Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

Geoscientific Model Development Discussions

Automated model optimisation using the Cylc workflow engine (Cyclops v1.0)

Richard M. Gorman[1], Hilary J. Oliver[2]

[1] National Institute of Water and Atmospheric Research, PO Box 11-115, Hamilton, New Zealand
[2] National Institute of Water and Atmospheric Research, Private Bag 14901, Wellington, New Zealand

Correspondence to: Richard M. Gorman (Richard.Gorman@niwa.co.nz)

Abstract. Most geophysical models include a number of parameters that are not fully determined by theory, and can be 'tuned' to improve the model's agreement with available data. We might attempt to automate this tuning process in an objective way by employing an optimisation algorithm to find the set of parameters that minimises a cost function derived from comparing model outputs with measurements. A number of algorithms are available for solving optimisation problems, in various programming languages, but interfacing such software to a complex geophysical model simulation, presents certain challenges. To tackle this problem, we have developed an optimisation suite ("Cyclops") based on the Cylc workflow engine (http://cylc.github.io/cylc/ and https://zenodo.org/badge/latestdoi/1836229) that implements a wide selection of optimisation algorithms from the NLopt python toolbox (Johnson, 2014). The Cyclops optimisation suite can be used to calibrate any modelling system that has itself been implemented as a (separate) Cylc model suite, provided it includes computation and output of the desired scalar cost function. A growing number of institutions are using Cylc to orchestrate complex distributed suites of interdependent cycling tasks within their operational forecast systems, and in such cases application of the optimisation suite is particularly straightforward.

As a test case, we applied the Cyclops to calibrate a global implementation of the Wavewatch III™ (v4.18) third generation spectral wave model, forced by ERA-Interim input fields. This was calibrated over a one-year period (1997), before applying the calibrated model to a full (1979-2016) wave hindcast. The chosen error metric was the spatial average of the root-mean-square error of hindcast significant wave height compared with collocated altimeter records. We describe the results of a calibration in which up to 19 parameters were optimised.

1 Introduction

Geophysical models generally include some empirical parameterisations that are not fully determined by physical theory, and which need calibration. The calibration process has often been somewhat subjective and poorly documented (Voosen, 2016), but in a more objective approach has the aim of minimising some measure of error quantified from comparisons with measurement (Hourdin et al., 2017). We can turn this into an optimisation problem: namely to find the minimum of an objective function $f(\vec{x})$ where $\vec{x}$ represents the set of adjustable parameters, and f is a single error metric (e.g. the sum of RMS differences between measured and predicted values of a set of output variables) resulting from a model simulation with that parameter set. The most efficient optimisation algorithms require the derivative $\vec{\nabla} f(\vec{x})$ to be available alongside $f(\vec{x})$. This, however, is rarely the case for a geophysical modelling system, so will restrict our attention to the field of Differential Free Optimisation (DFO), in which the objective function f can be calculated, but its gradient is not available.

Various methods exist, many of which are summarised in the review of Rios and Sahinidis (2012). Some are good at exploring parameter space to improve the likelihood of finding global rather than merely local minima. Others are preferred for quickly moving to the absolute minimum once in its neighbourhood. The algorithms are encoded in various languages (e.g. Fortran, C, Python, Matlab), and usually require the user to supply a subroutine to compute $f(\vec{x})$, that can be called as required by the optimisation programme.

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

This is satisfactory for many problems where the objective function is readily expressed as an algorithm, but is somewhat less straightforward to interface an existing geophysical model, as well as all the methods needed to process and compare measurement data with an optimisation code, in this way. Nevertheless, examples of this approach can be found in hydrological and climate modelling applications. For example, Seong et al. (2015) developed a calibration tool (using R software) to apply the Shuffled Complex Evolution optimization algorithm to calibrate the Hydrologic Simulation Program-Fortran (HSPF) model. In climate modelling, Severijns and Hazeleger (2005) used the downhill simplex method to optimize the parameter values of the subgrid parameterizations of an atmospheric general circulation model. More recently, Tett et al. (2013) applied a Gauss–Newton line search optimization algorithm to climate simulations with the Hadley Centre Atmosphere Model version 3 (HadAM3) forced with observed sea surface temperature and sea ice, optimising an objective function derived from reflected shortwave radiation and outgoing longwave radiation comparisons. The Tett et al. (2013) method was subsequently applied to optimize the sea ice component of the global coupled HadCM3 climate model (Roach et al., 2017).

Such custom applications of one particular optimisation algorithm to a specific model, however, can require significant effort to switch to alternative optimisation algorithms, or to be applied to new models.

Modern coupled climate models, or operational forecast systems for weather and related processes, encompass a diverse set of software tools, often running on multiple platforms. Ideally, we would like to be able to optimise performance of the modelling *system* (not just a single model code) without major reconfiguration of software between the calibration and operational/production versions of the system.

The Cylc workflow engine is now applied in several operational centres to manage the scheduling of tasks within such systems. So it seems natural to consider the possibility of developing a framework within Cylc for the optimisation of the modelling systems under its control.

2 Methods

In very general terms, a derivative-free optimisation algorithm will explore parameter space, selecting values of the parameter vector $\vec{x}$ in some sequence. As each $\vec{x}$ is selected, it calls the (user-supplied) subroutine to evaluate the objective function $f(\vec{x})$. In our case, this would amount to running a complete model simulation with the corresponding parameter settings, comparing outputs to measurements, from which a defined error metric is computed to provide the return value of f. This can involve a lengthy simulation, needing a run time T_{model} perhaps of order hours or days to reproduce months or years of measurements.

A self-contained optimisation program, with an explicitly-coded function-evaluation subroutine, will run much faster, with a run time per iteration T_{iter} typically being some small fraction of a second, and will run in many orders of magnitude less time than a typical geophysical model even if a number of iterations N of order 1000 are required. This might be the case for "deliberately difficult" test problems: we might expect that a well-tested geophysical model will come with reasonable defaults that in many new implementations will produce a result within a relatively simple "basin of attraction" so that O(10) iterations may suffice to get very close.

If the optimisation procedure calls for a full model run to evaluate the objective function, and N iterations are required for convergence, the total run time would be

$$T \approx T_0 + N(T_{model} + T_{iter}) \quad (1)$$

including an overhead T_o for initial and final tasks.

As T_{model} is orders of magnitude larger than T_o and T_{iter}, the geophysical modelling system totally dominates run time, and we can comfortably afford not to be concerned with reducing the efficiency of the optimisation routine, even by a few orders of magnitude.

So let's consider a simple measure we might introduce to allow us to recover from an interruption part way through a long optimisation process. Normally, the optimisation code will retain in memory the values of each $\vec{x}$ and its objective function

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

$f(\vec{x})$ that has already been evaluated, to use in selecting further points to be evaluated. If we write these values to file each time the function evaluation is called, we can build up a lookup table to use in case we need to restart the process. In that case, we could have the function evaluation subroutine first search the lookup table for a match to $\vec{x}$ (within some acceptable tolerance), in which case it could return the tabulated error value. Only in the case where a tabulated value was not found would the full model simulation be required to compute the return value of *f*.

Now rather than actually perform that computation, the function evaluation subroutine could simply write the $\vec{x}$ values (for the n^{th} iteration, say) to file, and terminate the optimisation. We could then run the model in its usual way, outside the optimisation code, using those $\vec{x}$ values as parameters, and add that result to our lookup table before restarting the whole process from scratch. This time, assuming the optimisation algorithm is deterministic, with no random process influencing the sequence of $\vec{x}$ values, the first *n* points would be exactly the same sequence that was selected previously, and could be quickly handled by table lookup, and the algorithm would either find that a convergence criterion had been satisfied, or select a new point *n*+1 to be passed to the model for simulation.

