National Oceanography Centre, Southampton

Research & Consultancy Report No. 67

A marine model optimization test-bed for ecosystem model evaluation: MarMOT version 1.0 description and user guide

J C P Hemmings

2009

With minor corrections

August 2010

National Oceanography Centre, Southampton
University of Southampton, Waterfront Campus
European Way
Southampton
Hants SO14 3ZH UK

Author contact details:
Tel: +44 (0)23 8059 7793
Email: jch@noc.soton.ac.uk

DOCUMENT DATA SHEET

AUTHOR
HEMMINGS, J C P

PUBLICATION DATE 2009

TITLE

A marine model optimization test-bed for ecosystem model evaluation: MarMOT version 1.0 description and user guide.

REFERENCE

Southampton, UK: National Oceanography Centre, Southampton, 111pp.
(National Oceanography Centre Southampton Research and Consultancy Report, No. 67)
(Unpublished manuscript) *With minor corrections August 2010.*

ABSTRACT

In response to scientific challenges in the modelling of plankton ecosystems and their role in biogeochemical cycles, the Marine Model Optimization Test-bed (MarMOT) software has been developed as a tool for comprehensive evaluation of plankton ecosystem models against observational data. It provides a common physical and computational environment in which different ecosystem models can be calibrated and compared. The system is designed specifically to support computationally intensive experiments involving parameter optimization, in which models are evaluated many times with different input data in a 1-D framework.

The core of the system is the MarMOT Model Evaluator (MME), which is implemented as a specific application within a system called the Generic Function Analyzer (GFAn). The MME runs one or more simulation cases, producing a cost function value summarizing the model-data misfit over all cases, together with detailed model output, diagnostics and misfit data for each case. It provides various options for modelling photosynthesis, each of which can be applied to any ecosystem model implemented.

GFAn provides a generic data management framework that adapts to the requirements of the application, together with an optimizer for cost function minimization and a flexible experiment control interface. The data management framework allows different instances of all model inputs (parameters, forcing data and initial conditions) to be easily combined in different ways to drive ensemble simulations for sensitivity and uncertainty analyses or multi-site calibration experiments.

A baseline version of the MarMOT system is described in which two ecosystem models are implemented. In future versions, it is expected that a wide range of ecosystem models will be made available for research purposes through collaborative work with different modelling groups. Investigations of all models should benefit from independent improvements in the functionality of the MME application and the GFAn system, extending the power and range of potential analyses.

KEYWORDS

BIOGEOCHEMISTRY, CARBON CYCLE, DATA ASSIMILATION, ECOSYSTEM MODELLING, HADOCC, MODEL INTER-COMPARISON, NPZD, PARAMETER OPTIMIZATION, PLANKTON MODELS, SENSITIVITY ANALYSIS, UNCERTAINTY ANALYSIS

ISSUING ORGANISATION **National Oceanography Centre, Southampton**
University of Southampton, Waterfront Campus
European Way
Southampton SO14 3ZH
UK

Contents

Chapter 1

Introduction

The need to better understand and forecast environmental change demands reliable models of plankton ecosystems that can be coupled with 3-D ocean circulation models to make inferences about biogeochemical cycles. However, the complexity of biological systems together with the paucity of suitable observational data for constraining models provides major challenges for modellers. Plankton models, although described as mechanistic, are necessarily empirical to a very large degree. They rely on many parameters that are poorly known or difficult to quantify. Although some of these values can be determined experimentally under controlled conditions, the corresponding values in nature are generally highly variable in space and time or across taxa. Fasham and Evans (1995) and Matear (1995) started to address this problem by fitting plankton models to observations from time series stations in the Atlantic and Pacific respectively, using non-linear data assimilation techniques to optimize parameters. A number of other studies have since followed this approach.

Most calibration studies have been based on single sites, although investigators have started to optimize parameters for multiple sites simultaneously in an effort to develop more widely applicable parameter sets (Hurtt and Armstrong, 1999; Schartau and Oschlies, 2003a,b). Satellite ocean colour data can be used to derive parameter sets that are in some sense applicable at global scales, although initial studies suggest that attempts to derive global parameter values for current models are likely to result in a poor compromise for all locations (Hemmings *et al.*, 2003, 2004; Tjiputra *et al.*, 2007). Allowing spatial variation in parameter values, as done by Losa *et al.* (2004, 2006), might be a useful way of learning about model deficiencies, with a view to seeking process parameterizations that would allow fixed parameter models to respond better to regional differences in physical forcing.

A desire to address important scientific questions has led to the development of more complex plankton models. However, the level of complexity that can be justified, given the available data, has been a subject of debate (Anderson, 2005; Le Quéré, 2006). To resolve this we must be able to comparatively evaluate models on the basis of their structure and process formulations. Robust calibration procedures for reducing uncertainty associated with tunable parameters are a fundamental prerequisite for this type of comparison (Dadou *et al.*, 2004; Friedrichs *et al.*, 2006, 2007)

Objective evaluation of plankton models is difficult because of uncertainties in their external input data. Errors in the physical forcing data, for example, can have a major impact on biogeochemical simulations, causing a calibration process to yield inappropriate parameter values (Friedrichs *et al.*, 2006). To make significant progress, we need to be able to assess the plankton models independently from the physical circulation models in which they are designed to run. We need to explore the sensitivity of plankton models to their external inputs, which include physical forcing, initial conditions and horizontal fluxes, and find ways of interpreting model misfit to biogeochemical data that allow for expected errors in these input data. Only by weighting effectively for such uncertainty can we develop the robust calibration procedures that are needed.

A further problem is that the number of free parameters that can be adequately constrained by the available data is typically much less than the number of potential free parameters in a given model. Attempts to fit too many parameters can cause a model to perform badly when evaluated against an independent data set (Friedrichs *et al.*, 2006, 2007). Sensitivity analyses are required to select parameter subsets that impact simultaneously on the key model outputs of scientific interest and those for which observations exist to constrain the model behaviour. Properly allowing for uncertainty in external inputs will tend to further reduce the number of constrainable parameters and this problem will need to be countered by combining many different types of observations from different sources. This should include satellite and *in situ* data. The most effective ways of combining different data sets are yet to be determined.

Together, these challenges faced by ecosystem modellers have provided the rationale for the development of the Marine Model Optimization Test-bed (MarMOT) software described here. Following the test-bed concept of Friedrichs *et al.* (2006, 2007), MarMOT provides a common physical and computational environment in which different marine ecosystem models can be calibrated and compared. It is designed to support computationally intensive experiments in which models are evaluated in a 1-D framework many times with different inputs. A flexible interface makes it easy to apply to a wide range of sensitivity analyses, uncertainty analyses and parameter optimization experiments. MarMOT does not include a 1-D

physical model. All physical forcing is instead provided by external input data. Responses of ecosystem models to a wide range of different physical environments can be examined by providing different instances of the forcing data.

Two ecosystem models are included at present in a baseline version of the system. In future versions, it is intended that a wide range of ecosystem models will be made available for research purposes through collaborative work with different modelling groups. On-going development of the MarMOT system will provide additional functionality that can be applied to all implemented models.

The MarMOT system has been developed at the UK National Oceanography Centre with funding from the Natural Environment Research Council, via the National Centre for Earth Observation (a NERC collaborative centre) and the Oceans 2025 research programme. The author would particularly like to thank Mike Fasham, Andreas Oschlies, Tom Anderson, Peter Challenor and Ian Robinson for their support and encouragement of the development work. It was Mike Fasham's pioneering work on parameter optimization of plankton models that originally motivated the development of such a test-bed facility and provided a firm foundation on which to build.

1.1 MarMOT System Components and Organization

Figure 1.1 gives an overview of the MarMOT system in terms of its main components and the data flows between them. Simulations are controlled by a range of different of input data types, organized as a number of multi-variate *input item tables*. Each input item table can contain one or more instances of a particular *input item*. Different instances of all simulation input items can be combined in different ways using a simple *case table*, making it easy to set up a wide variety of ensemble runs. Each entry in the case table defines a specific combination of input data. It is identified by the values of its *case variables*, which are site and ensemble member labels. Free model parameters can be optimized over all cases in a given case table, so it is straight-forward to set up multi-site calibration experiments.

The core of the MarMOT system is the MarMOT Model Evaluator (MME) that performs ecosystem model runs according to the specifications in the case table. It can provide a range of different outputs that are selected or de-selected according to user requirements. It also calculates a *cost function* value dependant on the misfit between simulation variables and a set of observations or other reference values, if such data are provided. The data are supplied as an additional input item. If

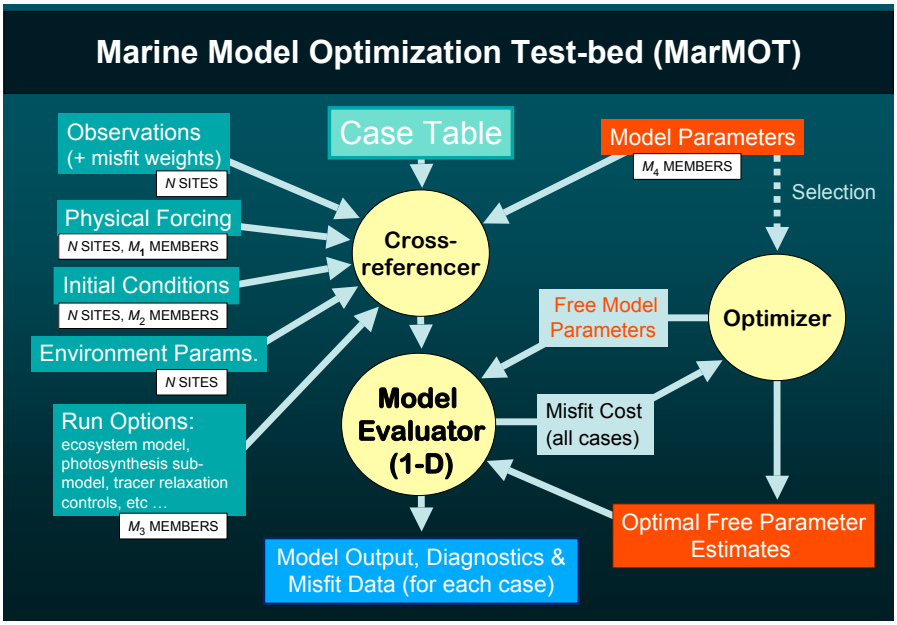

Figure 1.1: Schematic diagram of the MarMOT system, showing the main system components and data flows. Data tables (internal and external) are shown as boxes and processing components as circles.

appropriate, these can be output data from a previous simulation experiment.

The MME is implemented as a specific application within a system called the Generic Function Analyzer (GFAn). GFAn provides a generic *data management framework* that adapts to the requirements of the application, a *cross-referencer* for selecting the required instances of each input item, an *optimizer* for cost function minimization and a flexible *experiment control interface* that can be used to run many experiments in batch mode. The optimizer is well suited to non-linear problems in multi-dimensional parameter space: it includes a genetic algorithm for identifying promising areas of a finite parameter space and a non-gradient direction set algorithm for local minimization. The two algorithms can be used in combination or independently. The direction set method can be applied to bounded or unbounded minimization problems. Log-transformation of parameters and repeat searches from different initial points in parameter space are supported.

GFAn input and output data take the form of ASCII tables with one table per file, so data are easily transferred to or from spreadsheets and other applications. A dedicated suite of programs called Table Toolbox is also available for working with data files in this format. Details of the file format and Table Toolbox programs are

given in the appendices. Extraction from NetCDF data sets is supported for gridded input data.

The GFAn code is written in C. The MME user interface is also implemented in C but the interface code for the models and the ecosystem models themselves are in Fortran. This document describes MarMOT version 1.0 (referring to version 1.0 of the MME application). The GFAn system has been developed alongside the MME. The descriptions relating to GFAn in this document refer to GFAn version 1.0.

1.2 Running Experiments

In a non-optimizing experiment, GFAn calls the MarMOT model evaluator once to perform a single evaluation of the cost function and reports the result as the *objective function value*. The objective function value is zero if there are no observations applicable to any of the input cases. Data are written to any requested *application output tables* as the run proceeds. In an optimizing experiment, GFAN calls the model evaluator many times. Output tables are suppressed automatically during optimization and written out once for the simulation or simulations corresponding to the final parameter values.

A single GFAn experiment is defined by one input case table (optional for a single case) and an input item table for each item (optional if defaults are to be used). One or more additional input tables are needed for optimization experiments. Each item table contains one or more records, each indicating a different item instance. Instances of items that are inherently case-dependent in the experiment are identified by site labels or ensemble member labels or both. Items that are not inherently case-dependent are identified by separate *item key variables*. The cross-referencer matches particular item instances to case specification records in the case table either by *contextual referencing* using the case variable values or *explicit referencing* using item keys.

To avoid expensive re-loading of data common to more than one case, the data for each experiment are assembled in a memory-resident database before processing individual cases. The experiment control interface can handle multiple experiments in batch mode, with and without optimization, without re-loading data items that are already resident. Experiments are configured by providing one or more *control tables* that indicate experiment-specific file names for the input and output tables. GFAn produces a log file including details of all input data used for the experiments, or a user selected subset of these data.

The model evaluator can produce a range of output tables, the variables of which can be selected or deselected via GFAn using *variable selection tables*. These can be provided for any subset of the output tables as needed. The default action in the absence of a variable selection table is to output all variables. Output for different cases is written sequentially to each requested output table, the records corresponding to each case being identified by their case variable values. The Table Toolbox program 'tabsplit' can be used to split an output table into separate files for individual cases.

Chapter 2

The GFAn Data Management Framework

The MarMOT model evaluator communicates with the user via GFAn. It defines the application environment in terms of the dimensions of the model domain (depth and time), the input items, the output tables and the case variables. GFAn then uses the application-defined names for these elements in the user interface. The input items are defined in terms of generic item types that GFAn knows how to handle. These are either parameter sets or domain items that are referenced to one or more domain dimensions. The latter may be gridded or non-gridded. GFAn also recognizes matrix items, although none are used in the current version of MarMOT.

This chapter describes the concepts behind the GFAn data management framework in which the MME runs. The description is given in generic terms applicable to any application. The notation <*symbol*> is used here to denote symbolic names for which another name will be substituted. MarMOT specifics will be described in Chapter 3.

2.1 Input Items

GFAn provides 4 types of data item for organizing input data. Each input item can be multivariate such that instances of an item in the item table can contain many variables that share the same structure. The application defines the number of items of each type, the name of each item and the variables it may contain. Unrecognized variables in the input tables are ignored. The application also specifies whether or not particular items or variables are optional.

The GFAn item types are:

- *parameter set item*: one or more individually named values
- *matrix item*: one or more matrices (or vectors) sharing the same number of rows and columns
- *gridded domain item*: one or more data arrays defined on a common regular grid with axes corresponding to one or more dimensions of the model domain
- *non-gridded domain item set*: one or more vectors of values co-located at arbitrary points on the model domain axes

Different instances of matrix and domain items can vary in structure if appropriate. That is they may have different numbers of rows and/or columns or be defined at different sets of locations in the domain.

Each parameter set item is defined by a single input table. The internal item tables for the other more complex types of input item are compiled using data from 2 or more input files. The first file accessed contains an *item specification table*, the records of which contain metadata that tells GFAn what data to retrieve and how the data values are organized conceptually. Data values are then extracted either from an ASCII format *item data table* or from one or more NetCDF data sets. NetCDF data input is described in Section 2.2.

In each of the ASCII input tables (parameter set item table, item specification table or item data table), specific instances of items are identified by the values of one or more *id variables* as described in the Section 2.3.1. No id variables are needed if there is only one instance. Otherwise, records must be in id variable order. For items with separate item specification and data tables, the instances matching those in the specification file are extracted from the data table to populate the internal item table. Instances must be of the size implied by the metadata and GFAn checks for consistency when loading the data.

In the parameter item tables and the item specification tables, the value '`.`' causes replication of values from previous records. This allows input tables to be prepared in a way that makes it easy to see differences between records and makes it simple to change whole columns of identical values. Other non-numeric data values are treated as missing data ('`_`' is used as standard). In item data tables, all non-numeric values are treated as missing data and represented by the application's missing data value `OBJF_NODATA` (-9999 for MarMOT).

2.1.1 Parameter Sets

Each parameter set item requires a separate input table with one variable representing each parameter. (Variable names are synonymous with parameter names.) Each record in the table represents one instance of the parameter set. In tables with more than one record, different instances are identified by one or more id variables.

2.1.2 Matrices

Item Specification Table

Each record in a matrix item specification table represents one instance of the matrix item, identified by one or more id variables (if needed). The remainder of the record indicates the number of rows and columns (variable names: **nrow** and **ncol**). These can vary between different instances of the item.

Item Data Table

The data table contains item-specific variables as defined by the application plus the id variable(s), with $nrow \times ncol$ data records for each instance.

2.1.3 Gridded Domain Items

Item Specification Table

Each record in a gridded domain item specification table represents one instance of the gridded domain item, identified by one or more id variables (if needed). The remainder of the record indicates the grid specification for each dimension to be included. A particular dimension can be mandatory, optional or disabled for a particular item, according to the requirements of the application. Applications may use default values for missing dimensions or interpret the data values as applying to lines, planes or volumes in the domain (according to the number of undefined dimensions) instead of specific points. The grid specification (including the number of undefined dimensions) can vary between different instances of the item.

There are 6 parameters associated with each dimension in the grid, defined by

the following variables.

- <*dimensionname*>**1**: location of first value
- <*dimensionname*>**step**: grid interval
- **n**<*dimensionname*>: number of grid points
- <*dimensionname*>**n**: location of last value
- <*dimensionname*>**min**: grid minimum
- <*dimensionname*>**max**: grid maximum

A triplet of parameters is required as input to define the grid specification in each dimension. The remaining parameters are determined by GFAn.

The dimension <*dimensionname*> is included in the grid if either of the variables <*dimensionname*>**1** or <*dimensionname*>**min** are present (with non-missing values). The presence of <*dimensionname*>**1** indicates that data are grid registered (i.e. values defined at grid nodes). The presence of <*dimensionname*>**min** indicates that data are interval registered (i.e. values defined at interval mid-points). Grid registration is assumed if both are present. A mix of grid-registered and interval-registered dimensions can be used in the same grid.

If there is only one grid point in a particular dimension, it can be defined by the single variable <*dimensionname*>**1**. Otherwise, for a grid-registered dimension, one of the following pairs of variables is used to complete the specification.

- <*dimensionname*>**step** and **n**<*dimensionname*>
- <*dimensionname*>**n** and **n**<*dimensionname*>
- <*dimensionname*>**n** and <*dimensionname*>**step**

For an interval-registered dimension, one of the following pairs of variables is used to complete the specification.

- <*dimensionname*>**step** and **n**<*dimensionname*>
- <*dimensionname*>**max** and **n**<*dimensionname*>

- <*dimensionname*>**max** and <*dimensionname*>**step**

These variable pairs are listed here in order of preference (i.e. the order in which GFAn looks for them). The remaining grid-specification parameters are calculated from the other three and included in the GFAn log file. If they are present in the input table, their input values are ignored.

Item Data Table

The data table contains item-specific application variables plus the id variable(s), with the appropriate number of data records for each instance (i.e. one record for each grid point). It is important to note that, although the input might also contain dimension variables for user reference, these are ignored by GFAn and replaced by values calculated from the grid specification. The latter appear in the GFAn log file.

2.1.4 Non-gridded Domain Items

Item Specification Table

Each record in a non-gridded domain item specification table represents one instance of the non-gridded domain item, identified by one or more id variables (if needed). The remainder of the record indicates the number of data points for each instance (variable name: **npoint**) and may contain one or more flag variables of the form <*dimensionname*>**flag**. As for gridded domain items, a particular dimension can be mandatory, optional or disabled, according to the requirements of the application. A flag variable is required for each optional dimension and takes the value 1 or 0 to indicate the presence or absence of dimension values for each instance.

Item Data Table

The data table contains item-specific application variables plus the id variable(s). It also contains one or more dimension variables (<*dimensionname*>) giving the location on the respective dimension axis. Dimension variables are expected for all mandatory dimensions and any optional dimensions for which flag variables are present in the item specification table. Optional dimensions that are included are allowed to contain missing values; mandatory ones cannot.

2.2 Input from NetCDF Data Sets

For gridded domain items, the ASCII data file can be replaced by one or more NetCDF data sets from which data are extracted. Multiple data sets are specified by a *NetCDF file table* that contains a list of NetCDF files to be used.

File names for the different types of input files (ASCII, NetCDF data set or NetCDF file table) are derived from the name given for the file containing the item specification table. The naming convention is described in Chapter 4. If the expected data file is not found, GFAn looks instead for a NetCDF data set. If the expected NetCDF file is not then found, it looks for a NetCDF file table. As well as allowing the specification of multiple data sets, file tables can be used to associate parameters with individual NetCDF data sets that control how they are processed. These can include parameters that specify variable name translations to map application variable names onto corresponding NetCDF variable names.

2.2.1 Data Extraction and Gridding

The NetCDF data set must contain variables or dimensions that match the names of the dimensions specified in the gridded domain item specification table (after translation if applicable). These are treated as the co-ordinate variables for the grid so each must be a vector (i.e. a uni-dimensional NetCDF variable or a NetCDF dimension).

The variables to be extracted are any with names matching the item variable names defined by the application (after translation if applicable). These variables must have the expected internal NetCDF dimensions, as determined from those of the co-ordinate variables. The variable dimensions can appear in any order but the order must be the same for all variables to be extracted.

For each instance of the gridded domain item, data are extracted from the NetCDF data set and mapped onto the grid defined by the metadata in the item specification table. A search radius of half the step size in each specified dimension is used to locate relevant data. Multiple data values within the search radius of a particular grid point are averaged.

2.2.2 NetCDF File Tables

NetCDF file tables contain one file name path per record in the variable **ncfile**. They may also contain any of the following processing parameter variables or variable groups.

- **fillval**: NetCDF value to be treated as missing data
- <*dimensionname*>: NetCDF co-ordinate variable name for application's dimension
- **scale.**<*dimensionname*>: scale for pre-calibrating co-ordinate variable
- **offset.**<*dimensionname*>: offset for pre-calibrating co-ordinate variable
- <*itemvariablename*>: NetCDF variable name for application's item variable
- **scale.**<*itemvariablename*>: scale for calibrating item variable
- **offset.**<*itemvariablename*>: offset for calibrating item variable

Variable name translations are performed for each of the application dimension variables (<*dimensionname*>) and item variables (<*itemvariablename*>) that are included in the file table. Name translation for dimension variables includes an option that allows NetCDF dimension indices to be used directly for data location. This is necessary when searching on dimensions for which NetCDF co-ordinate variables are not provided. Access by dimension index is requested by supplying a name of the form **num.**<*NetCDFdimensionname*> as the value of <*dimensionname*>, instead of a NetCDF co-ordinate variable name. The NetCDF dimension's index vector, with values 1 .. <*dimensionlength*> is then treated as if it were the co-ordinate variable.

