. Return.</td>
</tr>
<tr>
<td></td>
<td>• If DUMP is 'ON' write solution for the current extinction interval.</td>
</tr>
<tr>
<td></td>
<td>• Compare the total biomass of the current extinction interval with the maximum obtained so far in all previous intervals. If the current maximum is lower, return. Otherwise update the maximum biomass variable, the maximum interval indicator and the array with the numbers of the types that have a zero minimum reduced cost (See also NUSOL, RUN, and SOLVLP).</td>
</tr>
<tr>
<td>PRODUC</td>
<td>Subroutine to compute production, respiration, mortality, flushing and grazing rates of the bloom under steady state conditions. Language: FORTRAN System dependence: none. Caller: BLOOM RUN Calls: DIEL POOLS</td>
</tr>
<tr>
<td></td>
<td>• Write heading for output on unit 17.</td>
</tr>
<tr>
<td></td>
<td>• Compute energy fluxes at equilibrium for each phytoplankton species.</td>
</tr>
<tr>
<td></td>
<td>• Compute totals in units mg C per m² per day.</td>
</tr>
<tr>
<td></td>
<td>• Convert totals to units mg O₂ per m² per day.</td>
</tr>
<tr>
<td></td>
<td>• Call subroutine DIEL if option 'DIURNAL' is 'ON'.</td>
</tr>
<tr>
<td></td>
<td>• Call subroutine POOLS if option 'CARBON' is 'ON'.</td>
</tr>
<tr>
<td></td>
<td>• Increase counters for yearly totals of the gross production rate.</td>
</tr>
<tr>
<td></td>
<td>• Write totals when the bloom computation for the final period is completed.</td>
</tr>
<tr>
<td>PROMPT</td>
<td>Subroutine to write prompting messages for program options and parameter change commands. These messages are intended to be written to a terminal and are only produced in an interactive run (option INTERACT is 'ON'). Language: FORTRAN System dependence: none. Caller: INPTDT INPTNM CHANGE</td>
</tr>
<tr>
<td></td>
<td>• Compute index number for message label.</td>
</tr>
<tr>
<td></td>
<td>• Write the appropriate message and return.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| PRPLOT | Subroutine to set up call for the print plot routine GRAPH.  
Language: FORTRAN  
System dependence: none.  
Caller: RUN  
Calls: GRAPH  
• Rewind the two files which will be read by GRAPH.  
• Put appropriate control words in array to invoke the options of GRAPH as required in the BLOOM II environment.  
• Call GRAPH. |
| RECORD | Subroutine to establish the initial biomass levels for the species. These values are used by subroutine MAXGRO to determine the right hand sides of the growth constraints. RECORD is only called if option 'GROCHECK' is 'ON'.  
Language: FORTRAN  
System dependence: none.  
Caller: RUN  
• If infeasible, use base level for all species and return.  
• If option 'FRACTION' was selected for 'GROCHECK', compute base level as a constant fraction of the steady state solution of each species. The maximum of the latter and the biomass of the previous time-step is used by MAXGRO.  
• If option 'CONSTANT' was selected for 'GROCHECK', base level is a constant for all species. The maximum of the latter and the biomass of the previous time-step is used by MAXGRO. |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| RUN  | Subroutine to determine the input for subroutine BLOOM, to perform several actions between the time-steps, and to perform final actions at the end of a job. Language: FORTRAN System dependence: none. Caller: MAIN Calls: PAGE1 INTPOL BLOOM DYNADY DYEHTI RECORD DOMINA Calls: MORESC PRODUC PRTPLOT  
- Call subroutine PAGE1 to write title page for output.  
- If the input for chlorophyll contains missing values, call INTPOL to perform a linear interpolation. Note: interpolated values are only used when the time-step size of the model is not equal to 1.  
- If option 'PRTPLOT' or 'SCRPLOT' is 'ON', rewind plot files.  
- If option 'GROCHECK' is 'ON', initiate base levels for all species to some large number. As a result the program starts with the steady state solution.  
- Initiate counters for several subroutines such as SETMAT and PRINT.  
- Start main computational loop of the program for each requested period.  
- Write several (optional) output messages.  
- Initiate counters for total biomass and total chlorophyll to be used by subroutine DOMINA.  
- Average the inputs for constraints over time. Note: with the nominal time-step of 1 this simply means that the weekly inputs are used.  
- Convert nutrient levels from mg/l to mg/m³.  
- Establish values for nutrients, background extinction, water temperature, solar intensity, and depth based upon their inputs, and (optionally) multipliers and increments. See also subroutines INPUT2 and CHANGE.  
- Call subroutine BLOOM to compute phytoplankton bloom level for each time-step.  
- Record types with zero reduced cost for each period. See also subroutines NUSOL, PRINT6 and SOLVLP.  
- If corresponding program options are 'ON' call subroutines DYNADY, DYEHTI, RECORD, and DOMINA.  
- Close loop for time-steps; perform some final actions before returning to MAIN.  
- Call DOMINA to write summary table if option 'DOMINANCE' is 'ON'.  
- Write warning message if potential degenerate solutions have been detected in any period.  
- Call PRODUC to compute total yearly production if option 'PRODUC' is 'ON'.  
- Call PRTPLOT to create a plots of several outputs if option 'PRTPLOT' or 'SCRPLOT' are 'ON'. |
### SETMAT
Subroutine to set-up matrix and vectors for the linear program.