In this scheme, and assuming that we start with an empty lookup table, the first pass has one iteration of the optimisation code, the second has two, etc. So, allowing an additional overhead $\hat{T}$ for the full process, the total run time to reach the termination condition(s) after *N* iterations should be

$$T' = \hat{T} + \sum_{n=1}^{N}(T_o + nT_{iter} + T_{model}) \quad (2)$$

$$= \hat{T} + N(T_o + T_{model}) + \frac{N(N+1)}{2}T_{iter} \quad (3)$$

As T_{model} is orders of magnitude larger than the other times, the ratio of the two run times is

$$\frac{T'}{T} \approx 1 + \frac{N+1}{2}\frac{T_{iter}}{T_{model}} \quad (4)$$

Given the expected relative magnitudes of the model and optimisation iteration times, and *N* of order 10s or 100s, the increase in run time through this approach is actually negligible.

On the other hand, this scheme has several benefits. Apart from being simple to code, the optimisation algorithm, including the user-defined function evaluation subroutine, can be completely generic, and applied unmodified to different modelling systems. The only requirements on the modelling system are that, at the start of each simulation, it read in the parameter values requested by the optimisation code and adapt them to its standard input formats, then at the end of the simulation, computes and writes to file a single error metric value. The optimisation code and the model system could then remain separate, both controlled by some form of scripting scheme, for example. This means that having invested considerable time and resources in developing a complex modelling scheme, no major reconfiguration needs to be made to prepare it for optimisation in this manner, and then to re-implement the optimised modelling system in operational or production mode.

2.1 Cylc

The Cylc workflow engine (http://cylc.github.io/cylc/) was developed at NIWA to control the EcoConnect operational forecasting system and related environmental forecasting research systems, and is now used by several other institutions for similar purposes. EcoConnect manages incoming data feeds of real time atmospheric and stream flow data observations, as well as daily 144-hour global weather forecasts from the UK Met Office. These provide inputs for New Zealand regional data-assimilating numerical weather forecasts running on six hourly cycles and a daily global wave forecast, which in turn provide inputs for regional wave, storm surge and river flow models. EcoConnect also has a multitude of tasks to process multiple data streams for both data assimilation and verification, and to generate products for dissemination.

Cylc orchestrates tasks in complex cycling workflows, or “suites”, respecting the interdependencies between them. Rather than explicitly calling tasks in a defined sequence, Cylc manages a pool of autonomous tasks (proxies for the actual computing jobs), with defined dependencies on other tasks. Any task can start (i.e. submit its job to run) when all its dependencies are

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

met. These dependencies, which combine to form a dependency graph for the suite, can be specified by a set of statements of the form

A:*status* => B

which denotes that initiation of task B requires task A to have reach a specified status. The trigger status defaults to successful completion, but other conditions may be used, e.g. start of execution, submission to a batch scheduler job queue, submission or execution failure, or user-specified task output messages.

Tasks may repeat on date-time cycling sequences (e.g. representing successive forecast cycles) defined by ISO 8601 date-time recurrence expressions, with intra-cycle dependencies (for different operations needed for each new forecast) and inter-cycle dependencies (e.g. where a forecast is initialised from outputs from a previous cycle).

Tasks associated with more than one cycle can run concurrently, if the dependencies allow it (i.e. you don't have to finish one cycle before starting the next). So, for example, when restarting after an extended system downtime, a task that processes near-real-time data from external sources may be able to interleaveseveral cycles concurrently to clear the backlog of data, allowing for an efficient catch-up.

Cylc also supports integer cycling to control workflows that are not date-time-based (e.g. to apply processing operations to a set of different data sources, such as from multiple recording instruments).

A single suite can control all the separate tasks of a coupled system. Alternatively, separate suites can interact through tasks in one suite polling tasks in another suite on which they depend. An example of this is seen in the present implementation of NIWA's EcoConnect forecasting system, in which each of the forecast models is implemented in its own Cylc suite, so that a regional wave forecast suite, for example, will poll the relevant tasks in the regional weather forecast suite and the global wave forecast suite to determine when necessary inputs produced by those tasks are available. Cylc also supports clock and event triggers to allow triggering of tasks off the real-time clock and external events such as arrival of new datasets.

2.2 Implementation

We have developed a Cylc suite ("Cyclops") to perform optimisation of a modelling system that has itself been set up as a separate Cylc suite. In the example we describe below, the model suite controls a multi-year wave model hindcast, including the preprocessing of necessary model inputs (principally wind fields) and verification data (satellite altimeter data), running the Wavewatch code, postprocessing of model outputs, and generation of error statistics from comparisons of predicted and observed significant wave height fields.

Typically, a model suite will use time-based cycling to run, for example, at successive forecast cycles, or to break up a long simulation into a succession of time blocks. The optimisation suite, on the other hand, uses integer cycling to step through iterations.

There are several tasks controlled by the optimisation suite. One of these is responsible for running an optimisation algorithm to either identify an optimal parameter vector from data provide by previous model runs, or identify the next parameter vector that needs to be evaluated in that process. This main optimisation task within the suite is implemented with python code calling the NLopt python toolbox (Johnson, 2014).

NLopt includes a selection of optimisation algorithms: both "local" solvers, which aim to find the nearest local minimum to the starting point as efficiently as possible, and "global" solvers, which are designed to explore the full parameter space, giving high confidence in finding the optimal solution out of a possible multitude of local minima. NLopt includes algorithms capable of using derivative information where available, which is not the case in our application, and Cyclops is restricted to the derivative-free algorithms listed in Table 1.

We have assumed that the sequence of parameter vectors tested by an optimisation algorithm is deterministic. Several of the algorithms available in NLopt have some inherently stochastic component. It is, however, possible to make these algorithms "repeatably stochastic" by enforcing a fixed seed for the random number generator.

In NLopt, any combination of the following termination conditions can be set:

1. maximum number of iterations
2. absolute change in the parameter values less than a prescribed minimum
3. relative change in the parameter values less than a prescribed minimum
4. absolute change in the function value less than a prescribed minimum
5. relative change in the function value less than a prescribed minimum
6. function value less than a prescribed minimum

We have implemented python code that uses NLopt calls to seek a minimum of an objective function *f* that represents a non-negative model error metric. As described above, the user-defined function evaluation has been implemented as a generic python function $f(\vec{x})$ that simply searches a lookup table (stored in a file). If $\vec{x}$ is found in the table it returns the corresponding *f* value, otherwise it saves the vector $\vec{x}$ to a file and returns a negative *f* value. Any of the termination conditions listed above can be set by the user: the last of these can use a positive prescribed minimum *f* value as a convergence condition, while a negative *f* value signalling that the optimisation algorithm has stopped because a new parameter vector *x* needs to be evaluated externally by a model simulation.

At present a generic namelist format is used as output from Cyclops to supply the names and values of parameters to the model suite. This was chosen as convenient for use with Wavewatch, which uses this format directly - other model suites will need to include a customised task to process the namelist file into the model's usual input formats. Namelists can include named groups of parameters, which may be helpful in this process in cases where these groups need to be treated differently (e.g. affecting different model input files).

A "parameter definition" file is used to specify parameter names and their initial values, as used within the model. If a parameter is to be allowed to be adjusted by the optimisation suite, an allowable range is also set. This choice will generally require some experience with the particular model. Within the optimisation suite, these adjustable parameters will be scaled linearly to normalised parameters $\vec{x}$ that lie between 0 and 1. Fixed parameters can be include for convenience, so that their names and initial values will be written to the namelist file but these are ignored by the optimisation suite.