Scales and offsets are used to apply a linear pre-calibration to the co-ordinate variables before searching occurs or to calibrate item variables on extraction. This facility is provided to allow access to data with different units from those used by the application.

Where item variables are optional, the item variables to be extracted are those present in the first NetCDF data set accessed. These then become mandatory with respect to subsequent data sets accessed via the same NetCDF file table. A fixed group of item variables are therefore active throughout the extraction process for a particular item. However, the corresponding NetCDF variable names can change between data sets if necessary.

2.3 Multi-case Support and Cross-referencing

2.3.1 Id Variables

Each case to be processed is identified by one or more application-defined case variables.

The case-dependent data for each case can be identified either by contextual reference or explicit reference. The method can vary between items depending on the presence or absence of the item key variable in the item table. For explicit referencing, this variable (variable name: <*itemname*>`key`) must be present. It then becomes the active id variable and any case variables in the input table are ignored. If the item key variable is absent, contextual referencing occurs using the values of the case variables present. These are then the active id variables. Particular data items may be dependent on some case variables but not others, so it is not necessary for all of the application's case variables to be present for each item.

The input data must be in order of the active id variable(s). All id variables are handled as text strings with the caviat that non-numeric strings precede numbers and numerical values are compared numerically. The case variables are compared in order of dominance, this being the order in which they are defined by the application. If item data are dependent on multiple case variables, the data records must therefore be organized in a hierarchy of sub-groups with the least dominant case variable varying fastest. Default instances of items can be supplied by using blank id values. Global defaults (i.e. those for which all id variables are blank) must appear at the top of the table. Sub-group defaults must appear first in each sub-group. GFAn validates the input data accordingly.

To avoid conflicts with Table Toolbox processing it is advisable to avoid the characters '`.`', '`~`' and '`/`' within id variable values.

2.3.2 The Case Table

In one objective function evaluation, the GFAn application processes one or more cases of its input data. (In the MME application, a simulation is performed for each case.) Each case is defined by one record in the case table. This case table entry indicates, contextually or explicitly, which instances of any case-dependent input items are to be used. Before the first objective function evaluation in a particular experiment, GFAn performs the cross-referencing to determine the required

instances for each case and their location in memory for rapid access. If there is only a single case to be processed and single instances of each item in the item tables, no input case table is needed. A dummy case table is then created.

A case table must include all of the application's case variables and the case table records must be ordered on these variables. (As in the item tables, the case variables are compared in order of dominance.) It must also include item key variables for any items that are to be referenced explicitly (variable name: <*itemname*>`key`). For other items the required data are selected to match the case, using the case variables (if any) present in the item tables. If one or more case variables are absent from an item table, the item can match more than one case. Items with no id variables in their item tables are case-independent and match all cases.

Duplicate cases, as identified by the combination of values for all case variables, are not allowed in the case table, even if the records have different values of the item key variables. This ensures that any output can be uniquely identified by the case variables.

2.3.3 Cross-referencing Errors

Cross-referencing fails under any one of the following conditions.

- Explicit referencing is specified (by the presence of an item key variable in the case table) but there is no key variable in the relevant item table, unless there is only one instance of the item.
- Explicit referencing is specified but there is no matching instance and no default.
- Contextual referencing is specified (by the absence of an item key variable in the case table) but the item key variable is present in the item table, unless there is only one instance of the item.
- Contextual referencing is specified and there are one or more case variables in the item file that have no matching values and no defaults.

Chapter 3

The MarMOT Model Evaluator (MME)

The MME outputs a cost function value that is essentially a weighted average of all model-data misfits, over all specified simulation cases (or zero in the absence of any applicable data). This value is the objective function value that is minimized in parameter optimization experiments. Optimization is usually applied to a particular ecosystem model. In non-optimizing experiments, the ecosystem model selected can vary between individual simulations. This is also possible in optimization experiments where the parameters to be optimized are not specific to one of the ecosystem models.

MarMOT provides a generic user interface common to all ecosystem models implemented. The MME handles a superset of prognostic and diagnostic variables and transfers the necessary information between the MarMOT data area and the active ecosystem model at each time step, allowing ecosystem models to be implemented with minimal changes to their native variables and code.

In general, although some variables are ubiquitous, different ecosystem models have different sets of prognostic variables. The variables can be either tracer concentrations or composition ratios for particular ecosystem components. Where two or more tracer variables are linked by composition ratios there are alternative sets of prognostic variables. While the tracer values are needed for calculating transports, it is usually more convenient to initialize linked tracers using composition ratios and this is the convention supported by MarMOT.

MarMOT maintains two sets of tracers: *primary tracers* and *derived tracers*. Only the primary tracers and composition ratios are initializable. Each derived

tracer is then determined by the values of one or more primary tracers and zero or more composition ratios. The primary tracers are nitrogen concentrations wherever possible. Carbon concentrations for all organic components are handled as derived tracers. Derived tracers may or may not feature as prognostic variables in a particular ecosystem model.

Where composition ratios are not handled by a particular ecosystem model, external values are used in the calculation of diagnostic tracer values if provided. It is possible for these ratios to be depth dependent. However, ratios not handled by the ecosystem model will remain fixed in time irrespective of any primary tracer transport fluxes, which means that depth variations in such ratios will lead to non-conservation of the diagnosed tracers.

3.1 GFAn Environment for the MME Application

This section describes the application environment components that are defined by the MarMOT Model Evaluator and used by GFAn to provide the application-specific user interface.

3.1.1 Case Variables

The MME application's case variables are **site** and **member**, **site** being the dominant variable. Their intended use is for identifying individual sites and ensemble members respectively but this can be adapted to user requirements. As id variables, they can take text string or numeric values.

3.1.2 Dimensions

The recognized dimensions are:

- **t**: time since origin in days
- **year**: calendar year (or year number)
- **tofyr**: time since start of year in days
- **k**: depth level number

- **z**: depth in m
- **i**: horizontal grid co-ordinate 1
- **j**: horizontal grid co-ordinate 2

The domain has 2 independent dimensions, time and depth. Each simulation is performed on a fixed vertical grid that divides the depth range into a number of discrete levels 1..**nk**, level 1 being the top (surface) level. The primary dimension variables used for input and output are **t** and **k**. The variables **year** and **tofyr** provide an alternative 2-dimensional representation of time that can be used in specifying initial conditions (and should be recognized for observations in future versions). The variable **z** is used in place of **k** in specifying observations. All of these dimension variables are included in the output tables.

The remaining dimension variables (**i** and **j**) are provided solely for use in the context of NetCDF data extraction, usually for specifying a location in the horizontal. One or both could also be used for locating data in a non-spatial dimension or dimensions (e.g. for selecting a member from an ensemble). They are not used within the MME.

3.1.3 Input Items

MarMOT inputs include a model-specific parameter set item for each model and 5 additional parameter set items for setting up various run options and environmental conditions. It recognizes 6 other input items that are domain-referenced. All input items are optional by default, making it relatively easy to set up simple simulation experiments with minimal input data. Where items become mandatory as a result of inter-dependencies, MarMOT performs consistency checks to ensure the required data are present.

The complete list of input items recognized is given below in the order that GFAn reads in the data. Parameter items are processed first, followed by the gridded domain items and a single non-gridded domain item.

- **model1**, **model2**: *model parameter set items* - specify external parameter sets for each ecosystem model to be used in place of internal defaults. Model parameter sets include any model-specific option flags.
- **option**: *option parameter set item* - specifies the particular model to be run and various options and values that are typically independent of the site for

which the model is run. Omitting this item causes the simulation to be run with no dynamics so that any initialized concentrations remain constant in time.

- **taxis**: *time axis parameter set item* - defines the time axis origin, year length if fixed and whether to use periodic annual forcing. The default axis starts at year 1 with periodic 365 day forcing.
- **timeperiod**: *time period parameter set item* - specifies the start and finish times for the simulation. The default time period is one calendar year starting at the time axis origin.
- **environ**: *environment parameter set item* - gives time invariant values that are typically site dependent, including maximum depth (default: 50 m), latitude and any constant forcing parameters. Default forcing values are always zero. There is no default latitude so an environment item is mandatory in cases where latitude is required to determine the light field for photosynthesis.
- **mfitctrl**: *misfit control parameter set item* - specifies options for determination of model-data misfit.
- **zlevel**: *vertical grid item* - gridded domain item referenced to the **k** dimension, specifying up to 500 depth levels. Omitting this item causes the simulation to be run for a single level stretching from the surface to the maximum depth.
- **init**: *initial profile item* - gridded domain item referenced to the **k** dimension (and optionally to **t** or **year** and **tofyr** dimensions), defines initial profiles of primary tracers and/or composition ratios. Omitting this item causes default initial conditions to be used (see Section 3.4).
- **ft**: *scalar forcing item* - gridded domain item referenced to the **t** dimension.
- **fkt**, **fkt2**: *profile forcing items* - gridded domain items referenced to **t** and **k** dimensions. Two identically defined items are recognized to allow variables to be divided between different time grids. The **t** dimension is optional and can be omitted for time-invariant profiles.
- **obset**: *observation set item* - non-gridded domain item referenced to **t** and **z** dimensions.

To support extraction of data from NetCDF data sets, the vertical grid, initial profile and forcing data items can optionally be referenced to the horizontal dimension variables (**i** and **j**), as well as to the dimension variables mentioned above. Likewise, the scalar forcing item can additionally be referenced to one of the vertical dimension variables (**k** or **z**). A list of possible variables in each input item, together with descriptions, is given in template files generated on request (see Chapter 4).

3.1.4 Output Tables

In addition to the cost function value, the MME can produce up to 5 different output tables, listed below.

- **outt**: *end-of-timestep scalar output* - includes scalar forcing variables, level 1 values from profile output (below) and scalar diagnostic variables.
- **outkt**: *end-of-timestep profile output* - includes full-depth forcing, primary tracers, composition ratios, derived tracers and full-depth diagnostics.
- **outtday**: *daily mean scalar output* - variables as for end-of-time-step scalar output.
- **outktday**: *daily mean profile output* - variables as for end-of-time-step profile output.
- **outmfit**: *misfit table* - includes model-data differences, weighted misfits, simulated and observed values and misfit weighting factors.

By default, the output tables include the complete superset of all forcing data, prognostic and diagnostic variables. Blank values are output for data that are not relevant to the active ecosystem model, which may differ between cases. A very large number of variables are available of which only a small subset may be needed. The tables can be tailored to user requirements by including variable selection tables for each in any particular experiment. A list of possible variables in each output table is given, together with descriptions, in variable selection table templates generated on request (see Chapter 4).

3.2 Time Axis

[*Input item:* **taxis**]

The time axis corresponds to the **t** dimension. Times specified in the 2-D space defined by dimensions **year** and **tofyr** map onto the **t** axis such that its origin is fixed at the start of a base year. The base year must be the earliest start year of all simulations to which it is applied, or some year before. This means that **t** is never negative in the output tables. Years can be fixed length or actual calendar years. Periodic annual forcing can be used with fixed length years only.

By default, the origin is at the start of year 1, the year is fixed length (365 days) and periodic forcing is enabled. The origin can be set to the start of a particular year by setting the **baseyear** parameter in the time axis parameter set item table. This changes the default configuration so that the year length varies between 365 and 366 days, according to the calendar year, and periodic forcing is disabled.

Periodic forcing is set on or off by setting the **periodic** parameter to 1 or 0 respectively. Including this parameter with the **baseyear** parameter causes the default year length to revert to a fixed 365 days. Irrespective of the base year and periodic forcing settings, a fixed year length can be set using the parameter **yearlen**. This can be any positive integer. If actual calendar years are required, **yearlen** and **periodic** variables must be omitted or have missing values.

3.3 Depth Axis and Vertical Grid

[*Input items:* **environ**, **zlevel**]

The depth axis corresponds to the **z** dimension. **z** is positive downward with the origin at the sea surface. For each simulation, the **z** axis is divided into a finite number of intervals or depth levels, with level number corresponding to the **k** dimension. Level 1 is the level immediately below the sea surface. The vertical grid is fixed for each simulation. An option for dynamic vertical grids may be required in future versions of MarMOT for shelf-sea applications, where tidal variations in water column depth are important. In the current version, the water column depth is a constant environment parameter (**maxdep**).

The vertical grid is defined by providing a bottom-of-level depth in m for each level. Each record in the vertical grid item specification table defines the number of levels **nk** for a particular instance on the grid (up to a maximum of 500). The first level number **k1** and the interval **kstep** must both be 1. The data table must contain a variable **zbot** with the required number of level bottom depths for each instance. The mid-point of the top level must lie above the maximum depth, defined by the parameter **maxdep**, at all sites to which the grid is applied.

If a horizontal location is needed, for extracting **zbot** data from a NetCDF data set, then a single point in one or both of the dimensions **i** and **j** can be defined in the vertical grid item specification table. This is done by specifying **i1** and **istep** and/or **j1** and **jstep**. The step variables determine the search radius in each dimension, which is half the step size.

3.4 Initial Conditions

[*Input items:* **option**, **init**]

Depth-varying initial conditions for primary tracers and/or composition ratios can be provided by including the relevant variables in the initial profile item data table. Default values for any primary tracer and composition ratio variables omitted can be provided in the option parameter set item table (using the same variable names). These depth-invariant values are used only where the variables are omitted from the initial profile item. They are not substituted for missing values of variables included therein.

The initializable variables are the 7 primary tracers and 4 composition ratios listed below.

- **din**: dissolved inorganic nitrogen (DIN) (mmol N m^{-3})
- **phy**: phytoplankton nitrogen (mmol N m^{-3})
- **zoo**: zooplankton nitrogen (mmol N m^{-3})
- **det**: detrital nitrogen (mmol N m^{-3})
- **nh4**: ammonium (mmol N m^{-3})
- **dic**: dissolved inorganic carbon (DIC) (mmol C m^{-3})
- **alk**: alkalinity (meq m^{-3})
- **rcchl**: phytoplankton C:chlorophyll ratio (g C (g Chl)$^{-1}$)
- **rcnphy**: phytoplankton C:N ratio (mol C (mol N)$^{-1}$)
- **rcnzoo**: zooplankton C:N ratio (mol C (mol N)$^{-1}$)
- **rcndet**: detritus C:N ratio (mol C (mol N)$^{-1}$)

This list will be expanded with the implementation of additional ecosystem models in MarMOT.

Initial profiles are defined on the **k** grid used in the simulation, which may be case-dependent. In the initial profile item specification table, the first level number **k1** and the interval **kstep** must both be 1. The number of values **nk** may differ from the number of levels in the grid but sufficient values must be provided to initialize

the whole water column, down to the maximum depth **maxdep**. Initial data for any levels with mid-points below this are ignored. The item data table is not expected to contain depth information. Depths corresponding to **k** values are determined from the vertical grid item data.

A time at which the initial conditions are valid can optionally be included in the initial profile item specification table. This is done by defining a single grid point in either the **t** dimension (by specifying **t1**) or in both the **year** and **tofyr** dimensions (by specifying **year1** and **tofyr1**). If a time is present, then it must be within 1 day of a simulation start time (ignoring fractions of a day) to be valid for that simulation.

If a horizontal location is needed, for extracting **zbot** data from a NetCDF data set, then a single point in one or both of the dimensions **i** and **j** can be defined in the initial profile item specification table. This is done by specifying **i1** and **istep** and/or **j1** and **jstep**. The step variables determine the search radii.

The specified initial conditions are used to initialize the prognostic variables required by the active ecosystem model at the start of a simulation. Any required tracer concentrations that are not explicitly initialized (or are negative) will be initialized to zero. Required composition ratios not explicitly initialized are set to model defaults. Initialized composition ratios will be overridden if particular values or ranges are imposed by the model. The actual initial values used are included in the end-of-timestep output tables, preceding the output for each simulation. Model-specific details are given in Section 6.

3.5 Environmental Forcing

[*Input items:* **option**, **environ**, **ft**, **fkt**, **fkt2**]

The ecosystem models are forced by time series of a number of different variables, of which some are scalars and some are full-depth profile vectors. The scalar forcing variables are

- **sol**: solar radiation incident on sea-surface (W m^{-2}, downwards)
- **solav**: daily mean solar radiation incident on sea-surface (W m^{-2}, downwards)
- **mld**: mixed layer depth (m)
- **sss**: sea-surface salinity

- **pco2atm**: atmospheric pCO_2 at air-sea interface (μatm)

The full-depth forcing variables are

- **vdc**: vertical diffusion coefficient at bottom of level ($m^2\ d^{-1}$)
- **w**: vertical velocity at bottom of level ($m\ d^{-1}$, upwards)
- **temp**: temperature (°C)
- **dinref**: reference DIN concentration (mmol N m^{-3})
- **phyref**: reference phytoplankton concentration (mmol N m^{-3})
- **zooref**: reference zooplankton concentration (mmol N m^{-3})
- **detref**: reference detritus concentration (mmol N m^{-3})
- **nh4ref**: reference ammonium concentration (mmol N m^{-3})
- **dicref**: reference DIC concentration (mmol C m^{-3})
- **alkref**: reference alkalinity concentration (meq m^{-3})

The solar radiation affects the light available for photosynthesis (see Section 3.6). The mixed layer depth (MLD), vertical diffusion coefficient and vertical velocity together determine the passive physical transport of the tracers by the water. The reference profiles and possibly the mixed layer depth are used in the context of tracer relaxation. Temperature, salinity and atmospheric pCO_2 are used in connection with air-sea CO_2 flux calculations and temperature is also used in determining some model rates. Details are model-specific (see Chapter 6). Mixed layer depth also has some model-specific uses. Further variables will be added as required when new ecosystem models are implemented.

3.5.1 Transport of Tracers

Passive transport of the tracers is fully determined by the mixed layer depth (MLD), the vertical diffusion coefficient and the vertical velocity. Vertical velocities and diffusion coefficients at top and bottom boundaries are always zero. No provision is made for slope currents at present and any non-zero values for the bottom level in the water column are ignored.

The use of MLD is controlled by the boundary layer mixing option **mixopt** in the option parameter set. If **mixopt** = 0 (the default), no explicit boundary layer mixing is applied (although equivalent mixing can be set up using vertical diffusion coefficient profiles). This does not prevent MLD input data from being used in tracer relaxation control (see Section 3.5.2) or model specific functions (see Chapter 6). Other options are (1) homogenize tracers over all levels wholly above the MLD at the end of each timestep or (2) do the same but also apply partial mixing to the level spanning the MLD. This involves mixing in tracer from the fraction of that level above the MLD and adjusting the concentration in the partially mixed level to conserve total tracer. Boundary layer mixing is applied immediately before tracer relaxation, which occurs at the end of each time step.

The use of the vertical diffusion coefficient is enabled or disabled by setting the option parameter **diffusion** to 1 or 0 respectively. The default is no diffusion (**diffusion** = 0).

The effect of the vertical velocity field on the tracers depends on the choice of advection scheme. Importantly, the advection scheme may also apply additional 'active' velocities (velocities relative to the water) that the ecosystem model assigns to particular tracers, if it is not handling them internally. Details and available options are model-specific.

Three different advection schemes are available, each introducing different amounts of implicit diffusion. A particular scheme is selected by the option parameter **advection**, set to one of the following.

0. no advection; no active tracer velocities if handled externally (default)
1. central difference scheme
2. upstream difference scheme
3. Multidimensional Positive Definite Centered Difference scheme (MPDCD)

The following general equation describes the tracer change due to advection in a particular grid cell or level with height Δz.

$$\frac{dX}{dt} = \frac{w_{\text{bot}} X_{\text{bot}} - w_{\text{top}} X_{\text{top}} + (w_{\text{top}} - w_{\text{bot}}) X}{\Delta z} \tag{3.1}$$

where w_{bot} and w_{top} are the upward velocities at the bottom and top of the grid cell, X_{bot}, X_{top} are tracer concentrations at the corresponding boundaries and X is the

tracer concentration within the cell. The last term in the numerator represents the horizontal tracer flux due to vertical divergence in the velocity field (zero for active velocities). The horizontal tracer gradient is assumed to be zero.

In the basic central difference scheme and in the MPDCD scheme, introduced by Lafore *et al.* (1998), the boundary concentrations are averages of the concentrations either side of the boundary. In the upstream scheme they are the concentrations for the upstream cell. Central differencing has the advantage of conceptual simplicity and low numerical diffusion but is ill-suited to biogeochemical simulations as it can produce negative tracer concentrations in regions of sharp gradients. The positive definite upstream differencing scheme solves the problem of negative concentrations but is problematic due to high implicit diffusion. The MPDCD scheme retains the advantages of central differencing but applies a flux limiter ratio to implement a sufficient condition for avoiding negative concentrations.

For a particular grid cell, the flux limiter ratio is given by

$$\beta^{\mathrm{OUT}} = \frac{X}{\frac{F^{\mathrm{OUT}}}{\Delta z}\Delta t} \tag{3.2}$$

where Δt is the length of the time step and F^{OUT} is the tracer flux out of the grid cell (expressed here as a flux per unit area).

$$F^{\mathrm{OUT}} = [w_{\mathrm{bot}} X_{\mathrm{bot}}]^- + [w_{\mathrm{top}} X_{\mathrm{top}}]^+ + [(w_{\mathrm{top}} - w_{\mathrm{bot}})X]^- \tag{3.3}$$

where $[.]^+ \equiv \max(0, .)$ and $[.]^- \equiv \max(0, .)$. The tracer concentrations are those at the beginning of the time step. When $\beta^{\mathrm{OUT}} < 1$, more than the initial amount of tracer in the cell would be removed during the time step. All fluxes out of the cell are then multiplied by the factor β^{OUT} to prevent this.