- **Language**: FORTRAN
- **System dependence**: none.
- **Caller**: BLOOM
- **Calls**: DYEXTI
  - If called for the first time, set objective function (C-vector); perform initial call to subroutine DYEXTI if option 'DYNAEXT' is 'ON'; initiate various constants.
  - Determine adjusted solar radiation level. Correct for surface reflectance, temperature, and unit conversion.
  - Set coefficients for the A-matrix of the linear program.
  - Compute right hand sides of the nutrient constraints. Correct for nutrients tied up in zooplankton.
- **Note**: the right hand sides of the energy constraints are different for each extinction interval. These are set in subroutine BLOOM.

### SOLVLP
Subroutine to set up the call for the simplex subroutine DOSP and to handle the returning variables. To speed up the solution algorithm of the model the restart option of DOSP is frequently used.

- **Language**: FORTRAN
- **System dependence**: none.
- **Caller**: BLOOM
- **Calls**: DOSP
  - Determine array indices and control variables when SOLVLP is called for the first time. Add rows for growth constraint to the A-matrix when option 'GRO-CHECK' is 'ON'.
  - Reset restart option for DOSP to 'OFF' if SOLVLP is called for the first time in a particular time-step. This means the algorithm starts with only 'slack' variables in the basis.
  - Determine types permitted in the current extinction interval. For each type not permitted,
    1. Change coefficient of objective function so that this type is minimized in the algorithm in stead of maximized.
    2. Remove it from the basis (the initial solution of the algorithm) if it was included in the optimal solution of the previous extinction interval.
  - Call subroutine DOSP to solve the linear program.
  - Put results in appropriate form: set return code, total maximum biomass, X-vector (output of variables), and indices for types with a reduced cost coefficient of zero.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPCSD</td>
<td>Subroutine to order extinction roots and to determine which types have a non-negative net growth rate in each interval.</td>
</tr>
<tr>
<td></td>
<td>Language: FORTRAN</td>
</tr>
<tr>
<td></td>
<td>System dependence: none.</td>
</tr>
<tr>
<td></td>
<td>Caller: BLOOM</td>
</tr>
<tr>
<td></td>
<td>• Initialize various arrays.</td>
</tr>
<tr>
<td></td>
<td>• Determine the extinction roots of each individual type.</td>
</tr>
<tr>
<td></td>
<td>• If a dynamic extinction computation is requested (option 'DYNAEEXT' is 'ON'), use two different base extinction levels. Use background extinction for types present at previous time-step, but use value between background extinction and total extinction for all other types. See also subroutine DYTEXTI.</td>
</tr>
<tr>
<td></td>
<td>• Sort the array with all extinction roots.</td>
</tr>
<tr>
<td></td>
<td>• Check if there is at least one feasible interval.</td>
</tr>
<tr>
<td></td>
<td>• Eliminate intervals with two negative roots, hence the first positive number in the sorted vector becomes the upper bound of the first extinction interval.</td>
</tr>
<tr>
<td></td>
<td>• Eliminate duplicate intervals.</td>
</tr>
<tr>
<td></td>
<td>• For each extinction interval determine which types have a non-negative net growth rate.</td>
</tr>
</tbody>
</table>