The major tasks carried out by Cyclops on each iteration are:

0. (first iteration only): **init**: write initial normalised parameters $\vec{x}_0$ to file, …
1. **optimise_step**: run the optimisation code, starting from $\vec{x}_0$ and evaluating every $\vec{x}$ in the sequence, until either a stopping criterion is met (in which case the task sends a "stop_iter" message), or a parameter set $\vec{x}$ is reached that is not in the lookup table so needs evaluating (signalled by a "next_iter" message)
2. **make_namelist**: Convert $\vec{x}$ to non-normalised parameters in a namelist file
3. **run_model**: Create a new copy of the model suite, copy the namelist file to it, and run it in non-daemon mode (i.e. so the task will not complete until the model suite shuts down). A new copy of the suite is made so that files created in one iteration do not overwrite those created on other iterations.
4. **update_table**: Read the resulting error value from the model suite, and update the lookup table

Within one iteration, the dependencies of the optimisation suite are simply:

optimise_step:next_iter => **make_namelist** => **run_model** => **update_table**

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

Geoscientific Model Development Discussions

to make these tasks run sequentially when no stop condition is met. The suite is made to iterate by setting a dependency on a previous cycle:

update_table[-P1] => **optimise_step**

(the notation –P1 denotes a negative displacement of one cycle period), while the stopping condition is handled by

optimise_step:stop_iter => **stop_iter**

Where the **stop_iter** task does a final wrap-up of the completed optimisation before the suite shuts down. For the purposes of good housekeeping, we can also add a **model_delete** task to delete each copy of the model suite once all its outputs have been used.

Figure 1 illustrates the workflow of the optimisation suite described above in graphical form.

2.3 Parallel iterations

For some DFO algorithms, at least some parts of the sequence of vectors tested is predetermined, and independent of the function values found at those points. For example, BOBYQA (which we chose to use in the test application described below) sets up a quadratic approximation by sampling the initial point, plus a pair of points on either side of it in each dimension. With N parameters, the first $2N+1$ iterations are spent evaluating these $2N+1$ fixed points, regardless of the function values obtained there. In such situations, the function values for each of these points could be evaluated simultaneously.

This can be done within Cylc by allowing tasks from multiple iteration cycles to run simultaneously. In practice, this means that multiple copies of the model suite are running simultaneously, to the extent allowed by resource allocation on the host machine(s). This makes it imperative that a new copy of the model suite is made for each iteration.

If parallel iterations are allowed, this means that at any time there are a certain set of parameter vectors for which the function values are still being determined (we can call this the "active" set). We can add another parameter vector to that set if it will be selected by the optimisation algorithm regardless of the function values at the active parameter vectors.

We would clearly like to determine that without needing specific knowledge of how the particular optimisation algorithm works. Instead we use a simple "empirical" method. To this end, we maintain a file (the "active file") listing the active vectors, and make an addition to the function evaluation subroutine, so that if it fails to find $\vec{x}$ in the lookup table, it then searches the active file and if it finds $\vec{x}$ there, assigns f a random positive value (in this application we don't re-initialise the random number generator with a fixed seed). Otherwise it writes $\vec{x}$ to file and returns a negative f to force the optimisation algorithm to stop as usual. The python code controlling the optimisation algorithm has also been modified. Now when the active file is empty it will act as before, but if there are active vectors will run a small set of repeated optimisations. If all of these result in the same choice of $\vec{x}$ value to be evaluated, a "next_iter" message is sent to trigger further tasks for this iteration cycle as before, since this choice is independent of the results for the active parameter vectors. If not, we do not have a definite $\vec{x}$ to evaluate, and we must wait until at least one of the presently active simulations has finished before trying again, so a "wait_iter" message is sent. But clearly this does not mean that the optimisation is complete.

This is handled within the Cylc suite by having several versions of the "optimise_step" task. Now "optimise_step _m" runs the optimisation algorithm when there are m active model simulations still running, with m ranging from 0 to a set maximum M. There are a more complex set of dependencies to ensure that this is the case. In particular, there is a condition

update_table[-P(m+1)] => **optimise_step_*m***

to ensure that the lookup table has been updated with the results of all completed (i.e. inactive) iterations, while the other requirements to trigger optimise_step _m are expressed as:

optimise_step_*m*+*1*:wait_iter | **optimise_step_*m*-*1***[-P1]:next_iter => **optimise_step_*m***

That is, we can either come to run the present iteration with *m* active simulations already running through the previous iteration (with *m*-1 active simulations) launching the m^{th} active simulation, or through trying to run with *m*+1 active simulations giving a wait condition.

Hence there are now a family of "optimise_step" tasks, represented in Figure 1 by octagonal figures which hide the individual family members, and their dependencies.

3 Application: a global wave hindcast based on ERA-Interim inputs

Here we describe a global wave simulation, using the Wavewatch III® model, forced by inputs from the ERA-Interim Reanalysis, covering the period from January 1979 to December 2016.

Such multi-year wave model simulations are a valuable means of obtaining wave climate information at spatial and temporal scales that are not generally available from direct measurements. It is rare for a particular location of interest to have a suitably long nearby *in situ* wave record, e.g. from a wave-recording buoy, to provide statistically reliable measures of climate variability on inter-annual time scales. And while satellite altimetry has provided near-global records of significant wave height that have been available for more than two decades, these have limited use for many climate applications for several reasons, including a return cycle that is too long to resolve typical weather cycles, limitations in providing nearshore measurements, and lack of directional information. Model simulations can in many cases overcome these limitations, but available measurements still play an essential role in calibrating and verifying the simulations.

In our case, one of the principal motivations for carrying out this hindcast is to investigate the role of wave-ice interactions in the interannual variability of Antarctic sea ice extent, which plays an important role in the global climate system. The ERA-Interim Reanalysis is a suitable basis for this work, providing a consistent long-term record, with careful control on any extraneous factors (e.g. changing data sources, or modelling methods) that might introduce artificial trends or biases into the records. While the ERA-Interim Reanalysis includes a coupled wave model, direct use of the wave outputs does not fully meet our requirements, which include the need for the wave hindcast to be independent of near-ice satellite wave, which were assimilated into the ERA-Interim Reanalysis. Hence we chose to carry out our own wave simulation, forced with ERA-Interim wind fields, but with no assimilation of satellite wave measurements.

3.1 Comparison of model outputs with altimeter data

Rather than being assimilated in the hindcast, satellite altimetry measurements of significant wave height were used as an independent source of model calibration. These were obtained from the IFREMER database of multi-mission quality-controlled and buoy-calibrated swath records (Queffeulou, 2004).

Swath records of significant wave height were first collocated to the hourly model outputs on the 1°×1° model grid. For each calendar month simulated, collocations were then accumulated in 3°×3° blocks of 9 neighbouring cells to produce error statistics, including model mean, altimeter mean, bias and root-mean-square error (RMSE), and correlation coefficient *R*. Spatial averages of these error statistics were taken over the full model domain between 65°S and 65°N (excluding polar regions with insufficient coverage).

The final error statistic used in the objective function was the spatially-averaged RMSE, normalised by the spatially-averaged altimeter mean, temporally averaged over the simulation period, excluding spinup.

3.2 Wavewatch parameters

For this simulation we used version 4.18 of the Wavewatch III® third generation wave model (Tolman, 2014). The model represents the sea state by the two-dimensional ocean wave spectrum $F(\vec{k},\vec{x},t)$, which gives the energy density of the wave field as a function of wavenumber $\vec{k}$, at each position $\vec{x}$ in the model grid and time *t* of the simulation.