The general equation for the application of the flux limiter ratio in the MPDCD scheme is

$$\hat{F}_i = \min(1, \beta_{i-1}^{\mathrm{OUT}})[F_i]^+ + \min(1, \beta_i^{\mathrm{OUT}})[F_i]^- \tag{3.4}$$

where F_i at the flux at the boundary between cells $i-1$ and i in the direction of increasing i. This gives the corrected flux $\hat{F}_i$ for each boundary. In the 1-D context here, the horizontal flux is corrected on the basis that β^{OUT} is horizontally uniform (consistent with zero tracer gradient). The corrected total flux is then

$$\begin{aligned}\hat{F} &= \min(1, \beta_{k+1}^{\mathrm{OUT}})[w_{\mathrm{bot}} X_{\mathrm{bot}}]^{+} + \min(1, \beta_{k}^{\mathrm{OUT}})[w_{\mathrm{bot}} X_{\mathrm{bot}}]^{-} \\ &\quad - \left(\min(1, \beta_{k}^{\mathrm{OUT}})[w_{\mathrm{top}} X_{\mathrm{top}}]^{+} + \min(\beta_{k-1}^{\mathrm{OUT}})[w_{\mathrm{top}} X_{\mathrm{top}}]^{-}\right) \\ &\quad + \min(1, \beta_{k}^{\mathrm{OUT}})(w_{\mathrm{top}} - w_{\mathrm{bot}})X. \end{aligned} \tag{3.5}$$

where k is the level number. Equation 3.1 is re-written in terms of the corrected flux, so

$$\frac{dX}{dt} = \frac{\hat{F}}{\Delta z}. \tag{3.6}$$

A comparison of the 3 advection schemes in the context of 3-D biogeochemical simulations is given by Oschlies and Garçon (1999). They showed the effect of numerical diffusion to be a little greater with the MPDCD scheme than with the simple central difference scheme but much less than with the upstream scheme.

3.5.2 Tracer Relaxation

The primary tracers can be relaxed to fixed or time-varying reference profiles. This feature can be used to compensate for missing horizontal fluxes or other sources of error. Relaxation rates (in fraction of departure corrected per day) are set for individual tracers in the option parameter set item. The variable name for each relaxation rate parameter is of the form **rlx**<*tracername*>. Default rates are zero, implying no relaxation. A relaxation rate r implies a relaxation flux

$$\frac{dX}{dt} = (X - X_{\mathrm{ref}})r \tag{3.7}$$

for a tracer having a concentration X and a reference concentration X_{ref}.

Setting the daily rate r to match the number of timesteps per day (24 by default or the value of option parameter **nstepday** if given) causes the tracer values to be fully replaced by the reference profiles at each timestep, allowing particular variables to be held constant for diagnostic purposes. If leapfrog time-stepping is used (see Section

3.7.2), relaxation applies over a double time step so r should be set to half the number of time steps per day. This feature can also be used with a time-varying reference field, directly inserting replacement data for one or more tracers at each timestep such that the tracer trajectory is determined externally and not dependent on the model. Replacement data could come from the output of a previous simulation.

Relaxation can be suppressed above or below a dynamic reference depth z, under the control of two option parameters: the relaxation mask flag **rlxmask** and the relaxation mask depth option **rlxmaskopt**. **rlxmask** can be (0) no mask, (-1) mask at z and above or (1) mask at z and below. **rlxmaskopt** sets the reference depth to either (1) the mixed layer depth, (2) the euphotic zone depth, defined as the depth at which PAR is 1% of surface value, or (3) the greater of MLD and euphotic zone depth. The relaxation mask is the same for all relaxed tracers. The euphotic zone depth is available as a diagnostic variable (**zeuphotic**) in the MarMOT output.

3.5.3 Options for Data Provision

MarMOT allows a lot of flexibility in the way the input data are provided so that simple experiments can be set up easily. Full-depth profile variables can be defined as scalars if no depth variation is required, avoiding the need to replicate values. Similarly, scalar forcing variables that are to be held constant can be defined as environment parameters.

Data for scalar forcing variables are given in the scalar forcing item data table and those for the profile variables are given in either one or two profile forcing item data tables. Use of multiple profile forcing items allows different profile forcing variables to be defined on different time grids in the same simulation. Alternatively, the feature can be used to set up a subset of profile forcing variables that are not time-dependent. Each variable is only allowed to appear in one of the profile forcing item data tables.

Any full-depth profile variables in the scalar forcing item serve as defaults for variables omitted from the profile forcing items. Likewise, any forcing variables appearing in the environment parameter set item serve as defaults for forcing variables not provided elsewhere. Forcing variables not defined anywhere are set to zero.

All instances of time-varying forcing data are defined at fixed time intervals, as determined by the entries in the respective item specification tables. Each time grid must span the time period of any simulation for which the forcing data are to be used. In cases of periodic forcing (time axis parameter **periodic** = 1), the forcing should span the time period from 0 to the year length on the **t** axis. Forcing for this

period is then applied to all years.

The profile forcing data are defined on the **k** grid used in the simulation. In the item specification table, the first level number **k1** and the interval **kstep** must both be 1. As for the initial conditions, **nk** may differ from the number of levels in the grid but sufficient values must be provided for the whole water column above the maximum depth **maxdep**. Once again, depths corresponding to **k** values are determined from the vertical grid item data.

If a horizontal location is needed, for extracting **zbot** data from a NetCDF data set, then a single point in one or both of the dimensions **i** and **j** can be defined in the item specification table. This is done by specifying **i1** and **istep** and/or **j1** and **jstep**. The step variables determine the search radii. For scalar forcing items, **k** or **z** can be used to define the vertical position if it is required for compatibility with the NetCDF data (by specifying **k1** or **z1** and the corresponding step variable).

3.5.4 Time Interpolation

Forcing data are interpolated from the input item grids to the mid-point of each simulation time step, or the end of each time in the case of any reference profiles for tracer relaxation. This is done according to the interpolation mode for each variable, determined by the option parameter set. Interpolation mode can be (0) not applied, (1) linear, (2) nearest grid point or (3) persistent from last grid point. Linear interpolation is the default. However, options (2) and (3) can sometimes be more appropriate for time-averaged variables such as solar radiation, as linear interpolation does not conserve the time integral. If the interpolation mode for a variable is 0 then any input forcing data are ignored and the variable is set to zero. The interpolation mode for a forcing variable <*fvar*> is set by the option parameter **fmode**<*fvar*>. The default forcing mode can be changed by setting the option parameter **fmode**.

Interpolation of data at grid points is not supported so MarMOT does not allow missing values in the forcing data.

3.6 Photosynthesis and Optics

[*Input items:* **option**, **environ**, **ft**]

Limitation of photosynthesis by available light is handled by a photosyntheis sub-

model. The sub-model is normally external to the ecosystem model and comprises individually selectable models for the attenuation of light penetrating down through the water column, the fraction of the available light that can be absorbed by the phytoplankton and the light-limitation of photosynthetic carbon fixation.

The following MarMOT option parameters are used to configure the photosynthesis sub-model.

- **photmodel**: light limitation model number (if any)
- **kdmodel**: light attenuation model number
- **kdpathopt**: light attenuation model option for path-length adjustment (if any)
- **achlmodel**: light absorption model number (if any)

Any sub-model configuration can be used with any ecosystem model. If no light limitation model is selected, the whole external photosynthesis sub-model is switched off, including the determination of available light. Any selected ecosystem model will then run with zero primary production unless it calculates photosynthesis internally. Internal photosynthesis sub-models, where available, are activated by setting option variables in the ecosystem model parameter sets (see Chapter 6).

The photosynthesis sub-models are forced by surface solar radiation. The forcing variable is either point-in-time irradiance (**sol**) or daily mean irradiance (**solav**), depending on the light limitation model selected. The derivation of photosynthetically available radiation (PAR) from total solar radiation is ecosystem model-specific, the PAR fraction being determined by model parameter values. An option to use surface PAR forcing directly should ideally be included in future versions. The environmental parameter latitude (**lat**) may be required, depending on the light-limitation model and the path-length adjustment option. Also, one or both of two additional option parameters may be required, depending on the light attenuation model:

- **attenwater**: downwelling PAR attenuation due to water (m^{-1})
- **attenpig**: downwelling PAR attenuation due to pigment (m^2 mg^{-1})

A full description of the photosynthesis-sub model options is given at the end of this section.

3.6.1 Process Overview

The biomass-specific rate of photosynthetic carbon fixation (in d^{-1}) can be expressed as

$$\mu = \frac{1}{P_{\mathrm{C}}} \times PAR \times \bar{a^{\star}} \times Chl \times \phi \tag{3.8}$$

where P_{C} is the concentration of phytoplankton carbon (in mmol C m^{-3}), PAR is the scalar photosynthetically available radiation (PAR, in E m^{-2} d^{-1}), $\bar{a^{\star}}$ is the absorption per unit chlorophyll (in m^2 (mg Chl)$^{-1}$) averaged over the PAR spectrum, Chl is the concentration of phytoplankton chlorophyll and ϕ is a quantum yield (in mmol C E^{-1}). PAR is the total irradiance $E(\lambda)$ in the waveband $\lambda = 400$ nm to $\lambda = 700$ nm. i.e.

$$PAR = \int_{400}^{700} E(\lambda)\mathrm{d}\lambda \tag{3.9}$$

The spectrally averaged chlorophyll-specific absorption (sometimes referred to as the spectrally averaged chlorophyll absorption cross-section) is then defined by

$$\bar{a^{\star}} = \frac{\int_{400}^{700} a^{\star}(\lambda)E(\lambda)\mathrm{d}\lambda}{PAR}, \tag{3.10}$$

where $a^{\star}(\lambda)$ is the spectrally varying chlorophyll-specific absorption for the phytoplankton (which includes the effect of absorption by any co-varying accessory pigments). The product $PAR \times \bar{a^{\star}} \times Chl$ is the rate at which light energy is absorbed.

The penetration of light through the water column is modelled in terms of downwelling PAR

$$PAR_{\mathrm{d}} = \int_{400}^{700} E_{\mathrm{d}}(\lambda)d\lambda. \tag{3.11}$$

For a depth interval z_1 to z_2

$$PAR_{\mathrm{d}}(z_2) = PAR_{\mathrm{d}}(z_1)\exp(-K_{\mathrm{dPAR}}(z_2 - z_1)) \tag{3.12}$$

where, K_{dPAR} is the diffuse attenuation coefficient for downwelling PAR, averaged over the depth interval. The vector quantity PAR_{d} is related to scalar PAR by a geometric correction factor g that depends on the optical characteristics of the water. At any given wavelength

$$g = \frac{K_{\mathrm{d}}}{a}\left[1 - R\left(\frac{K_{\mathrm{u}}}{K_{\mathrm{d}}}\right)\right] \tag{3.13}$$

where, K_{d} and K_{u} are the diffuse attenuation coefficients for downwelling and upwelling irradiance respectively, R is the reflectance and a is the absorption coefficient (an inherent optical property of the water body). The importance of this correction factor is discussed by Morel (1991).

In terms of the downwelling radiation, Equation 3.8 becomes

$$\mu = \frac{1}{P_{\mathrm{C}}} \times PAR_{\mathrm{d}} \times \bar{a_{\mathrm{d}}^{\star}} \times Chl \times \phi \tag{3.14}$$

where $\bar{a_{\mathrm{d}}^{\star}}$ is a new spectrally averaged chlorophyll-specific absorption defined in terms of downwelling PAR:

$$\bar{a_{\mathrm{d}}^{\star}} = \frac{\int_{400}^{700} a^{\star}(\lambda) g(\lambda) E_{\mathrm{d}}(\lambda) d\lambda}{PAR_{\mathrm{d}}}. \tag{3.15}$$

To avoid the use of a spectrally dependent absorption parameter, the photosynthesis equation can alternatively be expressed in terms of the photosynthetically usable radiation (PUR): the fraction of PAR that can be absorbed by the phytoplankton. This is given by

$$PUR = \frac{\bar{a_{\mathrm{d}}^{\star}}}{a_{\mathrm{max}}^{\star}} PAR_{\mathrm{d}} \tag{3.16}$$

where $a_{\mathrm{max}}^{\star}$ is the spectral maximum of the chlorophyll-specific absorption $a^{\star}(\lambda)$. Photosynthetic rate is then

$$\mu = \frac{1}{P_{\mathrm{C}}} \times PUR \times a_{\mathrm{max}}^{\star} \times Chl \times \phi. \tag{3.17}$$

3.6.2 Photosynthesis-Irradiance Curve

The response of the photosynthetic rate to increasing PAR is non-linear due to variation in the quantum yield. A number of different formulations have been proposed to describe the photosynthesis-irradiance curve (referred to as the P-E curve or P-I curve). In general though, it's shape is determined by its initial slope α and its maximum value $P_{\max}$ (the assimilation number), occurring at saturating light levels.

Alternative initial slopes α_{d} and α' apply for downwelling PAR or PUR, such that

$$\mu \approx \alpha \times PAR = \alpha_{\mathrm{d}} \times PAR_{\mathrm{d}} = \alpha' \times PUR \tag{3.18}$$

at low light levels. At vanishing light levels, the yield ϕ has a maximum value $\phi_{\max}$, giving one of the following expressions for the initial slope.

$$\alpha = \frac{1}{P_{\mathrm{C}}} \times \bar{a^{\star}} \times Chl \times \phi_{\max} \tag{3.19}$$

$$\alpha_{\mathrm{d}} = \frac{1}{P_{\mathrm{C}}} \times \bar{a_{\mathrm{d}}^{\star}} \times Chl \times \phi_{\max} \tag{3.20}$$

$$\alpha' = \frac{1}{P_{\mathrm{C}}} \times a_{\max}^{\star} \times Chl \times \phi_{\max} \tag{3.21}$$

from Equations 3.8, 3.14 and 3.17 respectively.

The initial slope of a P-E curve defined with respect to PAR (scalar or downwelling) is expected to vary with depth as the spectrum of the available light changes, changing the fraction of the light energy absorbed by the phytoplankton. The initial slope against PUR has no spectral dependency and is therefore constant with depth (in the absence of other sources of variation).

Each photosynthesis light limitation model is based on a particular P-E curve $J(P_{\max}, \alpha_{\mathrm{d}}, PAR_{\mathrm{d}})$ that parameterizes the effect of light limitation on the photosynthetic rate. The realized photosynthetic rate μ may or may not be equal to J depending on how the ecosystem model handles nutrient limitation. The effect of nutrient limitation may be included in the determination of $P_{\max}$ and/or applied retrospectively.

The PAR_{d} input to J varies strongly and non-linearly with time and depth. Photosynthesis models simulate either the daily mean photosynthetic rate, with some parameterization for the diel irradiance cycle, or the instantaneous rate calculated from point-in-time irradiance. In either case, photosynthesis is averaged over each depth level from 1 to k_{phot}, where k_{phot} is the deepest level with an upper depth above 200 m. For a given depth interval z_1 to z_2 the mean photosynthesis is

$$\bar{J} = \frac{1}{\tau} \int_0^{\tau} \frac{1}{z_2 - z_1} \int_{z_1}^{z_2} J(z,t) \mathrm{d}z \mathrm{d}t \tag{3.22}$$

or

$$\bar{J} = \frac{1}{z_2 - z_1} \int_{z_1}^{z_2} J(z,t) \mathrm{d}z \tag{3.23}$$

where τ is 24 h, z is depth and t is time of day.

3.6.3 Diagnostics

The following photosynthesis sub-model diagnostics are available in the MarMOT full-depth output.

- **kdpar**: diffuse attenuation coefficient for downwelling PAR K_{dPAR} (m^{-1})
- **parfracz**: fraction of surface downwelling PAR at level mid-point
- **parz**: downwelling PAR PAR_{d} at level mid-point (E m^{-2} d^{-1})
- **photmax**: maximum potential photosynthetic carbon fixation rate at present temperature (d^{-1})
- **limdin**: DIN uptake rate limitation factor (0 – 1, 0 for no DIN, 1 for no limitation)
- **pmax**: light saturated carbon fixation rate for P-E curve $P_{\max}$ (d^{-1})
- **alphad**: initial slope of P-E curve for downwelling PAR α_{d} ((E m^{-2})$^{-1}$)
- **pz**: potential carbon fixation J at level mid-point (d^{-1}) (point-in-time model only)

- **pk**: potential level mean carbon fixation $\bar{J}$ (d^{-1})

Two further sub-model diagnostics are provided as scalar output variables:

- **daylen**: day length (d) (daily mean models only)
- **zeuphotic**: euphotic zone depth, defined by 1% surface PAR criterion (m)

The full-depth profile of realized photosynthetic rate μ (d^{-1}) is available as the diagnostic variable **phot**. This is an ecosystem model diagnostic. Although the DIN limitation factor **limdin** is nominally a photosynthesis sub-model diagnostic, it is only treated as such if it affects the P-E curve. In other cases, the factor is applied internally within the ecosystem model.

The distinction between photosynthesis sub-model and ecosystem model diagnostics may be relevant to the time point, within a time step, to which the diagnostic output relates. This depends on the time-stepping options (see Section 3.7) and is discussed in detail in Section 3.8.

3.6.4 Light Limitation Options

The options for light limitation of photosynthesis (option parameter **photmodel**) are

0. no photosynthesis (default)
1. Evans and Parslow 1985 daily mean photosynthesis model
2. point-in-time version of Evans and Parslow 1985 photosynthesis model
3. Platt et al. 1990 daily mean photosynthesis model

The 'no photosynthesis' option is used to suppress the MarMOT photosynthesis calculations in cases where photosynthesis is handled internally by the ecosystem model. Otherwise MarMOT provides the photosynthesis sub-model, making use of applicable information provided by the ecosystem model. The different light-limitation models are described below.

1: Evans and Parslow 1985 daily mean photosynthesis model

The daily mean photosynthesis model of Evans and Parslow (1985) assumes a triangular variation of irradiance over the day. The model is forced by daily mean solar radiation (**solav**). Day length (diagnostic **daylen**) is determined from latitude (**lat**) and time of year. The P-E curve is described by

$$J = \frac{P_{\max}\alpha_{\mathrm{d}}PAR_{\mathrm{d}}}{\sqrt{P_{\max}^2 + (\alpha_{\mathrm{d}}PAR_{\mathrm{d}})^2}}. \tag{3.24}$$

2: point-in-time version of Evans and Parslow 1985 photosynthesis model

The basic P-E curve is the same as in the Evans and Parslow 1985 daily mean model but is integrated in depth only, not time. It is forced by point-in-time solar radiation at the sea-surface (**sol**) so that the diel cycle of photosynthesis is modelled is response to the external forcing.

3: Platt et al. 1990 daily mean photosynthesis model

Photosynthesis is calculated according to the integral approximation of Platt *et al.* (1990) that assumes a sinusoidal pattern of irradiance over the day. The model is forced by daily mean solar radiation (**solav**). Day length (diagnostic **daylen**) is determined from latitude (**lat**) and time of year. The basic P-E curve is of the form

$$J = P_{\max}\left[1 - \exp\left(-\frac{\alpha_{\mathrm{d}}PAR_{\mathrm{d}}}{P_{\max}}\right)\right]. \tag{3.25}$$

3.6.5 Light Attenuation Options

The light attenuation model options (option parameter **kdmodel**) are

0. constant PAR attenuation (default)
1. depth-independent PAR attenuation
2. Anderson 1993 PAR attenuation model with original depth layers

3. Anderson 1993 PAR attenuation model with alternative depth layers

In addition, the option parameter **kdpathopt** can be used to modify the output K_{dPAR} profile by applying a path length adjustment. In the absence of a PAR attenuation model (**kdmodel** $= 0$), the constant PAR attenuation coefficient is given by the option parameter **attenwater**. Descriptions of the other light attenuation options follow.

1: depth independent PAR attenuation model

The attenuation coefficient is given by

$$K_{\mathrm{dPAR}} = k_{\mathrm{water}} + k_{\mathrm{pig}} G \tag{3.26}$$

where k_{water} is the attenuation due to water (option parameter **attenwater**), k_{pig} is the attenuation due to pigment (option parameter **attenpig**) and G is the pigment concentration provided by the ecosystem model. The effect of changes in spectral distribution with depth are not accounted for.

2: Anderson 1993 PAR attenuation model with original depth layers

This is an empirical approximation to the 61 wave-band model of Morel (1988), developed by Anderson (Anderson, 1993) for use in general circulation models. Light penetration is based on a 3 layer model of the attenuation coefficient K_{dPAR}, as a function of a depth-invariant pigment concentration. The effect of variations in the pigment profile above is ignored on the basis that sensitivity of K_{dPAR} to such variations is acceptably low. The three optical layers are divided by layer boundaries at 5 m and 23 m. K_{dPAR} is determined from the local pigment concentration at each depth level, with a maximum at 15 mg m^{-3} (or above). Where model level boundaries do not coincide with optical layer boundaries, K_{dPAR} is depth averaged within levels.

3: Anderson 1993 PAR attenuation model with alternative depth layers

This is an alternative approximation of the 61 wave-band model, based on 3 optical layers with level boundaries at 10 m and 20 m. The approximation method is that of

Anderson (Anderson, 1993) but the coefficients for the polynomial fit are different. It's implementation here reproduces the original HadOCC sub-model without the implied assumption that the top two model levels coincide with the top two optical layers. The original HadOCC version is still available for comparison as part of the internal photosynthesis sub-model (see Appendix C).

Oschlies and Garçon 1999 noon path length adjustment

Setting the option parameter **kdpathopt** to 1 causes the K_{dPAR} profile from the attenuation model to be adjusted, following Oschlies and Garçon (1999), to allow for the effect of the sun's zenith angle on the path length between the surface and a given depth z. From Snell's law, an effective depth for the direct path is given by

$$\tilde{z} = \frac{z}{\sqrt{1 - (\frac{\sin\theta}{1.33})^2}} \tag{3.27}$$

where θ is the noon zenith angle, determined from the environment parameter **lat** and time of year, and 1.33 is an approximate value for the refractive index of water (slightly low for sea water and PAR waveband). The adjusted attenuation coefficient for downwelling PAR is

$$K'_{\mathrm{dPAR}} = \frac{K_{\mathrm{dPAR}}\tilde{z}}{z}. \tag{3.28}$$

The adjustment is based on the assumption that the direct path effect dominates, tending to bias K_{dPAR} high, while use of the noon zenith angle means that path lengths are minimized for the day, tending to bias daily mean K_{dPAR} low. It should be noted that the true effect of zenith angle on the attenuation coefficient is strongly wavelength dependent and decreases with depth (Zheng *et al.*, 2002). The depth dependency is not modelled. With the default option (**kdpathopt** = 0), no adjustment is applied.