| SPINDI | Subroutine to determine an array containing the numbers of the first and last type in each species.                                         |
|        | Language: FORTRAN                                                                                                                             |
|        | System dependence: none.                                                                                                                     |
|        | Caller: INPUT2                                                                                                                               |
|        |  • Determine the numbers of the first and of the last type in each phytoplankton species.                                                   |
|        |  • Check whether the number of different species names agrees with the number of species previously read by the program.                    |
17.0 REFERENCES

1. BIGELOW, J.H., J.G. BOLTEN, AND J.C. DE HAVEN,
   Protecting an Estuary from Floods—A Policy Analysis of the
   Oosterschelde: Vol. IV, Assessment of Algae Blooms, A potential
   Ecological Disturbance,
   The Rand Corporation, R-2121/4-NETH, April 1977.
2. BOUGIS, P.,
   Ecologie du Plancton Marin, I. Le Phytoplankton,
3. DANZIG, G.B.,
   Linear programming and extensions,
4. EPPLEY, RICHARD W.,
   Temperature and phytoplankton growth in the sea,
5. FOY, R.H., C.E. GIBSON, AND R.V. SMITH,
   The influence of daylength, light intensity and temperature on the
   growth rates of planktonic Blue Green algae,
6. GOLDMAN, J.C. AND E.J. CARPENTER,
   A kinetic approach to the effect of temperature on the algal
   growth
7. GORHAM, P.R.,
   Toxic Algae, in: D.F. Jackson, (ed.), Algae and Man,
8. HARRIS, G.P.,
   Photosynthesis, Productivity and Growth: The Physiological
   Ecology of Phytoplankton,
   1-171.
9. JORGENSEN, E.G.,
   The adaptation of plankton algae; IV Light adaptation in differ-
   ent algal species,
10. LIERE, L. VAN,
    On Oscillatoria agardhii gomont,
11. LOS, F.J.,
    Mathematical simulation of algae blooms by the model BLOOM II,
    Delft Hydraulics Laboratory, R1310-7, July 1982.
12. LOS, F.J.,
    Mathematical simulation of algae blooms by the model BLOOM II.
    Part 2. Modifications and extensions of the model.
    Delft Hydraulics Laboratory, R1310-7, in prep.
13. LOS, F.J., N.M. DE ROOIJ, J.G.C. SMITS, AND J.H. BIGELOW,
    Policy Analysis of Water Management for the Netherlands: VOL.
    VI, Design of Eutrophication Control Strategies,
    Rand Corp. and Delft Hydraulics Laboratory, N-1500/6-NETH,
    February 1982.
14. MORTON, STEPHEN D., AND TERRENCE H. LEE,
    Algal blooms—possible effects of iron,
    673-674.
15. REYNOLDS, C.S.,
    The ecology of freshwater phytoplankton,
16. PARSONS, T.R., AND M. TAKAHASHI,
    Biological Oceanographical Processes,
17. ROOIJ, N.M. DE, 
A chemical model to describe nutrient dynamics in lakes, 
In: Barica, J., and L.R. Mur (eds.), Hypertrophic ecosystems, 
Developments in Hydrobiology 2, 

18. ROOIJ, N.M. DE, 
Manual CHARON, 
Delft Hydraulics Laboratory, R1310, in prep.

EN R. KLOMP, 
Water quality modelling for Lake IJssel; impact of various san-
itation measures on eutrophication of Lake IJssel, 
Delft Hydraulics Laboratory, R1552, June 1982 (in Dutch).