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

The spectrum evolves subject to a radiative transfer equation

$$\frac{\partial N}{\partial t} + \vec{\nabla}_x \cdot (\dot{x}N) + \frac{\partial}{\partial k}(\dot{k}N) + \frac{\partial}{\partial \theta}(\dot{\theta}N) = \frac{S}{\sigma} \quad (5)$$

for the wave action $N(k,\theta,\vec{x},t) = F(\vec{k},\vec{x},t)/\sigma(k)$, where σ is the frequency associated with waves of wavenumber magnitude k through the linear dispersion relation, and θ is the propagation direction. The terms on the left hand side represent spatial advection, and the shifts in wavenumber magnitude and direction due to refraction by currents and varying water depth. The source term S on the right hand side represents all other processes that transfer energy to and from wave spectral components, including contributions from wind forcing, energy dissipation and weakly-nonlinear four wave interactions.

Adjustable parameters within the Wavewatch model that can influence a deep water global simulation such as the one described here are principally concentrated in the wind input and dissipation source terms. It is generally necessary to treat these two terms together as a 'package'. In this study we undertook two separate calibration exercises, based on two 'packages' of input/dissipation source terms, firstly that of Tolman and Chalikov (1996) (activated in Wavewatch by the ST2 switch), and secondly the Ardhuin et al (2010) formulation (using the ST4 switch).

In the Appendix we describe some of the details of these two packages. We also include some description of the WAM Cycle 4 (ST3) input source term formulation (Janssen, 1991), on which the ST4 input term is based, even though the ST3 package was not tested in this study.

In addition to the input and dissipation terms, the other main control on deep-water wave transformation is provided by weakly nonlinear four-wave interactions (Hasselmann, 1962). Unfortunately, acceptable run time requirements for multiyear simulations over extensive domains still preclude using a near-exact computation of these terms, such as the Webb, Resio, Tracy method (Webb, 1978;Tracy and Resio, 1982) that is available in spectral models including Wavewatch (van Vledder et al., 2000). Instead we use the much-simplified form of the Discrete Interaction Approximation (Hasselmann et al., 1985), treating its proportionality constant C as a tunable parameter.

Common to both optimisations, sea ice obstruction was turned on (FLAGTR=4) with non-default values for the critical sea ice concentrations $\epsilon_{c,0}$ and $\epsilon_{c,n}$ between which wave obstruction by ice varies between zero and total blocking: these were set to 0.25 and 0.75, respectively. All other available parameters beyond the input and dissipation terms were left with default settings, noting that shallow water processes, while activated, are not expected to have more than a negligible and localised influence on model outputs in a global simulation at 1° resolution.

For initial testing, in which two sets (ST2 and ST4) of optimisation parameters were compared, we used a one month (January 1997) spinup to a three month calibration period (February – April 1997). The selection of the calibration period from the full extent of the satellite record was arbitrary.

Relevant parameters used in the two calibrations are listed in Table 2 and Table 3, respectively, which refer to the parameter names as defined (more completely than we do here) in the Wavewatch user manual (Tolman, 2014), and as specified in namelist inputs to the model. These tables include the initial values of the parameters, the range over which they were allowed to vary, and the final optimised values. Some fixed parameters are also listed for completeness. For example, the input wind vertical level z_r (ST2) $= z_u$ (ST4) = 10 m is a property of the input data set, so was left fixed. Others were left fixed after an initial test confirmed that they had zero influence on the objective function, leaving 13 adjustable parameters for ST2 and 17 for ST4.

The selection of which parameters to tune, and the range over which they are allowed to vary, is an area where some (partly subjective) judgement is still required, based on some familiarity with the relevant model parameterisations. In this case, parameter ranges were chosen to be physically realistic, and to cover the range of parameter choices used in previous studies reported in the literature.

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

3.3 Optimisation settings

We elected to use the BOBYQA optimisation algorithm (Powell, 2009) for this study. Given that we expected Wavewatch to be already reasonably well-tuned for a global simulation such as our test case, we wished to use a local optimisation algorithm that could reach a solution to a problem with 10-20 variables in as few iterations as possible. Of the algorithms available in NLopt that were included in the intercomparison study of Rios and Sahinidis (2012), BOBYQA was found to be the most suitable in that respect.

Both optimisations were stopped when either the absolute change in (normalised) parameter values was less than 0.02, or the relative change in the objective function was less than 0.02. Following the two comparison simulations, the ST4 parameterisation was chosen for a final calibration, carried out over a 12 month period (January – December 1997) following a one-month spinup (December 1996). For this calibration, the same settings were initially used, but the ability of the optimisation suite to be restarted with revised stopping criteria was invoked to extend the optimisation with both criteria reduced to a value of 0.0001. This was a somewhat arbitrary choice made to observe the evolution of the solution. For practical applications the choice of stopping criteria should take into account the sensitivity of the objective function to measurement error in the data used for the calibration, to avoid unnecessary 'over-tuning' of the model.

The full hindcast, from January 1979 through December 2016 was then run using the optimised parameter set. Comparisons with altimeter data were made for each month from August 1991 onward.

Each Wavewatch simulation was run on 64 processors on a single core of an IBM Power6 machine. Other processing tasks within the suites were run on single processors. The resulting hindcast simulations required an average of approximately 25 minutes of wall clock time to complete each month of simulation.

4 Results

The BOBYQA algorithm develops a quadratic model of the objective function. To do so, the first iteration evaluates the objective function at the initial point, then perturbs each component in turn by a positive increment, then by an equal negative increment (leaving all other components at the initial value). This can be seen for the ST2 optimisation in Figure 2, in which the bottom panel shows the sequence of (normalised) parameter values tested. With 13 adjustable parameters, this amounts to 27 iterations in this preliminary phase. As this sequence of parameter values is fixed, independent of the resulting objective function values, all of the first 27 iterations could have been run simultaneously as detailed above, if permitted by the queuing system. We, however, applied a limit of 7 parallel iterations in line with anticipated resource limitations.

The 3-month ST2 optimisation only required a further 5 iterations after this initial phase to reach a stopping criterion. The ST2 default parameter settings used as the starting point for optimisation resulted in an objective function value of 0.1901, which was reduced to 0.1495 in the optimisation process.

In the optimal configuration, none of the tunable parameters were at either of the limits of their imposed range, indicating that convergence to a true minimum (at least locally) had been reached. Most of the parameters were only slightly modified from their initial values: the largest changes were in parameters b_0 (reduced from 0.0003 to 0.00022614) and b_1(0.47 to 0.27561), both influencing the low frequency dissipation term.

The ST4 3-month optimisation was initialised with the default settings from the TEST451 case reported by Ardhuin et al (2010), for which the objective function returned a value of 0.1427. Optimisation only managed to reduce this to 0.1422 (Figure 3), indicating that the default ST4 parameter set was already quite closely tuned for our case, having been selected by Ardhuin et al (2010) largely from broadly similar studies, i.e. global simulations (at 0.5° resolution) compared with altimeter records.