3.6.6 Light Absorption Options

The light absorption model options (option parameter **achlmodel**), controlling the variation of chlorophyll-specific absorption of PAR for use by the phytoplankton, are

0. depth independent light absorption (default)

1. Anderson 1993 PAR spectrally-averaged chlorophyll absorption model

By default, the variation of chlorophyll-specific light absorption with depth, due to changes in spectral distribution, is not modelled, so PUR is a fixed fraction of PAR. In addition, there is no model for the spectrally-dependent geometric correction to convert between scalar and downwelling irradiance, so no distinction is made between α and α_{d} or between $\bar{a^{\star}}$ and $\bar{a^{\star}_{\mathrm{d}}}$.

The Anderson 1993 model offers improved accuracy over the default option. However, following the recommendation of Kettle and Merchant (2008), the implementation of a model based on more recent, improved descriptions of the chlorophyll-specific absorption coefficient will be a priority. This would take into account its variation with chlorophyll concentration as well as wavelength.

1: Anderson 1993 spectrally-averaged chlorophyll absorption model

The initial P-E slope α_{d} varies with depth according to a model of the factor

$$\bar{a^{\#}} = \frac{\bar{a^{\star}_{\mathrm{d}}}}{a^{\star}_{\mathrm{max}}} \tag{3.29}$$

that relates it to the spectrally-independent initial slope of the photosynthesis-PUR curve α' (i.e. $\alpha_{\mathrm{d}} = \bar{a^{\#}}\alpha'$). $\bar{a^{\#}}$ is the spectrally-averaged non-dimensional (or normalized) chlorophyll-specific absorption for the phytoplankton with respect to downwelling irradiance. An empirical approximation to a 61 waveband model (Morel, 1988, 1991) is used to determine $\bar{a^{\#}}$ for each model level. $\bar{a^{\#}}$ varies with pigment concentration (in the range 0-15 mg m^{-3}) and depth and the value at each level depends on the values for levels above. The initial slope α', defined at the peak of the chlorophyll absorption spectrum, is taken to be 2.602 times the value of the initial slope α, for scalar PAR, immediately below the surface.

3.7 Simulation Options

[*Input items:* **option**, **timeperiod**]

The options available for controlling a simulation that have not already been covered in the previous sections are described here.

3.7.1 Ecosystem Model

The ecosystem model is selected by number using the option parameter **model**. The models currently available are described in Section 6. Full mathematical descriptions are included in Appendix C.

If no ecosystem model is selected, the simulation will be run, with any initialized tracers, in physics only mode. All tracers are treated identically as passive tracers and transported according to mixing, diffusion and advection processes (if enabled), controlled by the forcing variables **mld**, **vdc** and **w** respectively. Tracer relaxation may also be applied.

3.7.2 Time Stepping

By default MarMOT uses an Euler-forward scheme to solve the generic equation

$$\frac{d\boldsymbol{X}}{dt} = f(X) \tag{3.30}$$

so that

$$\frac{\boldsymbol{X}_{n+1} - \boldsymbol{X}_n}{\Delta t} = f(\boldsymbol{X}_n) \tag{3.31}$$

where n represents the current time step and Δt the length of the time step.

A single time step is split into 5 partial steps (each of length Δt) solved independently in the sequence below.

1. ecosystem model step (including photosynthesis-sub model)
2. advection step
3. diffusion step
4. boundary layer mixing step
5. relaxation step

Autonomous vertical velocities such as those of sinking particles may either be handled in the ecosystem model step or externally in the advection step, according to model-specific options (see Section 6).

The tracer-dependent fluxes for the ecosystem model step are determined from the end-of-time-step tracer concentrations from the previous time step. This convention allows individual terms in the ecosystem model equations to be calculated, if needed, from the end-of-time-step simulation output. The ecosystem model step is executed first and the resulting tracer concentrations are then updated in sequence by the advection, diffusion, mixing and relaxation steps (with $\boldsymbol{X}_n$ taken from the previous partial step).

Sometimes, a particular ecosystem model parameterization dictates applying advection and diffusion before the ecosystem model step, causing the standard sequence to be over-ridden. Fluxes for the ecosystem model step are still determined from end-of-time-step concentrations in such cases.

The time step Δt is 1 hour by default. To change this, the number of time steps per day can be set by the option parameter **nstepday**. To improve accuracy in the solution of the ecosystem model equations, the ecosystem model step (in the standard sequence only) can be broken down into a number of biology sub-steps by setting the option parameter **nstepbio** to a value greater than 1. The photosynthesis sub-model (if external to the ecosystem model) is executed at the beginning of the time step to determine the light-limited concentration-specific photosynthetic rate for the phytoplankton, which then remains constant for the duration of the time step. Other tracer-dependent factors are re-evaluated at each biology sub-step, so they become inconsistent with the tracer concentrations in the end-of-time-step output.

Leapfrog Scheme

An optional leapfrog time-stepping scheme with Robert-Asselin filter (Robert, 1966; Asselin, 1972) is selected by setting the option parameter **leapfrog** to 1. The scheme is described by

$$\frac{\boldsymbol{X}_{n+1} - \bar{\boldsymbol{X}}_{n-1}}{2\Delta t} = f(\boldsymbol{X}) \tag{3.32}$$

$$\bar{\boldsymbol{X}}_n = \boldsymbol{X}_n + \alpha(\boldsymbol{X}_{n+1} - 2\boldsymbol{X}_n + \bar{\boldsymbol{X}}_{n-1}) \tag{3.33}$$

The value of α is given by the option parameter **tfparm**. The 5 partial steps are evaluated over the double time step $2\Delta t$, to get the simulation state $\boldsymbol{X}_{n+1}$. The ecosystem model step of length $2\Delta t$ is broken down into **nstepbio** biology Euler sub-steps if requested. The end-of-time-step output, available at Δt intervals, is the filtered value $\bar{\boldsymbol{X}}_n$.

In practice, there is some variation in the time level of the state $\boldsymbol{X}$ used to determine f. Conceptually, $\boldsymbol{X} = \boldsymbol{X}_n$ is preferable. This convention is used for determination of the tracer effect on the light field for photosynthesis. However, to support multiple biology sub-steps and simplify implementation of new ecosystem models, a convention $\boldsymbol{X} = \bar{\boldsymbol{X}}_{n-1}$ is used initially for the main ecosystem model step, with updates for each sub-step. Any tracer concentrations affecting the photosyntheis sub-model other than via the light field are also represented by $\bar{\boldsymbol{X}}_{n-1}$ (though not updated in sub-steps unless an internal photosynthesis sub-model is used). Advection and diffusion steps are also implemented with $\boldsymbol{X} = \bar{\boldsymbol{X}}_{n-1}$.

When the leapfrog scheme is selected, the forcing data normally interpolated to the middle of the output time step before use is instead interpolated to the middle of the double time step, which is the end of the output time step. Reference profiles for tracer relaxation, normally interpolated to the end of the output time step before use, are instead interpolated to the end of the double time step, which is one standard time step Δt ahead of the output time step. If relaxation rates are set to half **nstepday**, tracers are fully replaced by the reference data at the end of each double time step. In the case of time-varying reference data the result is that the output will be a filtered version of the reference tracer field.

3.7.3 Simulation Time Period

The time period for a simulation is set by the time period parameter set item. The parameters **startyear** and **startday** define the start time of the simulation and **finyear** and **finday** define the end time. By default, each simulation will start at the beginning of a day. The parameter **startstep** can be used to specify a different

time of day by skipping *startstep* - 1 model steps and initializing at the beginning of the step specified.

By default both **startyear** and **finyear** are both 1. These parameters will therefore need to be set explicitly to be compatible with any time axis parameter sets in which the base year is later than year 1. The parameter **startday** also defaults to 1. **finday** defaults to the year length, which is either fixed or dependent on **finyear** according to the time axis configuration (see Section 3.2).

3.8 Simulation Output and Diagnostics

[*Output tables:* **outt**, **outkt**, **outtday**, **outktday**]

Output tables and particular variables within output tables are selected or deselected for each experiment as described in Chapter 4. Headers are written for each required output table by GFAn prior to function evaluation. Each header comprises one comment line (ignored by Table Toolbox programs) describing each selected variable, followed by the variable name record. Data records are then added during the model runs as the data become available. In the profile output tables, separate data records are output for each level in order of increasing depth. Output buffers are flushed at each time step so that progress can be monitored in real time if necessary. Data for different simulation cases are output in sequence and distinguished by the case variable values.

End-of-timestep output tables include an additional leading record or set of depth-dependent records for each simulation case. These data define the initial state in terms of the values of the tracers and composition ratios at the simulation start time. Only data relevant to the current simulation are represented. These may include prognostic and diagnostic variables and possibly external data, such as composition ratios, supplied for diagnosing un-modelled tracer values.

3.8.1 Variables

The complete list of variables available, together with their descriptions and units, are given in the template files for the variable selection tables (see Chapter 4). They include a number of derived tracers and other diagnosed variables that are useful for budgeting purposes. In particular, the total concentrations for each element (carbon and nitrogen) are calculated as derived tracers where possible, together with the horizontal fluxes of these elements associated with vertical divergence.

Any additional fluxes due to imposed relaxation forcing are output as separate diagnostics. The two types of fluxes are also available for the individual primary tracers. Vertical integrals are available for all tracer concentrations and for the elemental fluxes.

Variables appear in the output tables in the following order.

- case variables
- time variables
- depth variables
- scalar forcing variables (in scalar output only)
- profile forcing variables – non-tracer
- profile forcing variables – reference tracer concentrations
- primary tracers
- composition ratios
- derived tracers
- profile diagnostics for photosynthesis sub-model
- profile diagnostics for ecosystem model
- horizontal element fluxes
- horizontal primary tracer fluxes
- elemental relaxation fluxes
- primary tracer relaxation fluxes
- scalar diagnostics for photosynthesis sub-model (in scalar output only)
- scalar diagnostics for ecosystem model (in scalar output only)
- vertical tracer integrals, primary and derived (in scalar output only)
- vertical integrals of elemental fluxes (in scalar output only)

The values of any selected variables not relevant to the current simulation are represented as missing data in the output tables, irrespective of whether they are available in the input data.

Time is represented in 4 different ways if all variables are selected. A particular time is specified by (i) the value of dimension variable **t** (giving time relative to the start of the base year), (ii) the dimension variables **year** and **tofyr**, (iii) the variable **year**, the day number (**day**) and the model step number within that day (**step**) or (iv) the variable **year**, the month (**mth**), the day of the month (**date**) and the time of day in hours (**tofd**).

Depth is represented in the profile output tables by dimension variables **k** and/or **z**. Variables **ztop**, **z** and **zbot** refer to the depth at the top, mid-point and bottom of the level respectively. All variables are level mean values unless otherwise specified in their descriptions. In the scalar output tables, the profile variables are represented by their values for the top level (**k** = 1).

3.8.2 Time of Validity

In the end-of-time-step output tables, the tracer values (reference tracer concentrations / primary tracers / derived tracers / vertical tracer integrals) and the composition ratios pertain to the time specified in each output record (the output time). For the daily mean output tables, these variables are taken to vary linearly during the time-step and averaged accordingly. The remaining variables, comprising the non-tracer forcing fields and various diagnostics, are either constant for the duration of the time step or treated as such for averaging purposes.

For a non-tracer forcing variable, the end-of-time-step output value is the value used during the time-step. This is a general principle too for diagnostics that are intermediate values used in the simulation. In cases not using multiple biology sub-steps or leapfrog time stepping, the output values for such diagnostics are values used throughout the time-step. They are determined using beginning-of-time-step tracer values so are consistent with tracer values in the previous output record. Other diagnostics, specifically primary production (**pprod**) and the biologically driven vertical carbon transport (**ctranbio**), are calculated by the MME from information supplied by the model after the time-step has executed. These are diagnosed from a rate for the time step and an end-of-time-step biomass so are consistent with the biomass tracer values in the current output record.

If multiple biology sub-steps are used (option parameter **nbiostep** > 1) then diagnostics that are intermediate values used in the ecosystem model step relate

to the last sub-step, rather than the beginning of the time step. This does not affect photosynthesis sub-model diagnostics as these are not re-calculated for each sub-step.

If leapfrog time stepping is used, then the photosynthesis sub-model diagnostics pertain to the middle of the double time step, which is the output time. They are therefore consistent with the tracer values in the current output record. Other intermediate-value diagnostics are still based on beginning-of-time-step tracer concentrations, except in the case of multiple biology sub-steps. When there are multiple sub-steps, intermediates are determined using tracer concentrations at the beginning of the last sub-step in the double time step. When **nbiostep** > 2 this is actually ahead of the output time.

3.9 Observation Data Set

[*Input item:* **obset**]

The observation set may contain actual observation data or any other reference data against which to compare the simulations. Any of the variables available in the model output tables (from scalar forcing variables onwards) can be provided as variables in the observation data set. The observation set may also contain a weight variable for each of these observation variables. The weight variable for observation variable $<x>$ is **w.**$<x>$. Its value is used to scale the squared differences between simulated and observed values of $<x>$ in the cost function, as described in the next section. It is typically the reciprocal of an error variance used to normalize the model-data misfit so that the significance of the weighted misfits is consistent with knowledge of uncertainty in the observations.

In the current prototype, observations must be referenced to the **t** dimension, the origin of which can be case-dependent. This is not ideal and the alternative time dimension variables, **year** and **tofyr**, should be recognized in future versions.

Observations may also be referenced to the **z** dimension. A clear distinction is made between observations of scalar variables, which inherently have no depth dependency, and observations of profile variables, which can be depth-dependent. The **z** dimension is optional and can be omitted if an observation set contains only scalar variables. Otherwise the item specification table must contain the **zflag** variable and this must be set to 1 for all instances that contain depth-dependent variables, to indicate the possible presence of **z** values in the item data table.

Each record in the data table defines one, possibly muti-variate observation. Observations must appear in order of increasing time within each instance of the observation set item. Observations with the same time but different depths can appear in any order. Multiple observations at the same time and depth are allowed. In any mixed observation sets, having observations of scalar and profile variables, scalar variable values are only allowed in data table records with blank **z** values (depth-independent records) and profile variable values are only allowed in records with numeric **z** values (depth-specific records).

When comparing simulation and observation data for misfit calculations, MarMOT linearly interpolates the simulation data from level mid-points to depth **z**. For forcing variables **vdc** and **w** that are defined at level boundaries this may be a problem if the observation depths are actual depths. To solve this they would need to be adjusted externally in a grid-dependent way to ensure accurate location matching. However, they are commonly derived from previous MarMOT output on the same grid and in such cases no adjustment is needed.

To use either scalar data or profile data from a previous MarMOT run as a synthetic observation set, the applicable output table can be used as the item data table without any modification of its contents, provided the necessary dimension variables are present. However, a corresponding item specification table must also be present. (The two tables are associated by means of the filename convention described in Chapter 4.) For efficiency, MarMOT detects that the observation depth matches the level mid-point depth (within the floating point precision) and does not attempt to interpolate all the variables.

3.10 Cost Function and Simulation Misfit Data

[*Input item:* **mfitctrl**; *Output table:* **mfit**]

3.10.1 Cost Function Definition

The cost function value J, returned by GFAn as the objective function value, is a weighted average of the squared difference between simulated and observed values (or other reference values in the observation set). Values from the nearest simulation time step are matched to the observation time without interpolation (for efficiency). Profile variable values are interpolated linearly in depth from level mid-points. For observations above or below the extreme mid-point depths within the water column, the mid-point values are used. The misfit cost is then defined by

$$J = \frac{1}{N}\sum_{k=1}^{C}\sum_{i=1}^{n_k}\sum_{j=1}^{m} p_{ijk} w_{ijk} (x_{ijk} - y_{ijk})^2 \tag{3.34}$$

$$N = \sum_{k=1}^{C}\sum_{i=1}^{n_k}\sum_{j=1}^{m} p_{ijk} \tag{3.35}$$

where C is the number of cases, n_k is the number of observations for case k, m is the number of observed variables, x_{ijk} is the simulated value of the jth variable at the ith observation point [1] and y_{ijk} is its observed value. The coefficient p_{ijk} is 1 if the variable is present in the observation set or 0 otherwise. The coefficient w_{ijk} is the weight specified in the observation set or 1 if no weight is given.

If the observation set is the same for all cases, say for an ensemble run at a single site, the cost function can be written

$$J = \frac{1}{N}\sum_{k=1}^{C}\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} w_{ij} (x_{ijk} - y_{ij})^2 \tag{3.36}$$

$$N = C\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} \tag{3.37}$$

The cost function can also be expressed in the simpler form:

$$J = \frac{1}{N}\sum_{i=1}^{N} w_i (x_i - y_i)^2 \tag{3.38}$$

Model-data differences may optionally be calculated in log or square root space, in which case x is replaced by $\log_{10} x$ or $\sqrt{x}$ respectively and y is likewise replaced by $\log_{10} y$ or $\sqrt{y}$. Log transformations emphasize relative error and may be appropriate for variables that tend to exhibit log-normal distributions. However, in ecological

[1] There is no special treatment of multiple observation records occurring at the same time and depth: the simulation is simply evaluated for multiple observation points that happen to be co-located

analyses it is often unclear whether absolute or relative errors should be considered more important. Square root transformations have been applied as a compromise in some studies for this reason (Fasham and Evans, 1995; Evans, 1999). Transformations are applied to variables individually, according to parameters in the misfit control parameter set. To apply a transformation to the variable <x> the parameter **tr.**<x> must be present and set to either 1 for log or 2 for square root. A missing value or 0 indicates no transformation is to be applied.

3.10.2 Misfit Table

The misfit table contains a detailed breakdown of the misfit by individual observation points and variables. Records, each corresponding to one observation point (possibly co-located with others), are numbered sequentially. The observation number (**num**) is followed by the variables **ob.t**, **ob.year** and **ob.tofyr** which describe the time of the observation (the latter part of the variable name matching that of the corresponding variable in the simulation output tables). The variable **z** then gives the observation depth.

The remaining variables are organized as follows.

- model-data time difference (**d.t**)
- model-data differences δ (**d.**<x>)
- weighted misfit $w\delta^2$ (**mf.**<x>)
- model output time variables (**t**, **year**, **day**, **step**, **tofyr**, **mth**, **date**, **tofd**)
- model level (**k**)
- model output matched to observation time x (<x>)
- observation values y (**ob.**<x>)
- misfit weighting factor w (**w.**<x>)

The model-data difference δ is defined by

$$\delta = x' - y' \tag{3.39}$$

where x' and y' are the transformed values of the model value x and the observation value y respectively (equal to x and y under the null transformation). The model level can have a fractional part. This is an interpolation offset, expressed as a fraction of the separation height between the level mid-point above the observation depth and that below.

3.10.3 Weighting Considerations

The appropriate weighting of δ^2 (the un-weighted simulation misfit at the observation point) depends on the experimental aims. Weights are typically used to allow for the uncertainty to be associated with δ as an estimate of some displacement between a modelled state and a real-world state. The desired displacement may be the deviation from the real-world state for a particular simulation, reflecting the performance of the simulation. Alternatively, it may be the deviation from the real-world state for a particular ecosystem model with uncertain inputs, reflecting the performance of the model. The uncertain inputs could include model parameters if the design of the model is being assessed in terms of its structure and formulation (although not parameters that are being controlled as free parameters in an optimization experiment). The likely error in the estimate of the desired displacement thus depends on the context. In addition, it is usually time and space dependent and different for different variables.

In experiments aimed at assessing the simulation there is, by definition, no error in the value x provided by the simulation. The observed value y is taken as an approximation to the true state, denoted x_{true}. In transformed variable space

$$y' = x'_{\text{true}} + \epsilon_{\text{obs}}. \tag{3.40}$$

where ϵ_{obs} is the observation error term. This is the only error term in δ, so the misfit weighting need only depend on the expected variance of ϵ_{obs}. However, if the aim is to assess the performance of the ecosystem model, then errors in its input data must also be considered. x is taken as an estimate of the true ecosystem model simulation state (i.e. the state that would be obtained with the true inputs). So

$$x' = x'_{\text{mod}} + \epsilon_{\text{ext}} \tag{3.41}$$

where x'_{mod} is the transformed true simulation state and ϵ_{ext} is the simulation error (in transformed variable space) at the observation point due to error in external

inputs. The model-data difference δ becomes an estimate of the difference (in transformed variable space) between the true ecosystem model simulation and the true state, with an error $\epsilon_{\text{ext}} - \epsilon_{\text{obs}}$. The relevant error variance for weighting purposes is then the sum of the individual error variances.

In the current version of MarMOT, model-data differences for individual observations can only be weighted independently. In experiments aimed at assessing the simulation this is a relatively minor problem because observation errors can often be treated as independent. In ecosystem model assessment, it becomes a more significant issue; future modification of the cost function to allow for dependencies in time and depth within particular simulation cases is desirable.

Chapter 4

Running MarMOT: the Experiment Control Interface

This section describes how to run non-optimizing experiments with MarMOT. Parameter optimization is covered in Chapter 5.

4.1 Command Line Initiation

MarMOT is executed from the command line and takes as arguments any number of control files. Each control file record defines one experiment to be performed with the application in terms of a set of input and output file names. All records from all control files are appended into a single control table internally before processing. The records are then processed in sequence, reading in new input data only when input file names change between successive records. Progress information is written to a log file (filename: `gfan.log`).

The usage is:

```
marmot [log_file_number] control_file_1 [control_file_2 ...]
```

The optional argument `log_file_number` can be used to supply a suffix of the form $.n$ (where n is an integer) for GFAn to append to the log file name. This is needed to avoid conflict if multiple GFAn jobs are run simultaneously. Any numeric first argument will be treated as a log file number.

Running MarMOT with no arguments causes it to write out a GFAn banner,

followed by some information about the MarMOT application and then prompt for arguments with a usage message. In this interactive mode, the user is given the opportunity to review the control information for each experiment in turn before proceeding.

4.2 Input File Templates

Entering no arguments in response to the MarMOT prompt causes MarMOT to offer to create input file templates. This facility is provided by GFAn, on behalf of the MME application, to simplify the procedure of setting up new experiments.

If requested, the templates are written to files of the form `gfan.template.*` in the current directory. The templates are single record tables in *transposed table format* (see Appendix A) so that variables are in rows. Each template contains all recognized variables for a particular input table but each row is commented out by a leading '`#`' character and will be ignored unless this is removed. The data values are initially set to '`.`' and should be edited as required. Variable descriptions provided by the MME are included as comments.

The last part of the template filename is either `ctrlf` for the control table, `casef` for the case table, <*itemname*>`f` for parameter item or item specification tables, <*itemname*>`dataf` for item data tables or <*outputtablename*>`varf` for variable selection tables. Notes describing MarMOT option codes are written to a separate file: `gfan.template.readme`.