20. ROOIJ, N.M. DE, M.F.X. VEUL EN B. DE GROOT, 
Computation of the future water quality of a fresh lake Grevel-
ingen, the Volkerak and Zoommeer, 
Delft Hydraulics Laboratory, R1644 / R1645, December 1982 (in 
Dutch).

21. RYTHER, J.H., 
Photosynthesis in the Ocean as a Function of Light Intensity, 

22. SAKSHAUG, E., 
Limiting nutrients and maximum growth rates for diatoms in Nar-
ranganset Bay, 

23. SCAVIA D., AND B.J. EADIE, 
The use of measurable coefficients in process formulations - 
zooplankton grazing, 

24. SMITS, J.G.C., F.J. LOS EN N.M. DE ROOIJ, 
Evaluation of the models CHARON, BLOOM II and CHARON-BLOOM II 
by application to the model reservoirs in the 'Grote Rug', 
Delft Hydraulics Laboratory, S594, 1984 (in Dutch).

25. SMITS, J.G.C., J.C. STANS, F.J. LOS EN N.M. DE ROOIJ, 
Water quality modelling of Rijnland, 
Delft Hydraulics Laboratory, R1651, 1985 (in Dutch).

26. STRICKLAND, J.D.H., 
Measuring the Production of Marine Phytoplankton, 
Fisheries Research Board of Canada, Bullitin No. 122, Ottawa, 
1960.
nutrient cycle 5
purpose 2
selection of species 3-4
steady state assumption 21-23
BLOOM II program computations 76-80
growth constraints 76-78
LP solution, general 78
LP solution, saving CPU 78-80
computer costs 76
CPU requirement 76
file definitions 47-49
input, optional 63-75
general 63
instruction layout 64-65
model coefficients 67-76
model options 66-67
processing, batch and interactive 63-64
input, standard 51-62
autolysis 56
autolysis, standard 51-62
background extinction 60
background extinction, control 53
chlorophyll a 60
chlorophyll a, missing values 60
control indices 52
day length 51
depth 51
depth, control 54
file FT05F001 51-59
file FT11F001 59-62
file FT12F001 62
general 51
mineralization, chlorophyll a 54-55
mineralization, overall 54
mortality, control 55
mortality, measured 51
nutrient concentrations 60
nutrient, control 52-53
period identifier 59
period number 60
periods 52
solar radiation 60
solar radiation, control 53-54
species coefficients 56-57
species stoichiometry 56
temperature 60
temperature, control 53
title 51
zooplankton 51
zooplankton, control 55-56
job control 46-50
output, general 81
output, optional 90-104
biomass partitioning 101-103
dump 90-93
file for editor 103
fluxes, carbon and oxygen 95-96
fluxes, diurnal 02 97-98
fluxes, diurnal 02 for OXYMOD 99-100
maximum potential grazing rates 93-94
print plot 103-105
output, standard 81-89
nutrient partitioning 88-90
nutrient partitioning, dissolved fraction 88
nutrient partitioning, planktonic fraction 88
summary solution 83-86
title page 81-83
zooplankton iterations 86-88
overview 43
system dependence 45-46, 50
BROWSE command 91

CALCULAT program instruction 108
CARBON program instruction 101, 106
carriage control 81
CARTODRY program instruction 109
See also BLOOM II program, input, standard, species coefficients
cc
See carriage control
CHL-MINE program instruction 108
See also BLOOM II program, input, standard, mineralization, chlorophyll
chlorophyll 9
CHLTOCAR program instruction 109
See also BLOOM II program, input, standard, species coefficients
CMS program instruction 107, 91, 108
console file 49-50
CONTINUE program instruction 63, 67, 106

DEPTH program instruction 68, 108
See also BLOOM II program, input, standard, depth, control
detritus 6
DIURNAL program instruction 97, 67, 105
DOMINACE program instruction 83, 107
DYNADEAD program instruction 106
DYNAEXT program instruction 106