Three of the parameters ended the optimisation at one end of their allowed range, in each case at the same value at which it was initialised. The 16th adjustable parameter (s_B) controls the assumed directional spread of the dissipation spectrum, and the fact that it remained at its upper limit suggests that the optimisation may be improved by assuming the dissipation spectrum to

have a narrower directional distribution than anticipated. On the other hand, parameters 14 (C_{ds}^{BCK}) and 15 (C_{ds}^{HCK}) are associated with an alternative breaking formulation proposed by Filipot and Ardhuin (2012), who chose values $C_{ds}^{BCK} = 0.185$ and $C_{ds}^{HCK} = 1.5$ (and correspondingly, turned off the default saturation-based dissipation term by setting $C_{ds}^{sat} = 0$) whereas this term is turned off in the ST4 default, hence both were initially set to zero. On the face of it, one might think that the optimisation algorithm would have been free to explore solutions with positive values of these parameters, resulting in an optimal 'hybrid' total dissipation term. In fact the way the dissipation algorithm is coded, this form of the dissipation term is not computed at all in the event that $C_{ds}^{BCK} = 0.0$, which would have been the case when the BOBYQA algorithm explored sensitivity to C_{ds}^{HCK} in the initial stages. This means that our choice of initial values may have spuriously caused the BOBYQA algorithm to underestimate sensitivity to C_{ds}^{HCK}, and may have missed a distinct second local minimum (approximately corresponding to the parameter settings of Filipot and Ardhuin (2012)).

In the final 12 month ST4 optimisation, two additional parameters were allowed to vary that were fixed in the 3-month optimisation, bringing the number of adjustable parameters to 19. These were the critical sea ice concentration parameters $\epsilon_{c,0}$ and $\epsilon_{c,n}$ between which wave obstruction by ice varies between zero and total blocking: these were set to 0.25 and 0.75, respectively. Otherwise, the initial parameters (Table 4) again corresponded to the ST4 defaults, which in this case produced an error metric of 0.1436. At the termination after 89 iterations (with the more stringent stopping criteria), this had decreased to 0.1431.

Most of the resulting optimised parameters were close to the values obtained from the 3-month optimisation (Table 3). An exception was the 11th adjustable parameter, C_{turb} , scaling the strength of the turbulent contribution to dissipation, which finished the 3-month optimisation at 0.41298, but at 0.0 (the lower bound) in the 12 month simulations.

For this longer optimisation, we have additionally computed a measure of the sensitivity of the objective function, using the initial phase of the BOBYQA iterations to estimate the change in the (un-normalised) parameter required to produce a 0.1% change in the objective function. This is listed as "Delta" in the seventh column of Table 3, and provides a measure, at least in relative terms, of the bounds within which each parameter value has been determined.

The full hindcast, run from 1979 to 2016, could be compared with satellite data from August 1991 onward. The resulting bias in significant wave height, averaged over the August 1991 – December 2016 comparison period, is shown in Figure 4. Positive biases are obtained in latitudes south of 45°S, particularly south of Australia and in the South Atlantic. This is also seen in the vicinity of some island groups (notably French Polynesia, Micronesia, the Maldives, Aleutians, Carribean, Azores), which may be indicative of insufficient sub-grid scale obstruction. On the other hand, negative biases are seen near the western sides of major ocean basins, and in the "swell shadow" to the northeast of New Zealand. A similar pattern is seen in the results reported by Ardhuin et al (2010) for their TEST441 case (their Figure 9).

Normalised root-mean-square error (i.e. RMSE error divided by the observed mean) from the same comparison, again averaged over the period August 1991 – December 2016, is shown in Figure 5. Note that the objective function for our optimisation used this measure, spatially averaged over ocean waters between 61°S and 61°N. For the majority of the ocean surface, this lies in the range 0.08 – 0.14, but with higher values near some island chains and the western boundaries of ocean basins. Again, similar results were reported by Ardhuin et al (2010).

5 Discussion

In their review of methods used to tune Numerical Weather Prediction and climate models, Hourdin et al. (2017) observe that with the number and complexity of parameterisations to consider, the task of tuning these parameters was for a long time largely left to "expert judgement", and that objective methods have made a more recent appearance than in the statistical, engineering, and computing fields. The method we have presented here, along with the approaches of Severijns and Hazeleger (2005), Tett et al. (2013), Roach et al. (2017) described in the introduction, perform model tuning through the relatively direct

approach of defining and minimising a cost function. Our method has the advantage of employing a tool (Cylc) that is already becoming commonly used to control complex workflows for weather forecasting and climate modelling systems, to optimize the parameters of such a system under its control, in a way that is simple to implement, and flexible in choice of optimisation algorithm.

We have shown it to be a practical method for optimising 10-20 parameters in a model application of sufficient complexity to require several hours per simulation in a parallel processing computing environment. For applications that are yet more time-consuming, it is becoming increasingly common (Bellprat et al., 2012;Wang et al., 2014;Duan et al., 2017) to first build a surrogate model to provide a statistical emulator for the actual model, and then apply an optimisation algorithm to the surrogate model. Such multi-stage model optimisation frameworks are presently beyond the scope of our technique, but it may be worth considering whether the flexibility of our approach may also bring benefits within such frameworks. For example, it may be worth considering a hybrid approach of using a surrogate model to quantify the role of the full set of model parameters and perform an initial global optimisation, before switching to a method such as ours for a final refinement using the original model directly.

In our study we have restricted our attention to one local optimisation algorithm (BOBYQA), but our initial results suggest the need in some circumstances to apply a more global method. This is not difficult to do, with multiple algorithms, both global and local, implemented in Cyclops, but just not investigated in this initial study.

As we have seen, there remains a need for care with the choices of which parameters to attempt to optimise, and what bounds to set on their values. Most optimisation algorithms are intended for continuously variable parameters, and may rely on the objective function having a continuous dependence on these parameters. In many cases it is clear which parameters fall into this category, as opposed to discrete valued options. But in some cases, model code may make binary choices based on real parameters lying within discrete ranges, which may break this assumption. Hence the Cyclops optimisation suite is best employed in conjunction with a good understanding of the role each parameter plays in the model, and the interplay between them.

Conclusions

The Cyclops Cylc-based optimisation suite offers a flexible tool for tuning the parameters of any modelling system that has been implemented to run under the Cylc workflow engine. Minimal customisation of the modelling system is required beyond providing tasks to input and apply model parameter values in a simple namelist format, and output the value of the scalar error metric that is to be minimised. This then allows any of 16 optimisation algorithms (from the NLopt toolbox) to be applied to the optimisation. This optimisation suite is expected to be especially applicable to operational forecasting systems, where minimal re-configuration is required between “tuning” and “operational/production” versions of the forecast suite.

Results of the initial test case we have investigated, a global hindcast using a spectral wave model forced by ERA-Interim input fields, illustrate that the method is applicable to a modelling system of moderate complexity, both in terms of the number of parameters to tune, and the computational resources required, at least for the purposes of local optimisation to fine tune a model that already has a more-or-less well developed initial parameter set from previous studies. Investigations of systems that require a more global tuning approach, or are more computationally demanding remain for future work.

Code availability

Cyclops-v1.0 has been published through zenodo (https://doi.org/10.5281/zenodo.837907) under a Creative Commons Attribution Share-Alike 4.0 licence.

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

Appendix A: Wavewatch source term parameterisations

A.1 Tolman and Chalikov input + dissipation source term package

The input source term is defined as

$$S_{in}(k,\theta) = \sigma\beta N(k,\theta) \quad \text{(A1)}$$

Where β is a non-dimensional wind-wave interaction parameter, which has a parameterised dependence on wind speed and direction, through boundary layer properties influenced by the wave spectrum. These dependencies are, however, fully determined with no user-adjustable terms, so we omit the details here.

This input term was, however, adjusted by Tolman (2002) following a global test case to ameliorate an excessive dissipation of swell in weak or opposing winds, in which cases β can be negative. This is done by applying, when β is negative, a swell filtering scaling factor with a constant value X_s for frequencies below $0.6f_p$ (where f_p is the peak frequency), scaling linearly up to 1 at $0.8f_p$, with higher frequencies unmodified.