4.3 Output Variable Selection

A variable selection table can be provided for each output table produced by the MME application. The recognized variables in the variable selection table are the same as those available in the corresponding output table. A full list can be obtained from the relevant variable selection table template. In the absence of a selection table, all variables are selected for output by default.

The selection table may or may not include a data record. Any data records present after the first will be ignored. If the data record is absent, then all variables present in the selection table (and not commented out) will be selected for output. If the data record is present, it is expected to contain flag values that indicate whether or not each variable is to be selected. A flag value of 1 causes the variable to appear

in the output. A flag value of 0 causes it to be suppressed (as does omission of the variable). Whether or not data records are used is largely down to personal preference.

4.4 Setting up Experiments: the Control Table

The control table contains one record for each experiment to be performed. Its generation from one or more external control tables allows a high degree of flexibility in the way experiments are sequenced when GFAn is executed.

In MarMOT, all control file variables are optional (although at least one must be present). Each external control table may contain the following variables or variable groups.

- **gfanlogf**: alternative log file name for current experiment
- <*itemname*>**f**: input item table file name
- **casef**: case table file name
- <*outputtablename*>**f**: output table file name
- <*outputtablename*>**varf**: variable selection table file name

With the exception of **gfanlogf**, all of these variables are used to specify names of files to be used for input or output by the MME. For input items that are defined by two external input tables, the filename given is that for the item specification file. The item data file has the same name with the suffix `.dat`. (`.nc` is used instead for NetCDF data or `.nct` for a table of NetCDF data sets.) File names can be full path names so input and output files can be kept in separate directories.

File name variables can be omitted or have missing values for inputs/outputs that are not needed. It is not necessary for concatenated control files to contain matching sets of variables. Blank values of **gfanlogf** divert output back to the main log.

In the first control table record, input filenames must be given explicitly. In subsequent records (in any control table file), the '`.`' character can be used to indicate that the data should be the same as for the previous experiment. File names matching those in the previous record also cause GFAn to use the same data. In either case, the input files are not re-read; the memory-resident data is used instead.

A prefix of '-' added to any input file name specified in the control file causes the normal log output to be suppressed when that input file is read. If added to an output file name, all output to that file is suppressed. Output to particular output tables is also suppressed if their file name fields are blank. By default GFAn logs only the metadata for non-parameter set items. A prefix of '+' added to the input filename causes it to log the data values too.

4.5 GFAn Processing Sequence

Once the internal control table is set up and experiment details confirmed (if necessary), GFAn runs without intervention. For each experiment, GFAn performs the following steps (if applicable).

1. read output variable selection tables
2. create files for output tables and write variable names
3. read data into case table
4. read data into item table(s)
5. create/update cross references between case and item tables
6. perform application-dependent checks on input data
7. evaluate objective function (cost function), writing auxiliary output data to output tables

GFAn writes the control file information to the log file followed by a formatted copy of the input tables read. This may include a case table, parameter item tables, item specification tables and item data tables. Logging of input data items can be suppressed individually to avoid unnecessary output of large volumes of data. Item tables are shown in the log with an additional variable (variable name: <*itemname*>**num**) that gives a sequential item number to each instance. If cross-referencing is successful, a linked version of the case table is written to the log including the numbers of the items linked to each case record. These can be checked to ensure the input items being used for each case are those intended.

A number of progress messages are also written to the log including a final message giving the value of the objective function. This should be zero if no observations are supplied. In interactive mode, these messages are also written to the standard output stream.

Chapter 5

Parameter Optimization

Parameter optimization in MarMOT is performed by the GFAn optimizer, incorporating a genetic algorithm based on David L. Carroll's micro-genetic algorithm code and a local direction set algorithm for function minimization without derivatives designed by Powell (1964). The optimizer features are summarized below.

- Seeks minimum of cost function (the objective function) in free parameter space, over all cases specified
- Free parameters for each optimization experiment are selectable by name from any one input parameter set item
- Genetic algorithm
  - User-definable bounds for each parameter or log-transformed parameter
  - Initialization from input table of parameter vectors or random
- Local non-gradient direction set algorithm
  - User-definable fixed or open bounds for each parameter or log-transformed parameter
  - Repeat searches with different initial parameter vectors from input table or genetic algorithm output
- Optionally outputs all cost function values and corresponding parameter vectors to evaluations table(s)
- Model evaluator output suppressed until completion

5.1 GFAn Optimizer: User Interface

5.1.1 Input Data

For a particular optimization experiment, the optimizable parameter set is selected by number. This is done using an adapted version of the control table driving the experiment control interface described in Chapter 4. The ecosystem model parameter sets are numbered 1 to n, where n is the number of ecosystem models implemented. The remaining parameter sets recognized by the MME application are numbered from $n + 1$ onwards, in the order that the corresponding parameter set items are listed in Section 3.1.3.

If the selected parameter set is one of the ecosystem model parameter sets, the parameter **model** in the MME option item must match its number for all simulation cases. This restriction prevents irrelevant simulations being run that are unaffected by the free parameters.

The action of the optimizer is controlled by 3 input tables:

- *Optimizer configuration table* - provides various parameters for setting up the optimizer. A default configuration is used if omitted.
- *Free parameter table* - specifies parameters to be optimized and ranges if applicable. This table is mandatory if an optimizable parameter set is selected.
- *Parameter initialization table* - provides one or more initial parameter sets for use by optimizer. This table is optional by default but can be mandatory depending on the optimizer configuration.

Details of the variables in these optimizer input tables are given in Sections 5.5, 5.6 and 5.7. Templates for the input files are included with the other input file templates generated on request (see Chapter 4). Separate parameter initialization templates are available for each MME parameter set item.

5.1.2 Output Data

The optimizer can produce any or all of the following output tables.

- *New parameter table* - contains final results: one or more 'optimized' parameter vectors

- *G. A. function evaluation file* - contains results of all cost function evaluations performed by the genetic algorithm, with the corresponding free parameter vectors

- *Powell function evaluation file* - contains results of all cost function evaluations performed by the Powell alogorithm, with the corresponding free parameter vectors

A report file containing free format log information, including population details and statistics for each generation, can also be produced and the optimal parameter vector and corresponding cost function minimum are written to the GFAn log file.

5.2 Optimal Parameter Set Results

If the Powell algorithm is not enabled, the new parameter table contains the n lowest cost parameter vectors from the population of parameter vectors returned by the genetic algorithm, where n is determined by the optimizer configuration, as indicated in Section 5.6 ($n =$ **ga_nout**).

If the Powell algorithm is enabled, the new parameter table contains the parameter vectors at cost function minima located by the Powell algorithm. One vector is returned for each application of the Powell algorithm with a different start point in parameter space. Each of these optimal parameter vectors are output as soon as they are available.

The variables in a new parameter table for N free parameters are

- **psetkey**: parameter vector id.

- *<parametername>* $\times N$: parameter vector values.

- **funcmin**: cost function value at the location given by the parameter vector.

- **n.iter** number of Powell iterations performed to locate the minimum.

- **n.eval** number of cost function evaluations performed by the Powell algorithm.

- **cputime**: total cpu-time used by the Powell algorithm in seconds. [1]
- **time** elapsed time for the Powell algorithm in seconds.

5.3 The Genetic Algorithm

The genetic algorithm is designed to solve a bounded minimization problem using a method based on evolution by natural selection (survival of the fittest). It is a micro-genetic algorithm (Krishnakumar, 1989), the implementation of which is based on David L. Carroll's Fortran Genetic Algorithm Driver code (version 1.7a, 2nd April 2001), freely available from `<http://cuaerospace.com/carroll/ga.html>` (written here in C for interfacing with GFAn).

Each individual parameter vector from a population of such vectors is represented by a string of bits. This is treated as the parameter vector's 'genome'. Each parameter value is represented by a number of bits within the genome string that depends on the desired resolution in the corresponding dimension of the parameter space. The 'fitness' of each individual vector is determined by the smallness of it's associated cost function value. The initial population can be generated randomly or from a user-supplied table of n parameter vectors, where n is the population size defined by the variable **ga_npop** (see Section 5.6).

For a given population, the algorithm generates each individual of a new population by first selecting a parent pair, where each parent is the fittest of two chosen randomly with a shuffling technique, then performing a 'crossover' between the parents, where bits are swapped according to a stochastic scheme. There are two alternative schemes: either bits are swapped individually according to a crossover probability applied to each bit (uniform crossover) or a whole section of the genome starting at a random point is swapped (single-point crossover). Optionally, two child individuals may be generated from each parent pair instead of one. If 'elitism' is enabled then one of the new generation will always be a 'clone' of the fittest individual, rather than a child generated in the normal way.

After each generation, a check is performed to determine the proportion of bits, over the whole population, that differ from those of the fittest individual. If it is less than 5%, the population is considered to be converged. At this point, the algorithm is either terminated, if it has performed at least the minimum number of generations, or re-initialized with the fittest individual and $n - 1$ randomly generated ones.

[1] The alternative un-calibrated time variable **clock** is substituted if the number of clock cycles per second is undefined at compile time.

If an output file is specified, the following output is written to a G. A. function evaluation table every time the cost function is evaluated.

- **generation**: G. A. generation number
- **individual**: Individual number in population of parameter vectors
- <*parametername*> $\times N$: parameter vector values.
- **func**: cost function value at the location given by the parameter vector.
- **cputime**: cumulative cpu-time used by the Powell algorithm in seconds.

The output buffer is flushed between function evaluations so progress can be monitored in real time.

5.4 The Non-Gradient Direction Set Algorithm

Powell's direction set method (Powell, 1964) is used to locate a cost function minimum in the free parameter space of dimension N. This method does not use gradient information and therefore does not require the provision of an adjoint for the cost function. Starting at a given point (a 'first guess'), it involves successive iterations, each consisting of a sequence of minimizations along some set of N straight lines in the parameter space such that each line passes through the previous minimum found. At the end of each iteration, the direction set is updated by substituting the line through the iteration start and end points for one of the old directions. Iteration continues until the function stops decreasing, subject to a small tolerance value.

The version of Powell's algorithm used is that described in Press *et al.* (1992), with reference to Acton (1970). It differs from Powell's quadratically convergent method in that, after each iteration, the old direction discarded is specifically chosen to be that along which the largest decrease in the function was made (rather than the oldest direction in the set). This apparently paradoxical approach avoids the development of linear dependence between directions that can otherwise cause the search to collapse onto a lower-dimensional subspace. Line minimization is performed using Brent's method (Brent, 1973).

The optimization procedure can be repeated with different starting points in parameter space to increase the likelihood of locating a global minimum. Initialization data can be taken from the output of the genetic algorithm or, if the genetic algorithm is not enabled, data can be provided in the parameter initialization table.

The Powell algorithm treats the parameter space as infinite in N dimensions. However, to support bounded minimizations, transformations can be applied to any parameter values p (in original or log space) to provide an unbounded value p' for the optimizer.

$$p' = \left\{ \begin{array}{ll} \frac{p-p_{\rm mid}}{p-p_{\rm lower}} & , \quad p < p_{\rm mid} \\ \frac{p-p_{\rm mid}}{p_{\rm upper}-p} & , \quad p > p_{\rm mid} \end{array} \right\} \tag{5.1}$$

$$p_{\rm mid} = \frac{1}{2}(p_{\rm lower} + p_{\rm upper}) \tag{5.2}$$

where $p_{\rm lower}$ and $p_{\rm upper}$ are the required lower and upper bounds in the old finite parameter space. Transformations are dimension specific, so bounded and unbounded parameters can be optimized simultaneously.

The following output is written to the Powell function evaluation table if an output file is specified.

- **psetkey**: initial parameter vector id.
- **iternum**: Powell algorithm iteration number.
- **dirnum**: Powell algorithm direction number for current line minimization.
- **evalnum**: cost function evaluation number within current line minimization.
- <*parametername*> $\times N$: parameter vector values.
- **func**: cost function value at the location given by the parameter vector.
- **cputime**: cumulative cpu-time used by the Powell algorithm in seconds.

For efficiency, the output buffer is not flushed between each function evaluation.

5.5 Defining the Free Parameter Space

The free parameter table determines the parameters that are selected from the optimizable parameter set to define the optimizer's search space. The optimizable

parameter set item table may contain multiple instances of the parameter set if one or more of the parameters are case dependent. This is not a problem: no special configuration is required to work with a parameter set having a mixture of free parameters and case-dependent parameters.

The characteristics of each free parameter are defined by a separate record in the free parameter table with the following variables.

- **parmname**: parameter variable name
- **logmin**: lower bound for transformed parameter in $\log_{10}$ space (default = 0)
- **logmax**: upper bound for transformed parameter in $\log_{10}$ space (default = 0)
- **min**: lower bound for parameter (default = 0)
- **max**: upper bound for parameter (default = 1)
- **nbits_ga**: number of bits representing parameter value for genetic algorithm (default = 8)
- **powellbounds**: switch for using finite parameter bounds in application of Powell algorithm (default = ON)

All variables except **parmname** are optional. Defaults are applied to all parameters for any omitted variables. The log transform is applied to a particular dimension if one or both log space bounds **logmin** and **logmax** are specified explicitly. Otherwise, by default, **min** and **max** are used and no transformation is applied. Bounds are used by both optimizer algorithms if the **powellbounds** switch is on (**powellbounds** = `ON` or **powellbounds** = 1) or by the genetic algorithm only if the switch is off (**powellbounds** = `OFF` or **powellbounds** = 0).

It should be noted that the order in which parameters appear in the free parameter table affects the behavior of the optimizer and changing the order can potentially change the actual function minima located.

5.6 Optimizer Configuration

The default optimizer configuration causes both algorithms to be enabled, with the Powell algorithm being applied to 5 individual parameter vectors returned by the genetic algorithm. A customized configuration can be supplied by way of an optimizer configuration table, containing any subset of the following variables

- **ga_maxgen**: maximum number of G. A. generations (default = 1000)
- **ga_mingen**: minimum number of G. A. generations (default = 200)
- **ga_npop**: G. A. parameter vector population size (minimum = 2, default = 5)
- **ga_nout**: number of lowest cost parameter vectors output by G. A. (default = 5)
- **ga_nchild**: number of children per parent pair (1 or 2, default = 1)
- **ga_initflag**: switch for G. A. initialization from data table or random (0 or 1, default = 0: random)
- **ga_uniflag**: switch for uniform or single-point crossover (0 or 1, default = 1: uniform)
- **ga_eliteflag**: switch for elitism, i.e. forced copy of best individual to next generation (0 or 1, default = 1: on)
- **ga_pcross**: crossover probability (default = 0.5)
- **ga_seed**: initial random number seed for the G. A. run (negative integer, default = -1000)
- **powel_maxit**: maximum number of Powell iterations (default = 200)
- **powell_maxbr**: maximum number of Brent iterations in each line minimization (default = 100)

5.7 Parameter Vector Initialization

The parameter initialization table can be used to initialize the genetic algorithm population or the Powell algorithm searches. Each record specifies one free parameter vector. The table may optionally contain an id variable (**psetkey**) that identifies different instances of the parameter vector. It must also contain one variable for each free parameter (with variable name matching that for the parameter in the item table). The id variable must be numeric (unlike the item id variables) and is expected to take integer values. If the variable is not provided explicitly then it will be generated to match the sequential record number.

If the genetic algorithm is enabled (**ga_maxgen** $\neq$ 0), its population of free parameter vectors is either defined by the data in the parameter initialization table,

if **ga_initflag** = 1, or random points in the finite parameter space otherwise. The variable **ga_seed** is used to initialize the random number generator.

If the Powell algorithm is enabled (**powell_maxit** $\neq 0$), its initialization depends on whether or not the genetic algorithm has been applied. If it has, then each of it's output parameter vectors is used as a starting point for one application of the Powell algorithm. The parameter vector identifier (**psetkey**) is set to the individual number (**individual**) in the final population from which the output was selected.

If application of the Powell algorithm is not preceeded by a genetic algorithm search and a parameter initialization table is provided, then the Powell algorithm is applied to each initial parameter vector present in the table. In the absence of an initialization table, a single application of the Powell algorithm is applied. This starts from the center of the parameter space (as seen by the optimizer) if parameter bounds are defined. For any unbounded parameters, initial values are instead taken from the parameter set item table. If there are multiple cases, the case 1 value is used.

5.8 Running Optimization Experiments

The following control file variables are used for setting up optimization experiments, in addition to those described in Chapter 4.

- **optpset**: optimizable parameter set number
- **optconfigf**: optimizer configuration table file name
- **freeparmf**: free parameter table file name
- **parminitf**: parameter initialization table file name
- **funcgaf**: G. A. function evaluation table file name
- **funcpowellf**: Powell function evaluation table file name
- **parmnewf**: new parameter table file name
- **reportf**: G. A. report file name

As before, file names can be omitted or have missing values for inputs or outputs that are not needed. The '-' prefix can be used with specific input file names to suppress

log output when they are read and with specific output file names to suppress all output to that file.

If an optimization experiment is immediately followed by another experiment using the memory-resident parameter set that was optimized, the new optimal parameter values are used: the updated parameters are not reset to the values in the input parameter set item table. If this is not the desired action, it can be avoided by using a copy of the original table with a different file name for each experiment. This forces GFAn to reload the data.

In an optimization experiment, GFAn performs the additional steps shown in italics in the sequence below.

1. read output variable selection tables
2. create files for application output tables and write variable names
3. *read data into optimizer configuration and free parameter tables*
4. *create file for new parameter and function evaluation tables and write variable names*
5. *read data into parameter initial value table and check against bounds*
6. read data into case table
7. read data into item table(s)
8. create/update cross references between case and item tables
9. perform application-dependent checks on input data
10. *optimize free parameters, writing results to new parameter table and objective function (cost function) values to function evaluation tables*
11. evaluate objective function (cost function) for optimal parameter vector, writing auxiliary output data to application output tables

The optimizer input tables are logged as they are read in and results are written to the log in the form of a final free parameter table that includes the optimal parameter values and the corresponding objective function value.

Chapter 6

Available Ecosystem Models

The ecosystem model options (option parameter **model**) are

0. no ecosystem model

1. Oschlies & Garçon 1999 NPZD model (OG99NPZD)

2. Hadley Centre Ocean Carbon Cycle Model (HadOCC)

The models are described below in terms of their input and output data. Full descriptions are given in Appendix C. Process parameterizations vary between models and each ecosystem model has its own parameter set, although some parameters are common. Common parameters are given the same external names in MarMOT and expressed in the same units.

The parameter sets are defined in the tables in this section, together with default parameter values and logical restrictions applying to the value of each parameter These restrictions often allow parameter ranges to be much wider than the physically plausible ranges for the real-world values that the parameters are intended to represent. This is entirely appropriate in the context of analyzing experimental models of complex systems, since extraordinary parameter values may compensate for un-modelled or poorly represented processes. Each parameter table also gives the internal representation in terms of the variable names in the Fortran code. Relationships between parameters in different models are discussed in Section 6.3.

Table 6.1: OG99NPZD model parameters

Parameter	Description	Value in Default Set	Restriction	Internal Representation
rparsol	ratio of PAR to total downwelling solar irradiance at sea surface	0.43	0–1	parbio
rphypig	ratio of phytoplankton nitrogen to total pigment (mol N g^{-1})	0.5	> 0	rphypig
aphotmax	max. photosynthetic rate at 0 °C (d^{-1})	0.6	> 0	abio
bphotmax	max. photosynthesis - base for temperature variation factor	1.066	> 0	bbio
cphotmax	max. photosynthesis - temperature sensitivity of exponent ($°C^{-1}$)	1		cbio
alpha	initial slope of photosynthesis-PAR curve ($(E\ m^{-2})^{-1}$)	0.063[1]	≥ 0	alphabio $\times 2.52$
kdin	half-saturation conc. for DIN uptake (mmol m^{-3})	0.5	> 0	rk1bio
pmort	phytoplankton mortality rate (d^{-1})	0.03	≥ 0	phiphy
gmax	maximum grazing rate (d^{-1})	2	≥ 0	gbio
epsfood	prey capture rate (d^{-1}(mmol N $m^{-3})^{-2}$)	1	≥ 0	epsbio
betap	zooplankton food assimilation efficiency	0.75	0–1	a_npz
zexcr	zooplankton excretion rate (d^{-1})	0.03	≥ 0	d_npz
zmortdd	density dependent zooplankton mortality (d^{-1}(mmol N $m^{-3})^{-1}$)	0.2	≥ 0	phizoo
remin	detrital remineralization rate (d^{-1})	0.05	≥ 0	remina
dsink	detrital sinking rate (m d^{-1})	5	≥ 0	w_detr

[1]equivalent to 0.025 $d^{-1}(W\ m^{-2})^{-1}$ at sea-surface where 1 W $= 2.77 \times 10^{18}$ quanta s^{-1} for PAR (Morel and Smith, 1974) so 1 E $d^{-1} = 2.52$ W

6.1 Ecosystem Model 1: OG99NPZD

This model is the 4 compartment nitrogen model described by Oschlies and Garçon (1999). The nitrogen tracers are variables **din**, **phy**, **zoo** and **det**. The model parameter set is described in Table 6.1. For full details of process parameterizations, see Appendix C.

Initial values are required for each tracer (if not zero). Values for the composition ratios can be provided for generating diagnostic carbon and chlorophyll variables (**phyc**, **zooc**, **detc** and **chl**) but are not used by the model or updated during the simulation. These composition ratios would normally be depth-invariant to avoid spurious creation and destruction of carbon or chlorophyll as a result of nitrogen transport. However, the possibility of providing depth-dependent values is not excluded.

The forcing variables applicable to this model are either **sol** or **solav** (depending on the photosynthesis light limitation model; see Chapter 3, Section 3.6.4), **mld**, **vdc**, **w** and **temp** plus the reference profile variables for each tracer. The temperature **temp** affects the phytoplankton maximum photosynthetic rate. Other forcing variables are handled externally by MarMOT and have no model-specific effect.

6.2 Ecosystem Model 2: HadOCC

This model is a version of the Hadley Centre Ocean Carbon Cycle Model of Palmer and Totterdell (2001). It is an NPZD model carrying dissolved inorganic carbon (DIC) and alkalinity as additional tracers coupled to the nitrogen dynamics for determination of carbon fluxes. The carbon:nitrogen (C:N) ratio for each organic compartment is fixed.