EDIT program instruction 70, 72, 109, 110
END program instruction 66, 70, 72, 74, 105
ESCAPE program instruction 111
See also BLOOM II program, input, standard, zooplankton, control
euphotic depth 9
eutrophication 1
EXEC procedure
BLMBATCH
See BLMBATCH EXEC
BLOOM
See BLOOM EXEC
general description 46
PRTBLOOM
See PRTBLOOM EXEC

FILE editor command 70, 72
FILEDEF command 47
Files
existence 49
naming conventions 48
organization 49
FLUSH program instruction 108
FULDR command 49

G

grazing 14-16
prey selection 15
rate computation 15
GRAZINIT program instruction 111
See also BLOOM II program,
input, standard, zooplank-
ton, control
GRAZRATE program instruction 111
See also BLOOM II program,
input, standard, zooplank-
ton, control
GROCHECK program instruction 107
growth 9
average 16-21
maximum 11-12
GROWTH program instruction 72, 108
See also BLOOM II program,
input, standard, species
coefficients

H

HALFSAT program instruction 111
See also BLOOM II program,
input, standard, zooplank-
ton, control
HELP program instruction 66, 105, 106

I

INTERACT program instruction 63, 106
interactive processing
See processing, batch and
interactive

J

job control 46

M

MAXGRA program instruction 16, 105
See also BLOOM II program,
output, optional, maximum
potential grazing rates
MAXITER program instruction 111
See also BLOOM II program,
input, standard, zooplank-
ton, control
menu 47
model coefficients
See BLOOM II model, input,
universal
MORTAL program instruction 68, 108
See also BLOOM II program,
input, standard, mortal-
ity, control
See also BLOOM II program,
input, standard, species
coefficients
mortality 13-14
minimum estimate 14
revised estimate 14

N

NOMINAL program instruction 52
notations iv
NUTCOEFF program instruction 109, 111
See also BLOOM II program,
input, standard, species
stochiometry
See also BLOOM II program,
input, standard, zooplank-
ton, control
NUTRCONC program instruction 108
See also BLOOM II program,
input, standard, nutrient,
control
NUTRMINE program instruction 108
OPTION program instruction 68
ORG-MINE program instruction 108
See also BLOOM II program, input, standard, mineralization, overall
OMODOUT program instruction 99, 106
OXYMOD 02 model 99, 33

PARAM program instruction 68, 67, 107
PERIODS program instruction 108
See also BLOOM II program, input, standard, periods
PFKEYS program instruction 107
photosynthesis 9
P MAX program instruction 110
See also BLOOM II program, input, standard, species coefficients
PRINT program instruction 68, 70, 72, 74, 108
processing, batch and interactive 43-45
PRODUC program instruction 95, 67, 105
programs
BLCOMBIN
See BLOCOMIN
BLEFFORM
See BLEFFPRO
BLEFFPRO
See BLEFFPRO
BLEUNFO
See BLEUNFO
BLOOM
See BLOOM II program
PRTBLOOM EXEC 81
PRTPLOT program instruction 103, 107

QUIT editor command 70, 72

RELMIX program instruction 110
See also BLOOM II program, input, standard, species coefficients
RESET program instruction 66, 106
RESPIRAT program instruction 110
See also BLOOM II program, input, standard, species coefficients
respiration 12-13
RUN program instruction 63, 67, 106

SCRPLLOT program instruction 103, 107
SEDIMENT program instruction 108
simplex
See BLOOM II model, LP solution
SOLARRAD program instruction 68
SPECEXT program instruction 109
See also BLOOM II program, input, standard, species stoichiometry
STOCHIOMETRY program instruction 70, 108
See also BLOOM II program, input, standard, species stoichiometry
STOP program instruction 63, 67, 107
system definition
See BLOOM II model, input, universal

TEMPTURE program instruction 108
See also BLOOM II program, input, standard, temperature, control terminology
See notations
TITLE program instruction 108
See also BLOOM II program, input, standard, title

TOTALRAD program instruction 54

WHICH? program instruction 66, 107

ZOOPRINTON program instruction 74, 108
ZOOPREF program instruction 110

See also BLOOM II program, input, standard, species coefficients