The same study also led to the introduction of a correction for the effects of atmospheric stability on wave growth identified by Kahma and Calkoen (1992) by replacing the wind speed u with an effective wind speed u_e, with

$$\left(\frac{u_e}{u}\right)^2 = 1 + c_1 \tanh(\max(0, f_1\{\mathcal{ST} - \mathcal{ST}_0\})) + c_2 \tanh\left(\max\left(0, f_1 \frac{c_1}{c_2}\{\mathcal{ST} - \mathcal{ST}_0\}\right)\right) \quad \text{(A2)}$$

where $\mathcal{ST}$ is a bulk stability parameter

$$\mathcal{ST} = \frac{hg}{u_h^2}\frac{T_a - T_s}{T_0} \quad \text{(A3)}$$

in terms of air, sea and reference temperatures T_a, T_s and T_0, respectively, and u_h the wind speed at reference height h = 10 m, with g the gravitational acceleration. As air and sea surface temperature fields are available from the ERA-Interim dataset, it was possible to apply this parametrisation, treating c_0, c_1, c_2, f_1 and $\mathcal{ST}_0$ as adjustable dimensionless parameters.

The dissipation term consists of a dominant low-frequency constituent, with an empirical frequency dependence parameterised by constants b_0, b_1, ϕ_{min} and a high-frequency term, parameterised by constants a_0, a_1, a_2, the details of which we leave for the Wavewatch manual (Tolman, 2014) and original references therein.

A.2 WAM Cycle 4 source term package

The input source term implemented in WAM Cycle4 by Janssen (1982) was based on the wave growth theory of Miles (1957). The starting point is the assumption the wind speed U has a logarithmic profile, so that if the wind fields input to the model are specified at elevation z_u , then

$$U(z_u) = \frac{u_*}{\kappa}\log\left(\frac{z_u}{z_1}\right) \quad \text{(A4)}$$

where u_* is the friction velocity, defined by the total wind stress $\tau = u_*^2$, κ is von Karman's constant, and z_1 is a roughness length modified by wave conditions:

$$z_1 = \frac{z_0}{\sqrt{1-\tau_w/\tau}} \quad \text{(A5)}$$

in which τ_w is the magnitude of the wave-supported stress, while

$$z_0 = \alpha_0 \tau / g \quad \text{(A6)}$$

with α_0 a tunable dimensionless parameter.

The wave-supported stress can be equated to the rate of momentum transfer between wind and waves:

$$\vec{\tau}_w = \int dk\, d\theta \frac{\vec{k}}{C} S_{in}(k,\theta) \quad \text{(A7)}$$

where c is the wave phase velocity

The WAM Cycle 4 input source term is then given by

$$S_{in}(k,\theta) = \frac{\rho_a}{\rho_w}\frac{\beta_{max}}{\kappa^2} e^Z Z^4 \left(\frac{u_*}{C} + z_\alpha\right)^2 [\max(\cos(\theta - \theta_u), 0)]^{p_{in}} \sigma N(k,\theta) + S_{out}(k,\theta) \quad \text{(A8)}$$

with

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

$$Z = \log(kz_1) + \frac{\kappa}{\cos(\theta-\theta_u)\left(\frac{u_*}{c}+z_\alpha\right)} \quad \text{(A9)}$$

In these terms ρ_a and $\rho_w$are the densities of air and water, β_{max} is a dimensionless constant, z_α is a wave age tuning parameter and p_{in} is a parameter controlling the directional dependence relative to the wind direction θ_u.

The inter-dependence of τ_w and S_{in}expressed in (A7) and (A8) creates an implicit functional dependence of u_* on U and τ_w/τ. In practice, this dependence can be tabulated, using the resolved model spectrum for the low-frequency ($k < k_{max}$) part of (A7), above which a f^{-5} diagnostic tail is assumed.

The S_{out} term represents a linear damping of swells, in the form (Bidlot, 2012):

$$S_{out}(k,\theta) = 2s_1\kappa\frac{\rho_a}{\rho_w}\left(\frac{u_*}{c}\right)^2\left[\cos(\theta-\theta_u) - \frac{\kappa C}{u^*\log(kz_0)}\right]\sigma N(k,\theta) \quad \text{(A10)}$$

With s_1 set to 1(0) to turn on(off) the damping.

Dissipation is represented in the form

$$S_{ds}(k,\theta) = C_{ds}\bar{\alpha}^2\bar{\sigma}\left[\delta_1\frac{k}{\bar{k}} + \delta_2\left(\frac{k}{\bar{k}}\right)^2\right]N(k,\theta) \quad \text{(A11)}$$

where C_{ds} is a dimensionless constant, and δ_1 and δ_2 are weighting parameters. These take values $C_{ds} = -1.33$, $\delta_1 = 0.5$ and $\delta_2 = 0.5$ in the ECMWF implementation of WAM as reported by Bidlot (2012), but are adjustable within Wavewatch. Mean wavelength and frequency are defined as

$$\bar{k} = \left[\frac{\int k^p N(k,\theta)d\vec{k}}{\int N(k,\theta)d\vec{k}}\right]^{1/p} \quad \text{(A12)}$$

and

$$\bar{\sigma} = \left[\frac{\int \sigma^p N(k,\theta)d\vec{k}}{\int N(k,\theta)d\vec{k}}\right]^{1/p} \quad \text{(A13)}$$

with $p = 0.5$ and $p = 1$ being the respective WAM defaults (Bidlot, 2012) while mean steepness is

$$\bar{\alpha} = E\bar{k}^2 \quad \text{(A14)}$$

A.3 Ardhuin (2010) source term package

This package introduces a saturation-based dissipation term. In order to accommodate this, the WAM Cycle 4 input source function is modified by replacing u_* in (A8) with a frequency-dependent form

$$\left(u_*'(k)\right)^2 = \left|u_*^2 - |s_u|\left|\int_0^k dk'\int d\theta\frac{\vec{k'}}{c}S_{in}(k',\theta)\right|\right| \quad \text{(A15)}$$

In which $s_u \approx 1$ is a sheltering coefficient, to allow for balance with a saturation-based dissipation term. Also, a limit can be placed on the roughness length z_0, replacing (A6) with

$$z_0 = \min(\alpha_0\,\tau/g\,, z_{0,\max}) \quad \text{(A16)}$$

The swell dissipation parameterisation of Ardhuin et al. (2009) is used, consisting of terms

$$S_{out,visc}(k,\theta) = -s_5\frac{\rho_a}{\rho_w}\left[2k\sqrt{2\nu_a\sigma}\right]N(k,\theta) \quad \text{(A17)}$$

and

$$S_{out,turb}(k,\theta) = -\frac{\rho_a}{\rho_w}\left[16f_e\sigma^2 u_{orb,s}/g\right]N(k,\theta) \quad \text{(A18)}$$

due to effects of the viscous and turbulent boundary layers respectively. The latter depends on the significant surface orbital velocity

$$u_{orb,s} = 2[\int dkd\theta\,\sigma^3 N(k,\theta)]^{1/2} \quad \text{(A19)}$$

while ν_a is air viscosity and s_5 is a tunable coefficient of order 1. The two terms are combined in weighted form

$$S_{out}(k,\theta) = r_-S_{out,vis}(k,\theta) + r_+S_{out,turb}(k,\theta) \quad \text{(A20)}$$

with weights

$$r_{\pm} = 0.5(1 \pm \tanh((Re - Re_c')/s_7)) \quad \text{(A21)}$$

depending on a modified air-sea boundary layer Reynolds number

$$Re = 2u_{orb,s}H_s/\nu_a \quad \text{(A22)}$$

which is taken to have a threshold value depending on significant wave height:

$$Re_c' = Re_c(4m/H_s)^{1-s_6} \quad \text{(A23)}$$

The turbulent dissipation term is parameterised to depend on wind speed and direction:

$$f_e = s_1 f_{e,GM} + [|s_3| + s_2\cos(\theta - \theta_u)]u_*/u_{orb} \quad \text{(A24)}$$

based on the friction factor $f_{e,GM}$ from the Grant and Madsen (1979) theory of oscillatory boundary layer flow over a rough surface.