The implemented version of the model is based on code provided by the UK Met Office in 2005 (I. Totterdell & R. Barciela, pers. comm.). It differs from that described by Palmer and Totterdell (2001) in having an option for a dynamic phytoplankton carbon:chlorophyll (C:Chl) ratio, representing acclimation of the photosynthetic apparatus to the available light and DIN according to the balanced growth model of Geider *et al.* (1997). In addition, it incorporates the light penetration and photosynthesis model of Anderson (1993) as an optional internal photosynthesis submodel and includes some modifications to the pathways of material resulting from grazing and mortality. The Wanninkhof (1992) wind-dependent gas exchange formulation used by Palmer and Totterdell (2001) is not presently available in MarMOT and a constant transfer velocity for air-sea exchange of pCO_2 is used instead.

The primary tracers are variables **din**, **phy**, **zoo**, **det**, **nh4**, **dic** and **alk**. Ammonium (**nh4**) is included to track the regenerated fraction of DIN (**din**) for diagnostic purposes and does not affect the other prognostic variables. The difference between DIN and ammonium can be interpreted as nitrate. The model also updates the derived tracer **chl** (phytoplankton chlorophyll).

The model parameter set is described by Tables 6.2 and 6.3. For process parameterizations, see Appendix C. The version of HadOCC implemented includes the following options. The relevant option variable in the HadOCC parameter set is shown in brackets.

- Fixed or dynamic phytoplankton C:Chl ratio (**rcchlopt**)

Table 6.2: HadOCC model parameters

Parameter	Description	Value in Default Set	Restriction	Internal Representation
rcchl	C:Chl ratio if fixed (g C (g Chl)$^{-1}$)	40	> 0	cchl_fixed
rcchlmin	minimum C:Chl ratio (g C (g Chl)$^{-1}$)	20	> 0	cchl_min
rcchlmax	maximum C:Chl ratio (g C (g Chl)$^{-1}$)	200	$\geq$ **rchlmin**	cchl_max
rcnphy	phytoplankton C:N ratio	6.625	> 0	c2n_p
rcnzoo	zooplankton C:N ratio	5.625	> 0	c2n_z
rcndet	detrital C:N ratio	7.5	> 0	c2n_d
rparsol	ratio of PAR to total downwelling solar irradiance at sea surface	0.43	≥ 0	par
rchlpig	ratio of chlorophyll to total pigment	0.8	> 0	chl2pig
photmax	maximum photosynthetic rate (d^{-1})	2	≥ 0	psmax
alphachl	initial slope of photosynthesis-PAR curve (g C (g Chl)$^{-1}$ (E m^{-2})$^{-1}$)	5.556	≥ 0	alpha $\times 10^6$ / 3600
kdin	half-saturation conc. for DIN uptake (mmol N m^{-3})	0.1	> 0	grow_sat
presp	phytoplankton respiration rate (d^{-1})	0.05	≥ 0	resp_rate
pmortdd	density dependent phytoplankton mortality (d^{-1} (mmol N m^{-3})$^{-1}$)	0.05	≥ 0	pmort_rate
pminmort	threshold for phytoplankton mortality (mmol N m^{-3})	0.01	≥ 0	phyto_min
fpmortdin	fraction of phytoplankton mortality to DIN	0.01	0–1	F_nmp
gmax	maximum grazing rate (d^{-1})	0.8	≥ 0	graze_max
holling	Holling function exponent for grazing model (integer n)	2	+ve integer	holling_coef
epsfood	prey capture rate (d^{-1} (mmol N m^{-3})$^{-n}$)	3.2	≥ 0	graze_max / (graze_sat)^n
fmingraz	food threshold for grazing function (mmol N m^{-3})	0.01	≥ 0	graze_threshold
fingest	fraction of grazed material ingested	0.77	0–1	F_ingest
betap	zooplankton assimilation efficiency for phytoplankton	0.9	0–1	beta_p
betad	zooplankton assimilation efficiency for detritus	0.65	0–1	beta_d
fmessyd	fraction of messy feeding to detritus	0.1	0–1	F_messy
zmort	linear zooplankton mortality (d^{-1})	0.05	≥ 0	z_mort_1
zmortdd	density dependent zooplankton mortality (d^{-1} (mmol N m^{-3})$^{-1}$)	0.3	≥ 0	z_mort_2
fzmortdin	fraction of zooplankton mortality to DIN	0.67	0–1	F_zmort
nitrifeuph	nitrification rate of ammonium in euphotic zone (d^{-1})	0	≥ 0	rndecay_euphotic
nitrifaph	nitrification rate of ammonium in aphotic zone (d^{-1})	0.03333	≥ 0	rndecay_aphotic
dsink	detrital sinking rate (m d^{-1})	10	≥ 0	sink_rate_dt
rco3pprod	carbonate precipitated per unit primary production	0.013	≥ 0	rain_ratio

Table 6.3: Option variables in HadOCC model parameter set

Parameter	Description	Value in Default Set
rcchlopt	C:Chl option (0: fixed, 1: dynamic)	1
chltracer	chlorophyll tracer option (0: off, 1: on)	0
co2sys	carbonate system option (0: off, 1: on)	1
nh4tracer	ammonium tracer option (0: off, 1: on)	1
dsinkopt	implementation of detrital sinking (0: external, 1: internal)	1
photopt	implementation of photosynthsis (0: external, 1: internal)	0
vsupply	vertical nutrient supply within biology step (0: off, 1: on)	0

- Ammonium tracer on/off (**nh4tracer**)
- Carbonate system on/off (**co2sys**)
- Internal or external handling of detrital sinking (**dsinkopt**)

Switching the carbonate system off removes the tracers **din** and **alk**. Internal handling of sinking detritus implies model-specific treatment of detritus reaching the bottom of the water column. Three further option variables affect the numerical solution:

- Internal or external photosynthesis sub-model (**photopt**)
- Chlorophyll tracked as a tracer or derived from local C:Chl (**chltracer**)
- Activated or de-activated vertical DIN supply within biology step (**vsupply**)

If **chltracer** = 0, chlorophyll is not transported as a tracer. Instead its concentration is calculated at each time step from phytoplankton nitrogen using the local C:Chl ratio from the last ecosystem model step. C:Chl ratio is kept homogenous over all levels above the mixed layer depth (**mld**), simulating the effects of rapid mixing of the phytoplankton, but is not updated between ecosystem model steps by the external physical transport processes. If **chltracer** =1, chlorophyll is transported as a conservative tracer and the C:Chl ratio updated accordingly between time steps. Note that **chltracer** does not affect the solution if the C:Chl ratio is fixed (parameter **cchlopt** = 0).

If **photopt** = 1, the internal photosynthesis sub-model is used. It is equivalent to the external sub-model defined by option parameter values **photmodel** = 3, **kdmodel** = 3, **kdpathopt** = 0 and **achlmodel** = 1. However, the light attenuation model is applied with the assumption that the top two model levels coincide with the top two optical layers (i.e. 0 – 10 m and 10 – 20 m). If this is not the case, or if multiple biology sub-steps are used (**nstepbio** > 1), the simulation results will differ. In any case, minor differences in photosynthesis at the bottom of the euphotic zone can occur because the internal sub-model explicitly sets the available light to zero at the bottom of the deepest level for which photosynthesis is calculated.

If **vsupply** = 1, changes in DIN (and its ammonium fraction if modelled) due to vertical advection and diffusion are taken into account within the ecosystem model step. This feature is included in 3-D HadOCC runs to cater for the situation, exacerbated by long timesteps, where DIN can otherwise be used up intermittently despite there being an adequate vertical supply. It does cause the advection and diffusion steps to be performed before the ecosystem model step, reversing the normal MarMOT convention described in Chapter 3, Section 3.7.

Initial values are required for each primary tracer modelled (if not zero). Also, if the dynamic C:Chl option is active (parameter **cchlopt** = 1), the initial **rcchl** is needed. The model parameter **rcchl** will be used as a default for any missing values. Conceptually, initial values for **nh4** should be less than or equal to **din** throughout the water column but this is not enforced.

The applicable forcing variables are either **sol** or **solav** (depending on the photosynthesis light limitation model; see Chapter 3, Section 3.6.4), **mld**, **vdc**, **w**, **temp**, **sss** and **pco2atm** plus the reference profile variables for each tracer. The variables **temp**, **sss** and **pco2atm** are only applicable if the carbonate system is modelled (**co2sys** = 1) and **temp** is then only required for the top level (so can be provided as a scalar). These variables are used in the calculation of surface pCO_2 and air-sea CO_2 flux. Either **sol** or **solav** are used in determining the C:Chl ratio (if dynamic). **mld** and either **sol** or **solav** are used for selecting the appropriate nitrification rate of ammonium (if modelled). **mld** is also used for homogenizing boundary layer C:Chl within the ecosystem model step (if C:Chl is dynamic). **vdc** and **w** have no model-specific effect.

WARNING: In the current MarMOT implementation, there are no DIC or alkalinity updates in response to changes in salinity. Updates to these tracers would normally be expected for consistency to avoid spurious changes in air-sea CO_2 flux. It is therefore sensible to hold **sss** constant for the majority of experiments and it is best provided as an environment parameter.

6.3 Commonality and Diversity in Process Parameterizations

This section highlights the differences in the parameterizations of nitrogen cycle processes in the two ecosystem models. It also indicates the parameters that are either common to both models or directly related. Full descriptions of each model are provided in Appendix C.

6.3.1 Optics

The PAR fraction of total solar radiation at the sea-surface (parameter **rparsol**) is common to both ecosystem models. This relates the downwelling PAR immediately below the sea-surface to the total downwelling shortwave radiation immediately above the sea surface and so includes an implicit attenuation for reflection at the interface.

The pigment concentration G required for the light attenuation and absorption models is derived from phytoplankton nitrogen concentration in OG99NPZD or from chlorophyll concentration in HadOCC:

$$G = \frac{\mathbf{phy}}{\mathbf{rphypig}} \tag{6.1}$$

or

$$G = \frac{\mathbf{chl}}{\mathbf{rchlpig}} \tag{6.2}$$

in OG99NPZD and HadOCC respectively. The phytoplankton nitrogen to pigment ratio parameter **rphypig** is related to the chlorophyll to total pigment ratio parameter **rchlpig** by

$$\mathbf{rphypig} = \mathbf{rchlpig} \times \frac{\mathbf{rcchl}}{12.01 \times \mathbf{rcnphy}} \tag{6.3}$$

where 12.01 is the average atomic weight of carbon in the natural environment. Equivalent values of **rphypig** and **rchlpig** can thus be prescribed for both ecosystem models if the HadOCC C:Chl ratio (**rcchl**) is fixed.

6.3.2 Primary Production

There are a number of differences in the way primary production is modelled in the two ecosystem models. These are described below.

Firstly, the maximum photosynthetic rate is temperature dependent in OG99NPZD:

$$\mathbf{photmax} = \mathbf{photmaxa} \times \mathbf{photmaxb}^{(\mathbf{photmaxc} \times \mathbf{temp})}. \tag{6.4}$$

In HadOCC it is a fixed model parameter.

The interaction of DIN and light limitation of photosynthesis is handled differently in the two models. In OG99NPZD, light-limited and DIN-limited photosynthetic rates are calculated separately and the minimum rate is used (so for the light limitation model, **pmax** = **photmax** and the realized rate **phot** $\leq$ **pk**). In HadOCC, DIN limitation is applied to the light-saturated rate in the light limitation model (so **pmax** $\leq$ **photmax**). In both models, the DIN limitation factor (diagnostic **limdin**) is determined from the DIN concentration and the model parameter **kdin** using a common functional form. Normally, the realized photosynthetic rate in HadOCC is that from the light limitation model (**phot** = **pk**). However, a further timestep-dependent DIN limitation can occur, in the event that the calculated rate would cause all available DIN to be used up in less than 1 timestep. HadOCC then reduces the rate to reflect the additional limitation.

In the OG99NPZD model, the initial slope of the P-E curve is defined by the model parameter **alpha**. This is interpreted as the value of α (the slope with respect to scalar PAR) immediately below the sea surface. In HadOCC, the equivalent sea-surface value is given by

$$\mathbf{alpha} = \frac{\mathbf{alphachl}}{\mathbf{rcchl}} \tag{6.5}$$

The model parameter **alphachl** is equivalent to the product $\bar{a^\star}\phi_{\max}$ in Equation 3.19.

6.3.3 Phytoplankton Loss to Mortality and Respiration

Biomass-specific mortality of phytoplankton is linear in OG99NPZD but density dependent (quadratic) in HadOCC. All of this mortality flux goes to detritus in

OG99NPZD. In HadOCC, all the mortality nominally goes to detritus if the model parameter **fpmortdin** = 0. Otherwise, a fraction **fpmortdin** is diverted to DIN. If detritus is nitrogen poor, relative to phytoplankton, (i.e. **rcndet** > **rcnphy**, includes default) then an additional fraction of the mortality goes to DIN to maintain the constant C:N ratio of the detritus.

In HadOCC, there is an additional loss of phytoplankton to DIN for non-zero values of the respiration rate parameter **presp**. This loss is a consequence of the carbon loss due to respiration and the fixed phytoplankton C:N ratio. It is not intended to represent actual respiration of nitrogen (Palmer and Totterdell, 2001).

6.3.4 Grazing and Zooplankton Production

In OG99NPZD, zooplankton graze only on phytoplankton. In HadOCC, they graze on both phytoplankton and detritus according to the relative concentrations of each in terms of biomass. The maximum grazing rate and prey capture rate parameters **gmax** and **epsfood** are common to both models. If **holling**=2 (Holling type III function) and **fmingraz**=0 in HadOCC, the parameterization of grazing is the same in both models. In the absence of detritus, the grazing rate is then identical if **rcnphy** = **rcnzoo**.

Modelled grazing rates in d^{-1} are taken to be biomass-specific, rather than nitrogen-specific, whereas fluxes are modelled in terms of nitrogen. This is significant in HadOCC if **rcnphy** differs from **rcnzoo**. In such cases (which include the default), the loss rate of phytoplankton nitrogen in HadOCC due to grazing (in the absence of detritus) differs from that in OG99NPZD by a factor

$$\frac{B_\mathrm{P}}{B_\mathrm{Z}} = \frac{14.01 + 12.01\theta_\mathrm{P}}{14.01 + 12.01\theta_\mathrm{Z}},$$

where θ_P = **rcnphy**, θ_Z = **rcnzoo** and 14.01 and 12.01 are average atomic weights for nitrogen and carbon respectively. B_P and B_Z are ratios of biomass to nitrogen for phytoplankton and zooplankton. Biomass is assumed to be proportional to the sum of the masses of nitrogen and carbon present (Palmer and Totterdell, 2001).

HadOCC includes a parameterization of messy feeding. However, if **fingest** = 1, ingestion rate is equal to grazing rate, as it is in OG99NPZD.

The assimilation parameter **betap** for ingested phytoplankton is common to both models. However, in HadOCC the nitrogen assimilated is adjusted in cases where

the C:N ratio of the food potentially assimilated is less than the zooplankton C:N ratio. Nitrogen not assimilated is then diverted to DIN [1]. This does not occur with the default parameter set, which specifies a relatively high nitrogen content for zooplankton. It cannot occur if **rcnzoo** is less than both **rcnphy** and **rcndet**.

All egested nitrogen goes to detritus in OG99NPZD. The fate of egested nitrogen differs in HadOCC if **rcnphy** < **rcndet**. In such cases (which include the default), a fraction is diverted to DIN.

6.3.5 Zooplankton Losses: Mortality and Excretion

Biomass-specific mortality of zooplankton is density dependent (quadratic) in both models (with common parameter **zmortdd**. However, HadOCC includes an additional linear term if **zmort** is non-zero. All zooplankton mortality goes to detritus in OG99NPZD. In HadOCC, all the mortality nominally goes to detritus if the model parameter **fzmortdin** = 0. Otherwise, a fraction **fzmortdin** is diverted to DIN. It is not unreasonable for this fraction to be large, reflecting excretion and other losses to DIN consistent with a chain of higher predators that are not explicitly modelled (Fasham, 1990). If detritus is nitrogen poor, relative to zooplankton, (i.e. **rcndet** > **rcnzoo**, includes default) then an additional fraction of the mortality goes to DIN to maintain the constant C:N ratio of the detritus.

OG99NPZD does include a linear excretion term for zooplankton such that the excretion rate parameter **zexcr** is functionally equivalent to **zmort** in HadOCC. All excretion goes to DIN.

6.3.6 Particle Sinking

The process of detrital sinking is the same in both models with the sinking rate determined by the common parameter **dsink**. The default action for detritus reaching the sea-bed is for it to be immediately re-suspended, adding to the concentration in the bottom level. The fate of this settling detritus is different if handled internally by HadOCC (option parameter **dsinkopt** = 1). In that case, the material is distributed over the bottom 3 levels.

[1] This could be interpreted as excretion of assimilated nitrogen.

6.3.7 Remineralization

OG99NPZD uses a constant remineralization rate for the breakdown of detritus to DIN (model parameter **remin**). In HadOCC, the rate is constant down to 80.58 m, at 0.1 d^{-1}, and then decreases as depth increases. The rate and reference depth parameters are hard-coded in the present version but should ideally be included in the model parameter set.

Appendix A

GFAn Data File Format and Limits

Input and output files contain ASCII data, tabulated with variables in columns and data records in rows. This is the *standard table format*. An alternative *transposed table format* is also recognized for single record input tables. This format is easier to view and edit for such tables, especially those with a large number of variables.

Generic formatting rules:

- Lines are terminated by the NEWLINE character ('`\n`').
- A '`#`' character and any subsequent text on a line is treated as a comment and ignored.
- All non-blank lines must contain the same number of fields, separated by one or more white space characters: normally SPACE (' ') and/or TAB ('`\t`').
- The first character of each variable name must be non-numeric.
- Id variables can have numeric or non-numeric values. Other variables must be numeric.
- Missing data values (numeric/non-numeric) are represented by a single underscore ('`_`').

Formatting rules for standard table format:

- The first non-blank line is a header line containing the variable names.
- Each remaining non-blank line contains 1 data record.

Formatting rules for transposed table format:

- The first non-blank line contains the 2 upper case fields '`NAME`' and '`VALUE`'.
- The first column contains the variable names.
- The second column contains the data record.

If a variable name appears more than once in a file only the first occurrence is processed.

GFAn has internal limits that will prevent it handling exceptionally long lines or an exceptionally large number of variables. Currently these are:

```
LINE_MAX = 20000
NFIELD_MAX = 1000
```

There is also a restriction on the number of records allowed in case tables, parameter item tables, item specification tables and parameter initial value tables. It does not apply to item data tables. The limit is:

```
NREC_MAX = 100000
```

These limits are defined in a GFAn include file `toolbox.h`.

Appendix B

Table Toolbox Utilities

A dedicated suite of programs called Table Toolbox is available for working with ASCII text files of the format (or formats) defined in Appendix A. Each toolbox program (except 'tabview') supports UNIX data piping in ASCII or binary format, allowing complex operations to be built by combining the functions of different tools. No provision is made within Table Toolbox for plotting so data must be imported to another application for visualization.

Toolbox programs are run from the command line with input and output file names and other control information supplied via the argument list. Standard input and/or output streams may be used in place of files by substituting the '`-`' character for the filename in a program's argument list. Certain arguments need to be quoted to protect them from interpretation by the shell. Appending the suffix `.ttb` to an output filename will produce double precision Table Toolbox binary output. Binary piping is achieved by giving `-.ttb` as the output file name. Unless otherwise noted, programs reading binary input files will produce binary output files.

WARNING: output files that already exist are overwritten. (Note that input and output files can be identical. This works fine for small files where the whole file is loaded into the system buffer at once but is not generally recommended).

A subset of the most generally useful programs is described here. Programs are listed below by category. Full descriptions follow alphabetically by program name in Section B.2. Built-in values, referred to in the text by upper case italics, are given in Section B.3.

B.1 Program List

The codes given in brackets have the following meanings.

T – can be applied to input data in transposed table format.
K – processes data according to values of one or more key variables.
G – can process data records in groups defined by one or more key variables.

Key variables are special variables used by certain programs for identifying data records. Any variable present can be specified as a key variable at run-time. The list of key variables is separated from the list of variables to process by the word '`by`' in the argument list. A new group is indicated by a change in value of one or more of the key variables specified. Numerical key variables are compared as numbers rather than strings, to *SIGDIG_DEF* significant digits. Any extra digits will be lost in the output table. Variables are re-arranged so that the key variables appear first in the output.

For viewing data:

tabview displays data in page layout form with numbered records (T)

tabsqz squeezes data into equal width columns; converts binary files to text (T)

For handling or creating variables:

tabcopy copies variables by name (T)

tabcopyn copies variables by name or creates them if missing

tabdrop excludes variable by name

tabcalc generates new variables by evaluating numerical expressions

tabvname re-names variables

For selecting or organizing data records:

tabget extracts an individual record for output in column record format (T)

tabsel selects records by data value or range

tabenum enumerates data records (G)

tabsort sorts data records (K, G)

For dividing or combining tables:

tabsplit splits data into multiple tables according to key variable values (K)

tabcat concatenates tables (T)

tabcatn concatenates tables, creating missing variables as needed

tabmerge merges sorted tables on key variables (K)

tabjoin joins variables from two tables where records match on key variables (K)

For creating data records:

tabseq creates data sequences

tabrep duplicates records

For summarizing or comparing data:

tabcount gets count of non-missing data for each variable (G)

tablim gets lower and upper bounds for each variable (G)

tabmin gets lower bound for each variable (G)

tabmax gets upper bound for each variable (G)

tabmean gets mean value and count for each variable (G)

tabdiff compares data from two tables and outputs numerical differences

B.2 Program Descriptions

Italics are used below to indicate names for which values are substituted. Where words appear in lower case italics in the program descriptions, values should be substituted at run-time. Upper case italics are used to indicate built-in values. Program arguments in square brackets are optional and '...' is used to indicate zero or more repetitions of the previous argument formats.

`tabcalc` *infile outfile* [*var1*`=`*expr1* ...]
or `tabcalc` [*infile*] `=`*expr1* [...]

Generates new variables in the output file by evaluating expressions. Expressions may contain numerical constants, variable identifiers, `+`, `-`, `*`, `/`, `&`, `|`, `=`, `!=`, `<`, `>`, `<=`, `>=`, function names, and other bracketed expressions. The special constant `pi` is also allowed (if there is no variable **pi** in the input). Can be used without an input file by substituting the single character '.' for *infile*. Can also be used with no output file, following the alternative usage form shown above, to produce single un-named output values like an ordinary calculator.

Non-numeric variable values are treated as missing data. The missing data value *NODATA* is returned if the expression cannot be evaluated. Logical expressions return 0 for FALSE, 1 for TRUE. If logical expressions involving non-numeric strings need to be evaluated, 'tabsel' can be used first to convert simple conditions (e.g. `city="London"`) to 0 or 1. Omision of the right hand side in the conditional expressions '`=`' and '`!=`' implies the constant *NODATA*.

Functions supported are: ln, log10, exp, pow, sqr, sqrt, sin, cos, tan, asin, acos, atan, sinh, cosh, tanh, dtr, rtd, ceil, floor, int, frac, abs, min, max, if, not. Angles for trigonometric functions are expressed in radians. Definitions for the less obvious functions:

`dtr(`x`)` returns the value of x degrees in radians.
`rtd(`x`)` returns the value of x radians in degrees.
`ceil(`x`)` returns the smallest integer not less than x.
`floor(`x`)` returns the largest integer not greater than x.
`int(`x`)` returns the integer part of x.
`frac(`x`)` returns the fractional part of x.
`pow(`x`,`y`)` returns x^y.
`min(`x`,`y`)` returns the minimum of x and y.
`max(`x`,`y`)` returns the maximum of x and y.

`if(`*condition*`,`*value1*, *value2*`)` returns *value1* if *condition* is non-zero, otherwise *value2*.
`not(`*condition*`)` returns 0 if condition is non-zero, otherwise 1.

`tabcat` *infile1* [*infile2* ...] *outfile*

All input files are concatenated to the output file (with the header record appearing once only). All variables in *infile1* must be present in all files. If subsequent files include variables not in *infile1* then only those present in *infile1* are copied. The output file format (i.e. text/binary) will match the format of the first input file.

`tabcatn` *infile1* [*infile2* ...] *outfile*

As 'tabcat' but allows missing variables. Output variable names match those of $infile1$ and their values are set to the missing data value if unavailable.

`tabcopy` *infile* [*outfile* [*var1* ...]]

All variables named in the argument list are copied to the output file (or the standard output). If no variable names are given then all variables are copied. All variables must be present in the input file.

`tabcopyn` *infile outfile* [*var1* ...]

As 'tabcopy' but allows missing variables to be specified. These are created with missing data values.

`tabcount` *infile outfile* [`@`] [[`@`]*var1* ...] [`by` *keyvar1* ...]

Outputs the number of non-missing data values for each variable specified in the argument list. All input variables are processed if none are specified. By default only numerical values are counted. If an '`@`' character appears as the 3rd argument non-numeric values are included for all variables (i.e. all values are counted with the

exception of *NODATA_STR* or, in binary files, the value *NODATA*). Alternatively the '`@`' character may be applied to individual variables.

`tabdiff` *infile1 infile2* [*outfile* [`t=`*tolerance*[`%`]] [*var1* ...]]

Outputs the difference *infile1* value minus *infile2* value for all numeric values common to both files (i.e. matching on variable name and record number) or for specified variables only. If the number of records differs between input files, any extra records at the end of the larger file are ignored. There is one output record for each pair of records compared. It contains variables of the form **d.**<*varname*> where <*varname*> is the name of the variable compared. A tolerance may be specified as an absolute difference or a percentage difference. Any differences within the tolerance band are set to zero.

`tabdrop` *infile outfile* [*var1* ...]

All variables are copied to the output file with the exception of any variables named in the argument list.

`tabenum` *infile outfile* [*grp1keyvar1* ... [`by` *grp2keyvar1* ... [`by` ...]]]

Data records are numbered sequentially to generate a new variable **num** in the output file. If key variable(s) are specified the records in each group are enumerated separately and the groups are numbered in a new variable **grpnum1**. Nested groups may be specified by using multiple 'by' clauses, giving new variable(s) **grpnum2** etc. The highest numerical suffix to the variable name refers to the outermost level.

`tabget` *infile* [*outfile* [*recnum*] [*var1* ...]]

Extracts either the first record in *infile* or, if *recnum* is given, the *recnum*th record and outputs it in transposed table format. Outputs selected variables only if a variable list is given.

`tabjoin` [=]*infile1* [=]*infile2* *outfile* [*keyvar1* ...]

Joins *infile1* and *infile2* by key variable(s). One record is output for each pair of records with matching keys. If there are multiple key variables all variables must match. The output record contains the key variable(s) followed by the remaining *infile1* variables and then the remaining *infile2* variables.

Files must be sorted in key order (see 'tabsort' - numbers compared numerically). If no key is specified the key is made up from all variables common to both files, the order of significance of the key variables being their order of appearance in *infile1*. If no variables are common then the number of records in *infile2* is output for every record in *infile1* (WARNING: this can make very large files !!!). An '=' sign preceding an input file name forces the program to output at least one record for every input record even if no matches are found in the other file. In this case the output record will be padded with missing data values.

If one of the input files contains much larger groups than the other then this should ideally be *infile1* which is processed one record at a time, whereas *infile2* is processed one group at a time (the whole group being loaded into memory). The maximum group size for *infile2* is *NREC_MAX*.

The output file will be in binary format only if both input files are.

`tablim` *infile* *outfile* [*var1* ...] [`by` *keyvar1* ...]

Outputs the minimum and maximum numeric values for each variable specified. New variable names are generated by adding **l.** and **u.** prefixes to existing names for lower and upper bounds respectively.

`tabmax` *infile* *outfile* [*var1* ...] [`by` *keyvar1* ...]

Outputs the maximum numeric value(s) for each variable specified in the argument list. All input file variables are processed if none are specified.

`tabmean` *infile* *outfile* [*var1* ...] [`by` *keyvar1* ...]

Outputs the mean(s) and sample size(s) for each variable specified in the argument

list. All input file variables are processed if none are specified. Variable names for the mean are the same as the input names. New variable names are generated for sample size by adding **n.** to the front of the variable name.

`tabmerge` *infile1* [*infile2* ...] *outfile* `on` [`-`][`@`]*var1* [[`-`][`@`]*var2* ...]

Merges records in order from one or more sorted files. (Maximum number of files = 200). The key word 'on' introduces a sort specification used to determine the order. See 'tabsort' for more details. The output file format (i.e. text/binary) will match the format of the first input file.

`tabmin` *infile outfile* [*var1* ...] [`by` *keyvar1* ...]

Outputs the minimum numeric value for each variable specified in the argument list. All input file variables are processed if none are specified.

`tabrep` *infile outfile repeat_count*

Create *repeat_count* instances of each input record in the output file. *repeat_count* may be a constant or a variable name. In the latter case, the number of instances of each record to create is taken from the value of the specified variable in the input file, rounded to the nearest integer (negative values are treated as 0). The maximum *repeat_count* value allowed is *NREC_MAX*.

`tabsel` [`=`]*infile outfile* [[`@`]*var1*[*comptype*[*value1*]] ...]

Copies records which match all specified conditions to *outfile*. For *comptype* in the condition specification substitute one of `=`, `!=`, `<`, `>`, `<=` or `>=`. The left hand side of the condition must be a variable name. The right hand side may be a variable name or a constant. Numbers are always treated as constants. Non-numeric values are treated as variable names unless they are quoted in double quotes (e.g. `city="London"`). The whole argument must also be enclosed in single quotes to prevent the shell removing the double quotes!

Comparisons are performed as in 'tabsort', with the added functionality that

string comparisons are allowed for binary data. String comparisons between numbers are forced either by preceding variable names with the '`@`' character or by quoting numeric constants. Binary data are converted for the purpose, using *SIGDIG_DEF* significant digits.

If the right hand side of an expression is omitted, it is taken to be the value *NODATA* (or *NODATA_STR* if '`@`' is used). Note that if binary files are converted to text before processing then *NODATA* values are replaced by *NODATA_STR* and the expression *var*`=` will not select any data. To avoid this, when working with numeric data and the file format is unknown, use `@`*var*`=` to get all missing data.

If the condition is omitted as well (i.e. only the variable name is given), then all non-missing numeric data are selected (or all non-missing data if '`@`' is used).

An '`=`' sign preceding the input file name makes the program copy all of the input records and set a new flag variable **sel** to 1 if all conditions are true or 0 otherwise.

`tabseq` *outfile var1*`,`*n1*[`,`*start1*[`,`*step1*]] [*var2... ...*]
or `tabseq` *outfile var1*`=`*start1*`,`*stop1*[`,`*step1*]] [*var2... ...*]

Generates a data series (or grid) of one or more dimensions in *outfile*. One variable is specified for each dimension required. If a single variable is specified, the first form creates a series of *n* data points with the first value specified by *start* (default=1) and the interval by *step* (default=1). The second form creates a series of data points starting at *start* and finishing at, or as close as possible to, *stop* with an interval *step* (default=1 or -1 if *stop* < *start*). The series specified for each additional variable is nested within that for the preceding variable, values from the earlier series being duplicated to accommodate the new data.

`tabsort` *infile outfile* [`-`][`@`]*var1* [[`-`][`@`]*var2 ...*] [`by` *keyvar1 ...*]

Sorts records on the specified variables. The order of significance of the variables is the order in the argument list. A '`-`' sign in front of a variable name indicates that reverse order is to be used for that variable. An '`@`' character in front of a variable name forces string comparisons to be used but, unlike in 'tabsel', it is valid for text input files only. Otherwise numerical values are compared numerically, the *NODATA* value precedes all valid numbers and non-numeric strings precede all numbers. In all cases *NODATA_STR* precedes everything else.

The maximum number of records which can be sorted within a group is *NREC_MAX*. If larger groups need to be sorted then they must be divided into sub-groups (use 'tabenum' and 'tabcalc' to calculate sub-group numbers), sorted by subgroup, split up into separate files using 'tabsplit' and re-combined using 'tabmerge'. Up to 200 sub-groups can be processed in this way.

`tabsplit` *infile outfile_root keyvar1* [*keyvar2 ...*]

Splits the input file into multiple output files according to the values of the key variable(s). One file is created for each unique set of keys so if multiple groups have the same key(s) then they appear in the same output file. (Note: to prevent this 'tabenum' can be used to number the groups first. The group number can then be used as a unique key). The output file names are constructed from *outfile_root* by appending the key variable values. The root and each key variable are separated by a '`.`' character in the file name and this character is not allowed in the data values for key variables. The number of output files is limited to a maximum of 200.

`tabsqz` *infile* [*outfile* [`n=`*n_per_page*] [`s=`*col_spacing*] [`w=`*max_col_width*] [`j=`*justification*] [`d=`*sig_digits*]]

Reformats column widths to fit the longest string in each column, processing a page at a time. Binary files are converted to text on input with the number of significant digits given by *SIGDIG_DEF*. By default:
- number of records per page is *NREC_MAX*.
- column spacing is *NSPACE_DEF*.
- maximum column width is *LINE_MAX*.
- non-numeric strings are left justified and numeric strings are right justified.
- number of significant digits is the same as in the input.

Arguments can be specified as indicated to over-ride defaults. If the page length is specified explicitly variable names are written at the top of each page. Specifying `s=-` gives output separated by vertical lines rather than spaces. Strings exceeding the maximum column width are not truncated. Justification is given as `l` or `r` for left and right respectively. When the number of significant digits is specified, numbers are not converted if the conversion increases the field width.

`tabview` [`=`]*infile* [*outfile* [`n=`*n_per_page*] [`s=`*col_spacing*] [`w=`*max_col_width*] [`j=`*justification*] [`d=`*sig_digits*]]

Output data in page layout form, suitable for viewing but not necessarily readable by other Table Toolbox programs. By default, 20 records are written per page. Variable names are written at the top of each page. Rows longer than 80 characters are split across multiple pages. Records are numbered by default (on all pages). Numbering is suppressed if *infile* is preceded by an '`=`' character.

`tabvname` *infile outfile* [*oldname1*`,`*newname1 ...*]

Re-names variables as specified. Each specified name change is completed before interpreting the next. Where there are multiple occurrences of a variable name in the input file they can be re-named in order of appearance. Only the file header is updated. Data lines are not re-formatted.

B.3 Built-in Values

The following global values are defined in the Table Toolbox include file `toolbox.h`. Current values are given in brackets.

LINE_MAX (20000) – maximum line length for text data records.
NFIELD_MAX (1000) – maximum number of fields in one data record.
NREC_MAX (100000) – maximum number of data records which can be loaded into memory at one time.
SIGDIG_DEF (8) – default number of significant digits for converting numeric values to strings.
NSPACE_DEF (1) – default number of spaces between columns in text output.
NODATA (-9e99) – numerical missing data value.
NODATA_STR ('`_`') – missing data in text files.

Appendix C

Model Descriptions

Each of the following descriptions lists the tracers updated by the model and gives the rate equations for each tracer in terms of sources and sinks. Sinking particle fluxes are included as their parameterizations are model specific. The remaining transport fluxes (advection, diffusion and mixing) are omitted for clarity. The generic notation used is defined in Table C.1. Model-specific notation is described in each section.

Table C.1: Generic Notation for Model Descriptions

Symbol	Description	MarMOT name
μ_{P}	realized level mean photosynthetic rate (d^{-1})	**phot**
$\bar{J}$	potential level mean photosynthetic rate (d^{-1})	**pk**
$P_{\max}$	light-saturated photosynthetic rate for P-E curve (d^{-1})	**pmax**
α_{d}	initial slope for P-E curve ($(\mathrm{E\ m}^{-2})^{-1}$)	**alphad**
V_{P}	maximum photosynthetic rate (d^{-1})	**photmax**
Q_{N}	DIN growth limitation factor	**nlimfact**
K_{N}	half-saturation concentration for DIN uptake (mmol N m^{-3})	**kdin**
G_{X}	loss rate of X due to grazing (mmol N m^{-3} d^{-1})	
M_{X}	loss rate of X due to mortality (mmol N m^{-3} d^{-1})	
m_{X}	linear mortality/breakdown rate for X	<x>**mort**, **remin**
$m^{\star}_{\mathrm{X}}$	density dependent mortality rate for X (d^{-1} (mmol N m^{-3})$^{-1}$)	<x>**mortdd**
g	maximum grazing rate (d^{-1})	**gmax**
n	Holling function exponent for grazing function	
ϵ	prey capture rate (d^{-1} (mmol N m^{-3})$^{-n}$)	
β_{X}	zooplankton assimilation efficiency for food type X	**beta**<x>
s_{D}	detrital sinking rate (m d^{-1})	**dsink**
$\dot{D}_{\mathrm{add}}$	replacement of settled detritus (re-suspension) (mmol N m^{-3} d^{-1})	
h_{D}	height of detritus re-suspension (m)	
Δz	level height (m)	**dz**
z	depth at level mid-point (m)	**z**
z_1	depth at top of level (m)	**ztop**
z_2	depth at bottom of level (m)	**zbot**
z_{water}	maximum water depth	
k	level number	**k**
k_{phot}	deepest level at which photosynthesis occurs	
k_{bot}	deepest level in water column	

Table C.2: Model-specific notation for nitrogen cycle in OG99NPZD

Symbol	Description	MarMOT name
a	max. photosynthetic rate at 0 °C (d^{-1})	**aphotmax**
b	max. photosynthesis - base for temperature variation factor	**bphotmax**
c	max. photosynthesis - temperature sensitivity of exponent ($(°C)^{-1}$)	**cphotmax**
T	temperature (°C)	**temp**
η_Z	zooplankton excretion rate (d^{-1})	**zexcr**

C.1 OG99NPZD

C.1.1 Tracers

N Dissolved inorganic nitrogen (**din**)
P Phytoplankton nitrogen (**phy**)
Z Zooplankton nitrogen (**zoo**)
D Detrital nitrogen (**det**)

C.1.2 Nitrogen Cycle

For model-specific notation see Table C.2.

$$\frac{dP}{dt} = \mu_P P - G_P - M_P \tag{C.1}$$
$$\frac{dZ}{dt} = \beta_P G_P - \eta_Z Z - M_Z \tag{C.2}$$
$$\frac{dD}{dt} = (1-\beta_P)G_P + M_P + M_Z - m_D D + s_D \frac{D_{k-1} - D}{\Delta z} + \dot{D}_{add} \tag{C.3}$$
$$\frac{dN}{dt} = m_D D + \eta_Z Z - \mu_P P \tag{C.4}$$

$$\mu_P = \min(\bar{J}(P_{max}, \alpha_d), V_P Q_N) \tag{C.5}$$
$$P_{max} = V_P \tag{C.6}$$
$$V_P = a b^{cT} \tag{C.7}$$

$$Q_{\mathrm{N}} = \frac{N}{K_{\mathrm{N}} + N} \tag{C.8}$$

$$G_{\mathrm{P}} = \frac{g\epsilon P^2}{g + \epsilon P^2} Z \tag{C.9}$$

$$M_{\mathrm{P}} = m_{\mathrm{P}} P \tag{C.10}$$

$$M_{\mathrm{Z}} = m_{\mathrm{z}}^{\star} Z^2 \tag{C.11}$$

$$\dot{D}_{\mathrm{add}} = \left\{ \begin{array}{ccc} 0 & , & z < z_{\mathrm{water}} - h_{\mathrm{D}} \\ \frac{s_{\mathrm{D}} D_{\mathrm{k_{bot}}}}{h_{\mathrm{D}}} & , & z > z_{\mathrm{water}} - h_{\mathrm{D}} \end{array} \right\} \tag{C.12}$$

$$h_{\mathrm{D}} = (\Delta z)_{\mathrm{k_{bot}}} \tag{C.13}$$

C.2 HadOCC

C.2.1 Tracers

N	Dissolved inorganic nitrogen (**din**)
P	Phytoplankton nitrogen (**phy**)
Z	Zooplankton nitrogen (**zoo**)
D	Detrital nitrogen (**det**)
N_{r}	Ammonium (**nh4**)
C	Dissolved inorganic carbon (**dic**)
A	Alkalinity (**alk**)
Chl	Chlorophyll (**chl**)

C.2.2 Nitrogen Cycle

For model-specific notation see Table C.3.

$$\frac{dP}{dt} = \mu_{\mathrm{P}} P - G_{\mathrm{P}} - M_{\mathrm{P}} - \eta_{\mathrm{P}} P \tag{C.14}$$

$$\frac{dZ}{dt} = \phi_{\mathrm{aZ}} a_{\mathrm{N}} - M_{\mathrm{Z}} \tag{C.15}$$

$$\frac{dD}{dt} = \phi_{\mathrm{dD}} d_{\mathrm{N}} - G_{\mathrm{D}} - m_{\mathrm{D}} D + s_{\mathrm{D}} \frac{D_{\mathrm{k-1}} - D}{\Delta z} + \dot{D}_{\mathrm{add}} \tag{C.16}$$

$$\frac{dN}{dt} = \dot{N}_{\mathrm{regen}} - \mu_{\mathrm{P}} P \tag{C.17}$$

Table C.3: Model-specific notation for nitrogen cycle in HadOCC

Symbol	Description	MarMOT name
$P_{\text{threshold}}$	threshold for phytoplankton mortality (mmol N m^{-3})	**pminmort**
η_{P}	phytoplankton respiration rate (d^{-1})	**presp**
F	biomass† concentration of food available for grazing (mmol N m^{-3})	
F_{tot}	biomass† concentration of food (mmol N m^{-3})	
$F_{\text{threshold}}$	food threshold for grazing function (mmol N m^{-3})	**pmingraz**
K_{F}	half-saturation food concentration for grazing function (mmol N m^{-3})	
B_{X}	ratio of biomass† to nitrogen for X	
θ_{X}	C:N ratio of X (mol C (mol N)$^{-1}$)	**rcn**<*xname*>
θ_{R}	C:N ratio of standard biotic material = 106:16 (mol C (mol N)$^{-1}$)	
ϕ_{aZ}	fraction of potentially assimilated nitrogen retained	
θ_{a}	C:N ratio of potentially assimilated food (mol C (mol N)$^{-1}$)	
a_{N}	potential nitrogen assimilation rate (mmol N m^{-3} d^{-1})	
a_{C}	potential carbon assimilation rate (mmol C m^{-3} d^{-1})	
ϕ_{dD}	fraction of newly formed detrital nitrogen retained (mol C (mol N)$^{-1}$)	
θ_{d}	C:N ratio of newly formed detritus (mol C (mol N)$^{-1}$)	
d_{N}	formation rate of detrital nitrogen (mmol N m^{-3} d^{-1})	
d_{C}	formation rate of detrital carbon (mmol C m^{-3} d^{-1})	
ϕ_{MXN}	fraction of X mortality going directly to DIN	**f**<*x*>**mortdin**
ϕ_{GXD}	fraction of X loss due to grazing going to detritus	
ϕ_{I}	fraction of grazed material ingested	
ϕ_{mfN}	fraction of messy feeding going to detritus	**fmessyd**
r_{max}	maximum detrital remineralization rate = 0.1 (d^{-1})	
z_{rmax}	depth down to which maximum remineralization rate applies = 80.58 (m)	
$\dot{N}_{\text{regen}}$	rate of DIN regeneration (d^{-1})	
ν	nitrification rate of ammonium (d^{-1})	
ν_{euph}	nitrification rate of ammonium in euphotic zone (d^{-1})	**nitrifeuph**
ν_{aph}	nitrification rate of ammonium aphotic zone (d^{-1})	**nitrifaph**
I_{sol}	solar radiation incident on sea-surface (W m^{-2})	**sol** or **solav**
$I_{\text{threshold}}$	threshold surface solar radiation for euphotic nitrification = 50 (W m^{-2})	
z_{mld}	mixed layer depth (m)	**mld**
z_{euph}	euphotic zone depth (m)	**zeuphotic**

† Biomass is expressed in biomass-equivalent nitrogen units. As a biomass-equivalent quantity, 1 mmol N is defined as the biomass of a quantity of material with a standard C:N ratio (Redfield *et al.*, 1963) that contains 1 mmol N.