The dissipation term is based on the saturation of the wave spectrum, and takes the form

$$S_{ds}(k,\theta) = \sigma\frac{C_{ds}^{sat}}{B_r^2}[\delta_d \max(B(k) - B_r, 0)^2 + (1 - \delta_d)\max(B'(k,\theta) - B_r, 0)^2]N(k,\theta) + S_{bk,cu}(k,\theta) + S_{turb}(k,\theta) \quad \text{(A25)}$$

where the dissipation spectrum is integrated over a limited direction range, i.e.

$$B'(k,\theta) = \int_{\theta-\Delta_\theta}^{\theta+\Delta_\theta} \sigma k^3 \cos^{sB}(\theta - \theta')N(k,\theta)d\theta' \quad \text{(A26)}$$

and

$$B(k) = \max(B'(k,\theta), \theta \in [0,2\pi]) \quad \text{(A27)}$$

The cumulative breaking term, associated with large scale breakers overtaking short waves, is

$$S_{bk,cu}(k,\theta) = \frac{-14.2C_{cu}}{\pi^2}N(k,\theta)\int_0^{r_{cu}^2 k} dk' \int_0^{2\pi} d\theta' \max\left\{\sqrt{B(f',\theta')} - \sqrt{B_r}, 0\right\}^2 \quad \text{(A28)}$$

Where $r_{cu} = 0.5$ and C_{cu} is a tuning coefficient.

The turbulent dissipation term is

$$S_{turb}(k,\theta) = -2C_{turb}\sigma\cos(\theta_u - \theta)k\frac{\rho_a u_*^2}{g\rho_w}N(k,\theta) \quad \text{(A29)}$$

An alternative breaking formulation (Filipot and Ardhuin, 2012) based on a bore model uses a dissipation rate per unit crest length of

$$\epsilon_{CK} = \frac{1}{4}\rho_w g\left[\frac{C_{ds}^{BCK}H}{\tanh(kh)C_{ds}^{HCK}}\right]^3\sqrt{\frac{gk}{\tanh(kh)}} \quad \text{(A30)}$$

Competing interests

The authors declare that they have no conflict of interest.

Author contribution

Richard Gorman developed the Cyclops optimisation suite, carried out all simulations, and prepared the manuscript. Hilary Oliver is the principal developer of Cylc, assisted with the design of the optimisation suite, and contributed to the content of the manuscript.

Acknowledgements

This work was supported by Strategic Science Investment Funding provided to NIWA from the New Zealand Ministry of Business, Innovation and Employment (MBIE). The authors wish to acknowledge the contribution of the New Zealand eScience Infrastructure (NeSI: http://www.nesi.org.nz) to the results of this research. New Zealand's national compute and

analytics services and team are supported by NeSI and funded jointly by NeSI's collaborator institutions and through MBIE. We appreciate feedback on the manuscript from Lettie Roach

References

Ardhuin, F., Chapron, B., and Collard, F.: Observation of swell dissipation across oceans, Geophysical Research Letters, 36, 10.1029/2008GL037030, 2009.

Ardhuin, F., Rogers, E., Babanin, A. V., Filipot, J.-F., Magne, R., Roland, A., van der Westhuysen, A., Queffeulou, P., Lefevre, J.-M., Aouf, L., and Collard, F.: Semiempirical dissipation source functions for ocean waves. Part I: definition, calibration, and validation, Journal of Physical Oceanography, 40, 1917-1941, doi:10.1175/2010JPO4324.1, 2010.

Bellprat, O., Kotlarski, S., Lüthi, D., and Schär, C.: Objective calibration of regional climate models, Journal of Geophysical Research: Atmospheres, 117, n/a-n/a, 10.1029/2012JD018262, 2012.

Bidlot, J.-R.: Present status of wave forecasting at E.C.M.W.F., ECMWF Workshop on Ocean Waves, European Centre for Medium-range Weather Forecasts, 25 - 27 June 2012, 2012.

Brent, R.: Algorithms for Minimization without Derivatives, Prentice-Hall, 1972.

da Silva Santos, C. H., Gonçalves, M. S., and Hernandez-Figueroa, H. E.: Designing novel photonic devices by bio-inspired computing, IEEE Photonics Technology Letters, 22, 1177–1179, 2010.

Duan, Q., Di, Z., Quan, J., Wang, C., Gong, W., Gan, Y., Ye, A., Miao, C., Miao, S., Liang, X., and Fan, S.: Automatic model calibration: a new way to improve numerical weather forecasting, Bulletin of the American Meteorological Society, 98, 959-970, 10.1175/bams-d-15-00104.1, 2017.

Filipot, J.-F., and Ardhuin, F.: A unified spectral parameterization for wave breaking: From the deep ocean to the surf zone, J. Geophys. Res., 117, C00J08, 10.1029/2011jc007784, 2012.

Gablonsky, J. M., and Kelley, C. T.: A locally-biased form of the DIRECT algorithm, J. Global Optimization, 21, 27-37, 2001.

Grant, W. D., and Madsen, O. S.: Combined wave and current interaction with a rough bottom, Journal of Geophysical Research, 84, 1797-1808, 1979.

Hasselmann, K.: On the non-linear energy transfer in a gravity-wave spectrum. Part 1. General theory, Journal of Fluid Mechanics, 12, 481-500, 1962.

Hasselmann, S., Hasselmann, K., Allender, J. H., and Barnett, T. P.: Computations and parameterizations of the nonlinear energy transfer in a gravity wave spectrum. Part II: Parameterizations of the nonlinear energy transfer for application in wave models, Journal of Physical Oceanography, 15, 1378-1391, 1985.

Hendrix, E. M. T., Ortigosa, P. M., and García, I.: On success rates for controlled random search, Journal of Global Optimization, 21, 239-263, 10.1023/a:1012387510553, 2001.

Hourdin, F., Mauritsen, T., Gettelman, A., Golaz, J.-C., Balaji, V., Duan, Q., Folini, D., Ji, D., Klocke, D., Qian, Y., Rauser, F., Rio, C., Tomassini, L., Watanabe, M., and Williamson, D.: The art and science of climate model tuning, Bulletin of the American Meteorological Society, 98, 589-602, 10.1175/bams-d-15-00135.1, 2017.

Janssen, P. A. E. M.: Quasilinear approximation for the spectrum of wind-generated surface waves, Journal of Fluid Mechanics, 117, 493-506, 1982.

Janssen, P. A. E. M.: Quasi-linear theory of wind wave generation applied to wave forecasting, Journal of Physical Oceanography, 21, 1631-1642, 1991.

Jones, D. R., Perttunen, C. D., and Stuckmann, B. E.: Lipschitzian optimization without the lipschitz constant, J. Optimization Theory and Applications, 79, 157, 1993.

Kaelo, P., and Ali, M. M.: Some variants of the controlled random search algorithm for global optimization, J. Optim. Theory Appl., 130, 2006.