$$\mu_{\mathrm{P}} = \bar{J}(P_{\max}, \alpha_{\mathrm{d}}) \tag{C.18}$$

$$P_{\max} = V_{\mathrm{P}} Q_{\mathrm{N}} \tag{C.19}$$

$$Q_{\mathrm{N}} = \frac{N}{K_{\mathrm{N}} + N} \tag{C.20}$$

$$M_{\mathrm{P}} = mP^2 \tag{C.21}$$

$$m = \left\{ \begin{array}{ccc} 0 & , & P \leq P_{\mathrm{threshold}} \\ m^{\star}_{\mathrm{P}} & , & P > P_{\mathrm{threshold}} \end{array} \right\} \tag{C.22}$$

$$G_{\mathrm{F}} = \frac{gF^n}{K_{\mathrm{F}}^n + F^n} B_{\mathrm{Z}} Z \tag{C.23}$$

$$K_{\mathrm{F}}^n = \frac{g}{\epsilon} \tag{C.24}$$

$$F = \max\left(0, F_{\mathrm{tot}} - F_{\mathrm{threshold}}\right) \tag{C.25}$$

$$F_{\mathrm{tot}} = B_{\mathrm{P}} P + B_{\mathrm{D}} D \tag{C.26}$$

$$B_{\mathrm{X=P,Z,D}} = \frac{14.01 + 12.01\theta_{\mathrm{X}}}{14.01 + 12.01\theta_{\mathrm{R}}} \tag{C.27}$$

$$G_{\mathrm{X=P,D}} = \frac{G_{\mathrm{F}}}{F_{\mathrm{tot}}} X \tag{C.28}$$

$$\phi_{\mathrm{aZ}} = \min\left(\frac{\theta_{\mathrm{a}}}{\theta_{\mathrm{Z}}}, 1\right) \tag{C.29}$$

$$\theta_{\mathrm{a}} = \frac{a_{\mathrm{C}}}{a_{\mathrm{N}}} \tag{C.30}$$

$$a_{\mathrm{N}} = \phi_{\mathrm{I}}(\beta_{\mathrm{P}} G_{\mathrm{P}} + \beta_{\mathrm{D}} G_{\mathrm{D}}) \tag{C.31}$$

$$a_{\mathrm{C}} = \phi_{\mathrm{I}}(\theta_{\mathrm{P}}\beta_{\mathrm{P}} G_{\mathrm{P}} + \theta_{\mathrm{D}}\beta_{\mathrm{D}} G_{\mathrm{D}}) \tag{C.32}$$

$$M_{\mathrm{Z}} = m_{\mathrm{Z}} Z + m^{\star}_{\mathrm{Z}} Z^2 \tag{C.33}$$

$$\phi_{\mathrm{dD}} = \min\left(\frac{\theta_{\mathrm{d}}}{\theta_{\mathrm{D}}}, 1\right) \tag{C.34}$$

$$\theta_{\mathrm{d}} = \frac{d_{\mathrm{C}}}{d_{\mathrm{N}}} \tag{C.35}$$

$$\begin{aligned} d_{\mathrm{N}} = {} & (1 - \phi_{\mathrm{MPN}}) M_{\mathrm{P}} + (1 - \phi_{\mathrm{MZN}}) M_{\mathrm{Z}} \\ & + \phi_{\mathrm{GPD}} G_{\mathrm{P}} + \phi_{\mathrm{GDD}} G_{\mathrm{D}} \end{aligned} \tag{C.36}$$

$$\begin{aligned} d_{\mathrm{C}} = {} & \theta_{\mathrm{P}}(1 - \phi_{\mathrm{MPN}}) M_{\mathrm{P}} + \theta_{\mathrm{Z}}(1 - \phi_{\mathrm{MZN}}) M_{\mathrm{Z}} \\ & + \theta_{\mathrm{P}}\phi_{\mathrm{GPD}} G_{\mathrm{P}} + \theta_{\mathrm{D}}\phi_{\mathrm{GDD}} G_{\mathrm{D}} \end{aligned} \tag{C.37}$$

$$\phi_{\mathrm{GPD}} = (1 - \phi_{\mathrm{mfN}})(1 - \phi_{\mathrm{I}}) + (1 - \beta_{\mathrm{P}})\phi_{\mathrm{I}} \tag{C.38}$$

$$\phi_{\mathrm{GDD}} = (1 - \phi_{\mathrm{mfN}})(1 - \phi_{\mathrm{I}}) + (1 - \beta_{\mathrm{D}})\phi_{\mathrm{I}} \tag{C.39}$$

$$m_{\mathrm{D}} = r_{\max} \min\left(\frac{z_{\mathrm{rmax}}}{z}, 1\right) \tag{C.40}$$

$$\dot{D}_{\mathrm{add}} = \left\{ \begin{array}{ccc} 0 & , & z < z_{\mathrm{water}} - h_{\mathrm{D}} \\ \frac{s_{\mathrm{D}} D_{\mathrm{k_{bot}}}}{h_{\mathrm{D}}} & , & z > z_{\mathrm{water}} - h_{\mathrm{D}} \end{array} \right\} \tag{C.41}$$

$$h_{\mathrm{D}} = \sum_{k=k_{\mathrm{bot}}-2}^{k_{\mathrm{bot}}} (\Delta z)_{\mathrm{k}} \tag{C.42}$$

$$\begin{aligned} \dot{N}_{\mathrm{regen}} = {} & \phi_{\mathrm{MPN}} M_{\mathrm{P}} + \eta_{\mathrm{P}} P + \phi_{\mathrm{MZN}} M_{\mathrm{Z}} \\ & + \phi_{\mathrm{mfN}}(1-\phi_{\mathrm{I}})(G_{\mathrm{P}} + G_{\mathrm{D}}) + (1-\phi_{\mathrm{aZ}}) a_{\mathrm{N}} + (1-\phi_{\mathrm{dD}}) d_{\mathrm{N}} \\ & + m_{\mathrm{D}} D \end{aligned} \tag{C.43}$$

If sinking of detritus is handled externally (model option parameter **dsinkopt** = 0) then

$$h_{\mathrm{D}} = (\Delta z)_{\mathrm{k_{bot}}} \tag{C.44}$$

Optional ammonium tracer, active if **nh4tracer** = 1, diagnoses ammonium contribution to DIN:

$$\frac{dN_{\mathrm{r}}}{dt} = \dot{N}_{\mathrm{regen}} - \min\left(\frac{N_{\mathrm{r}}}{N}, 1\right) \mu_{\mathrm{P}} P - \nu N_{\mathrm{r}} \tag{C.45}$$

$$\nu = \left\{ \begin{array}{ccccc} \nu_{\mathrm{euph}} & , & z_1 \le \max(z_{\mathrm{mld}}, z_{\mathrm{euph}}) & \mathrm{AND} & I_{\mathrm{sol}} \ge I_{\mathrm{threshold}} \\ \nu_{\mathrm{aph}} & , & z_1 > \max(z_{\mathrm{mld}}, z_{\mathrm{euph}}) & \mathrm{OR} & I_{\mathrm{sol}} < I_{\mathrm{threshold}} \end{array} \right\} \tag{C.46}$$

C.2.3 Carbonate System

This part of the model is active if model option parameter **co2sys** = 1. Additional model-specific notation is given in Table C.4.

$$\begin{aligned} \frac{dC}{dt} = {} & \theta_{\mathrm{P}} \phi_{\mathrm{MPN}} M_{\mathrm{P}} + \theta_{\mathrm{P}} \eta_{\mathrm{P}} P + \theta_{\mathrm{Z}} \phi_{\mathrm{MZN}} M_{\mathrm{Z}} \\ & + \phi_{\mathrm{mfN}}(1-\phi_{\mathrm{I}})(\theta_{\mathrm{P}} G_{\mathrm{P}} + \theta_{\mathrm{D}} G_{\mathrm{D}}) + \phi_{\mathrm{aC}} a_{\mathrm{C}} + \phi_{\mathrm{dD}} d_{\mathrm{C}} \\ & + \theta_{\mathrm{D}} m_{\mathrm{D}} D + \dot{C}_{\mathrm{dis}} - (1+\gamma_{\mathrm{CO3}}) \theta_{\mathrm{P}} \mu_{\mathrm{P}} P + F_{\mathrm{CO2}} \end{aligned} \tag{C.47}$$

$$\frac{dA}{dt} = 2\dot{C}_{\mathrm{dis}} - 2\gamma_{\mathrm{CO3}} \theta_{\mathrm{P}} \mu_{\mathrm{P}} P - \frac{dN}{dt} \tag{C.48}$$

Table C.4: Notation specific to carbonate system sub-model in HadOCC

Symbol	Description	MarMOT name
ϕ_{aC}	fraction of potentially assimilated carbon not retained	
ϕ_{dD}	fraction of newly formed detrital carbon not retained	
γ_{CO3}	carbonate precipitated per unit primary production	**rco3pprod**
$\dot{C}_{\mathrm{dis}}$	carbonate dissolution rate (mmol C m^{-3} d^{-1})	**co3dis**
z_{lyso}	lysocline depth = 2000 (m)	
k_{lyso}	top level with mid-point below lysocline depth	
h_{CO3dis}	height over which carbonate is dissolved	
F_{CO2}	rate of DIC change due to air-sea CO_2 flux (mmol C m^{-3} d^{-1})	
f_{CO2}	air-sea CO_2 flux (mmol m^{-2} d^{-1})	**co2flux**
f_{CO2in}	CO_2 invasion (mmol m^{-2} d^{-1})	**co2in**
f_{CO2out}	CO_2 evasion (mmol m^{-2} d^{-1})	**co2out**
s_{CO2}	solubility of CO_2 (mmol kg^{-1} μatm^{-1})	
ρ_{w}	density of sea water = 1026 (kg m^{-3})	
k_{v}	gas transfer velocity = 4.800384 (m d^{-1})	
p_{CO2atm}	atmospheric pCO_2 (μatm)	**pco2atm**
p_{CO2}	sea surface pCO_2 (μatm)	**pco2**
S	salinity	**sss**

$$\phi_{\mathrm{aC}} = \max\left(0, 1 - \frac{\theta_{\mathrm{Z}}}{\theta_{\mathrm{a}}}\right) \tag{C.49}$$

$$\phi_{\mathrm{dC}} = \max\left(0, 1 - \frac{\theta_{\mathrm{D}}}{\theta_{\mathrm{d}}}\right) \tag{C.50}$$

$$\dot{C}_{\mathrm{dis}} = \left\{ \begin{array}{cc} 0, & z < z_{\mathrm{lyso}} \\ \frac{\gamma_{\mathrm{CO3}}\theta_{\mathrm{P}}}{h_{\mathrm{CO3dis}}} \sum_{k=1}^{k_{\mathrm{phot}}} (\mu_{\mathrm{P}} P\, \Delta z)_{\mathrm{k}}, & z > z_{\mathrm{lyso}} \end{array} \right\} \tag{C.51}$$

$$h_{\mathrm{CO3dis}} = \sum_{k=k_{\mathrm{lyso}}}^{k_{\mathrm{bot}}} (\Delta z)_{\mathrm{k}} \tag{C.52}$$

$$F_{\mathrm{CO2}} = \left\{ \begin{array}{cc} \frac{f_{\mathrm{CO2}}}{\Delta z}, & k = 1 \\ 0, & k > 1 \end{array} \right\} \tag{C.53}$$

$$f_{\mathrm{CO2}} = f_{\mathrm{CO2in}} - f_{\mathrm{CO2out}} \tag{C.54}$$

$$f_{\mathrm{CO2in}} = s_{\mathrm{CO2}}(T,S)\rho_{\mathrm{w}} k_{\mathrm{v}} p_{\mathrm{CO2atm}} \tag{C.55}$$

$$f_{\mathrm{CO2out}} = s_{\mathrm{CO2}}(T,S)\rho_{\mathrm{w}} k_{\mathrm{v}} p_{\mathrm{CO2}}(T,S,C,A) \tag{C.56}$$

The solubility s_{CO2} is calculated according to Weiss (1974). The surface water pCO_2 p_{CO2} is calculated, using the method of Bacastow (1981), as described by Palmer and Totterdell (2001).

C.2.4 Phytoplankton Carbon:Chlorophyll Ratio

Photo-acclimation changes in the C:Chl ratio $\theta_{\rm chl}$ alter the carbon-specific initial slope of the phytoplankton P-E curve according to

$$\alpha_{\rm d} = \frac{1}{\theta_{\rm chl}} \alpha_{\rm chl} \tag{C.57}$$

where $\alpha_{\rm chl}$ is the chlorophyll-specific initial slope for downwelling PAR. If model option parameter **cchlopt** $= 0$, $\theta_{\rm chl}$ is fixed at parameter value **rcchl**. If **cchlopt** $=$ 1, the acclimation model applies:

$$\theta_{\rm chl} = \min\left(\sqrt{\theta_{\rm min}\frac{\alpha_{\rm chl}E_{\rm d}}{J(\theta_{\rm chl})}}, \theta_{\rm max}\right) \tag{C.58}$$

$$J(\theta_{\rm chl}) = P_{\rm max}\left[1 - \exp\left(-\frac{\alpha_{\rm chl}E_{\rm d}}{\theta_{\rm chl}P_{\rm max}}\right)\right] \tag{C.59}$$

where $\theta_{\rm min}$ and $\theta_{\rm max}$ are the minimum and maximum C:Chl ratios (**rcchlmin**, **rcchlmax**), $E_{\rm d}$ is the downwelling PAR and J is the carbon-specific photosynthetic rate.

The model is based on the steady state solution of the Geider *et al.* (1997) photo-acclimation model, describing the light dependency of the C:Chl ratio under balanced growth conditions. Such conditions are rarely achieved in the upper boundary layer because of the interaction between acclimation and vertical mixing. Conceptually, the upper boundary layer defined by the **mld** variable is treated as fully-mixed for all tracers, including chlorophyll. So, for levels wholly within the boundary layer

$$\theta_{\rm chl} = \min\left(\sqrt{\theta_{\rm min}\frac{\alpha_{\rm chl}E_{\rm d}}{J(\langle\theta_{\rm chl}\rangle_{1..{\rm k_{mld}}})}}, \theta_{\rm max}\right) \tag{C.60}$$

where $\langle . \rangle_{1..{\rm k_{mld}}}$ indicates averaging over the levels above depth **mld**.

Implementation notes:

1. R.H.S. in Equations C.58 and C.60 is computed using θ_{chl} from the previous time step (hence **rcchl** is needed in the initial conditions). It can be modified by physical transport before use if **chltracer** option is selected.

2. Level mean $\theta_{\text{chl}}(E_{\text{d}}(z))$ is approximated by $\theta_{\text{chl}}\left(\frac{1}{2}(E_{\text{d}}(z_1) + E_{\text{d}}(z_2))\right)$, where z_1 and z_2 are the upper and lower depths for the level.

3. E_{d} is determined from I_{sol} which is **sol** or **solav** depending on the photosyntheis model. No parametrization of the diel cycle is implemented: both variables are treated as point-in-time irradiances. There is therefore potential for low bias in θ_{chl} due to under-saturation of light-limited photosynthesis when using **solav**.

Bibliography

Acton, F. S. 1970. Numerical Methods That Work: 1990, corrected edition, Washington: Mathematical Association of America, pp. 464-467.

Anderson, T. R. 1993. A spectrally averaged model of light penetration and photosynthesis. Limnol. Oceanogr. *38*, 1403-1419.

Anderson, T. R. 2005. Plankton functional type modelling: running before we can walk? J. Plankton Res. *27*, 1073-1081.

Asselin, R. 1972. Frequency filter for time integrations, Monthly Weather Rev. *105*, 487-490.G

Bacastow, R. B. 1981. Numerical evaluation of the evasion factor. In: Bolin, B. (Ed.), Carbon Cycle Modelling. Wiley, New York, pp. 95-101.

Brent, R. P. 1973. Algorithms for Minimization without Derivatives. Englewood Cliffs, NJ: Prentice-Hall, Chapter 5.

Dadou, I., G. Evans and V. Garçon. 2004. Using JGOFS *in situ* and ocean color data to compare biogeochemical models and estimate their parameters in the subtropical North Atlantic Ocean. J. Mar. Res. *62*, 565-594.

Evans, G. T. and J. S. Parslow. 1985. A model of annual plankton cycles. Biol. Oceanogr. *3*, 327-347.

Evans, G. T., 1999. The role of local models and data sets in the Joint Global Ocean Flux Study. Deep-Sea Res. I *46*, 1369-1389.

Fasham, M. J. R., H. W. Ducklow and S. M. McKelvie. 1990. A nitrogen-based model of plankton dynamics in the oceanic mixed layer. J. Mar. Res. *48*, 591-639.

Fasham, M. J. R. and G. T. Evans. 1995. The use of optimization techniques to model marine ecosystem dynamics at the JGOFS station at 47°N 20°W. Phil. Trans. Roy. Soc. Lond. B *348*, 203-209.

Friedrichs, M. A. M., R. R. Hood and J. D. Wiggert. 2006. Ecosystem model complexity versus physical forcing: Quantification of their relative impact with assimilated Arabian Sea data. Deep-Sea Res. II *53*, 576-600.

Friedrichs, M. A. M., J. A. Dusenberry, L. A. Anderson, R. A. Armstrong, F. Chai, J. R. Christian, S. C. Doney, J. Dunne, M. Fujii, R. Hood, D. J. McGillicuddy Jr., K. Moore, M. Schartau, Y. Spitz and J. D. Wiggert. 2007. Assessment of skill and portability in regional marine biogeochemical models: Role of multiple planktonic groups. J. Geophys. Res. *112*, C08001, doi:10.1029/2006JC003852.

Geider, R. J., H. L. MacIntyre and T. M. Kana. 1997. Dynamic model of phyto-

plankton growth and acclimation: Responses of the balanced growth rate and the chlorophyll *a*:carbon ratio to light, nutrient-limitation and temperature. Mar. Ecol. Prog. Ser. *148*, 187-200.

Hemmings, J. C. P., M. A. Srokosz, P. Challenor and M. J. R. Fasham. 2003. Assimilating satellite ocean-colour observations into oceanic ecosystem models. Philos. T. Roy. Soc. A *361*, 33-39.

Hemmings, J. C. P., M. A. Srokosz, P. Challenor and M. J. R. Fasham. 2004. Split-domain calibration of an ecosystem model using satellite ocean colour data. J. Marine Syst. *50*, 141-179.

Hurtt, G. C. and R. A. Armstrong. 1999. A pelagic ecosystem model calibrated with BATS and OWSI data. Deep-Sea Res. I *46*, 27-61.

Kettle, H and C. J. Merchant. 2008. Modeling ocean primary production: Sensitivity to spectral resolution of attenuation and absorption of light. Progr. Oceanogr. *78*, 135-146.

Krishnakumar, K. 1989. Micro-genetic algorithms for stationary and non-stationary function optimization, Proc. SPIE: Intelligent Control and Adaptive Systems *1196*, Philadelphia, PA, 289-296.

Lafore, J. P., J. Stein, N. Asencio, P. Bougeault, V. Ducrocq, J. Duron, C. Fischer, P. Hreil, P. Mascart, V. Masson, J. P. Pinty, J. L. Redelsperger, E. Richard and J. Vil-Guerau de Arellano. 1998. The Meso-NH Atmospheric Simulation System. Part I: adiabatic formulation and control simulations. Ann. Geophys. *16*, 90-109.

Le Quéré, C. 2006. Reply to horizons article 'Plankton functional type modelling: running before we can walk' Anderson (2005): I. Abrupt changes in marine ecosystems? J. Plankton Res. *28*, 871-872.

Losa, S. N., G. A. Kivman and V. A. Ryabchenko. 2004. Weak constraint parameter estimation for a simple ocean ecosystem model: what can we learn about the model and data? J. Marine Syst. *45*, 1-20.

Losa, S. N., A. Vézina, D. Wright, Y. Y. Lu, K. Thompson and M. Dowd. 2006. 3D ecosystem modelling in the North Atlantic: Relative impacts of physical and biological parameterizations. J. Marine Syst. *61*, 230-245.

Matear, R. J. 1995. Parameter optimization and analysis of ecosystem models using simulated annealing: A case study at Station P. J. Mar. Res. *53*, 571-607.

Morel, A. and R. C. Smith. 1974. Relation between total quanta and total energy for aquatic photosynthesis. Limnol. Oceanogr. *19*, 591-600.

Morel, A. 1988. Optical modelling of the upper ocean in relation to its biogenous matter content (case 1 waters). J. Geophys. Res. *93*,10749-10768.

Morel, A. 1991. Light and marine photosynthesis: A spectral model with geochemical and climatological implications. Prog. Oceanogr. *26*, 263-306.

Oschlies, A. and V. Garçon. 1999. An eddy-permitting coupled physical-biological model of the North Atlantic. 1. Sensitivity to advection numerics and mixed layer physics. Global Biogeochem. Cy. *13*, 135-160.

Palmer, J. R. and I. J. Totterdell. 2001. Production and export in a global ocean ecosystem model. Deep-Sea Res. I *48*, 1169-1198.

Platt, T., S. Sathyendranath and P. Ravindran. 1990. Primary production by phytoplankton: Analytic solutions for daily rates per unit area of water surface. Proc. R. Soc. Lond. Ser. B *241*, 101-111.

Powell, M. J. D. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. Computer J. *7*, 155-162.

Press, W. H., B. P. Flannery, S. A. Teukolsky and W. T. Vetterling. 1992. Numerical Recipes in C: the Art of Scientific Computing. Cambridge Univ. Press, Cambridge.

Redfield, A. C., B. H. Ketchum and F. A. Richards. 1963. The influence of organisms on the composition of sea-water. In: Hill, M. N. (Ed.), The Sea. Vol 2. Wiley-Interscience, New York, pp. 26-77.

Robert, A. J., 1966. The integration of a low order spectral form of the primitive meteorological equations, J. Meteorol. Soc. Jpn. *44*, 237-245.

Schartau, M. and A. Oschlies. 2003a. Simultaneous data-based optimization of a 1D-ecosystem model at three locations in the North Atlantic: Part I - Method and parameter estimates. J. Mar. Res. *61*, 765-793.

Schartau, M. and A. Oschlies. 2003b. Simultaneous data-based optimization of a 1D-ecosystem model at three locations in the North Atlantic: Part II - Standing stocks and nitrogen fluxes. J. Mar. Res. *61*, 795-821.

Tjiputra, J. F., D. Polzin and A. M. E. Winguth. 2007. Assimilation of seasonal chlorophyll and nutrient data into an adjoint three-dimensional ocean carbon cycle model: sensitivity analysis and ecosystem parameter optimization. Global Biogeochem. Cy. *21*, GB1001, doi:10.1029/2006GB002745.

Wanninkhof, R. 1992. Relationship between wind speed and gas exchange over the ocean. J. Geophys. Res. *97*, 7373-7382.

Weiss, R. F. 1974. Carbon dioxide in water and seawater: the solubility of a non-ideal gas. Mar. Chem. *2*, 203-215.

Zheng, X., T. Dickey and G. Chang, 2002. Variability of the downwelling diffuse attenuation coefficient with consideration of inelastic scattering. Applied Optics *41*, 6477-6488.