Kahma, K. K., and Calkoen, C. J.: Reconciling discrepancies in the observed growth of wind-generated waves, Journal of Physical Oceanography, 22, 1389-1405, doi:10.1175/1520-0485(1992)022<1389:RDITOG>2.0.CO;2, 1992.

Miles, J. W.: On the generation of surface waves by shear flows, Journal of Fluid Mechanics, 3, 1957.

Nelder, J. A., and Mead, R.: A simplex method for function minimization, Computer Journal, 7, 308-313, 1965.

Powell, M. J. D.: A direct search optimization method that models the objective and constraint functions by linear interpolation, in: Advances in Optimization and Numerical Analysis, edited by: Gomez , S., and Hennart, J.-P., Kluwer Academic, Dordrecht, 51-67, 1994.

Powell, M. J. D.: The NEWUOA software for unconstrained optimization without derivatives Proc. 40th Workshop on Large Scale Nonlinear Optimization Erice, Italy, 2004,

Powell, M. J. D.: The BOBYQA algorithm for bound constrained optimization without derivatives, Department of Applied Mathematics and Theoretical Physics, Cambridge, England, 2009.

Price, W. L.: Global optimization by controlled random search, J. Optim. Theory Appl., 40, 333-348, 1983.

Queffeulou, P.: Long-Term Validation of Wave Height Measurements from Altimeters, Marine Geodesy, 27, 495-510, 10.1080/01490410490883478, 2004.

Rinnooy Kan, A. H. G., and G. T. Timmer, G. T.: Stochastic global optimization methods, Mathematical Programming, 39, 27-78, 1987.

Rios, L. M., and Sahinidis, N. V.: Derivative-free optimization: a review of algorithms and comparison of software implementations, Journal of Global Optimization, 56, 1247-1293, 10.1007/s10898-012-9951-y, 2012.

Roach, L. A., Tett, S. F. B., Mineter, M. J., Yamazaki, K., and Rae, C. D.: Automated parameter tuning applied to sea ice in a global climate model, Climate Dynamics, 10.1007/s00382-017-3581-5, 2017.

Rowan, T.: Functional Stability Analysis of Numerical Algorithms, Department of Computer Sciences, University of Texas at Austin, Austin, Texas, USA, 1990.

Runarsson, T. P., and Yao, X.: Search biases in constrained evolutionary optimization, IEEE Trans. on Systems, Man, and Cybernetics Part C: Applications and Reviews, 35, 233-243, 2005.

Seong, C., Her, Y., and Benham, B.: Automatic Calibration Tool for Hydrologic Simulation Program-FORTRAN Using a Shuffled Complex Evolution Algorithm, Water, 7, 503, 2015.

Severijns, C. A., and Hazeleger, W.: Optimizing Parameters in an Atmospheric General Circulation Model, Journal of Climate, 18, 3527-3535, 10.1175/jcli3430.1, 2005.

Tett, S. F. B., Mineter, M. J., Cartis, C., Rowlands, D. J., and Liu, P.: Can Top-of-Atmosphere Radiation Measurements Constrain Climate Predictions? Part I: Tuning, Journal of Climate, 26, 9348-9366, 10.1175/JCLI-D-12-00595.1, 2013.

Tolman, H. L., and Chalikov, D.: Source terms in a third-generation wind wave model, Journal of Physical Oceanography, 26, 2497-2518, 1996.

Tolman, H. L.: Validation of WAVEWATCH III version 1.15 for a global domain, NOAA / NWS / NCEP / OMB, Technical Note Technical Note 213, 33, 2002.

Tolman, H. L.: User manual and system documentation of WAVEWATCH-III version 4.18, NOAA / NWS / NCEP / OMB, Technical Note Technical Note 316, 282, 2014.

Tracy, B. A., and Resio, D. T.: Theory and calculations of the nonlinear energy transfer between sea waves in deep water, US Army Engineer Waterways Experiment Station, Vicksburg, MS, WES Report 11, 1982.

van Vledder, G. P., Herbers, T. H. C., Jensen, R., Resio, D., and Tracy, B.: Modeling of nonlinear quadruplet wave-wave interactions in operational coastal wave models, Proceedings of the 27th International Conference on Coastal Engineering, Sydney, Australia, 2000, 797-811,

Voosen, P.: Climate scientists open up their black boxes to scrutiny, Science, 354, 401-402, 10.1126/science.354.6311.401, 2016.

Wang, C., Duan, Q., Gong, W., Ye, A., Di, Z., and Miao, C.: An evaluation of adaptive surrogate modeling based optimization with two benchmark problems, Environmental Modelling & Software, 60, 167-179, http://dx.doi.org/10.1016/j.envsoft.2014.05.026, 2014.

Webb, D. J.: Nonlinear transfer between sea waves, Deep Sea Research, 25, 279-298, 1978.

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

Geoscientific Model Development Discussions

Table 1 Derivative-Free Optimisation algorithms from the NLopt toolbox supported in the Cyclops optimisation suite

Global:
DIRECT: Dividing RECTangles (Jones et al., 1993)
DIRECT-L: Dividing RECTangles, locally optimised (Gablonsky and Kelley, 2001)
DIRECT-L-RAND: a slightly randomised variant of DIRECT-L (Johnson, 2014)
CRS: Controlled Random Search (Hendrix et al., 2001)
CRS2: Controlled Random Search (Price, 1983)
CRS2-LM: Controlled Random Search with Local Mutation (Kaelo and Ali, 2006)
MLSL: Multi-Level Single-Linkage (Rinnooy Kan and G. T. Timmer, 1987)
ISRES: Improved Stochastic Ranking Evolution Strategy (Runarsson and Yao, 2005)
ESCH: Evolutionary algorithm (da Silva Santos et al., 2010)
Local:
COBYLA: Constrained Optimization BY Linear Approximations (Powell, 1994)
BOBYQA: Bounded Optimization BY Quadratic Approximation (Powell, 2009)
NEWUOA: Unconstrained Optimization (Powell, 2004)
NEWUOA-BOUND: a bounded variant of NEWUOA (Johnson, 2014)
PRAXIS: Principal Axis (Brent, 1972)
Nelder-Mead Simplex (Nelder and Mead, 1965)
Sbplx: Nelder-Mead applied on a sequence of subspaces (Rowan, 1990)

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

Geoscientific Model Development Discussions

Table 2. Parameters used to calibrate the simulation using the source term package of Tolman and Chalikov (1996), for February through April 1997. The first two columns list the parameter as defined in the Wavewatch v4.18 user manual (Tolman, 2014), and as specified in Wavewatch namelist input (with namelist groupings in bold). Lower and upper bounds are specified for parameters adjusted during calibration, along with their final values, and the corresponding index n of the normalised parameter vector. Other parameters were fixed at the initial value.

Parameter	Code variable	Initial	Lower bound	Upper bound	Final	*n*
	SIN2:					
z_r	ZWND	10.0				
X_s	SWELLF	0.1	0.0	1.0	0.11449	1
c_0	STABSH	1.38	1.0	1.8	1.4009	2
$\mathcal{ST}_0$	STABOF	-0.01	-0.02	-0.001	-0.010247	3
c_1	CNEG	-0.01	-0.02	-0.001	-0.010261	4
c_2	CPOS	0.01	0.001	0.02	0.0097342	5
$-f_1$	FNEG	150.0	100.0	200.0	148.61	6
	SDS2:					
a_0	SDSA0	4.8	4.0	6.0	4.8036	7
a_1	SDSA1	1.7×10^{-4}	1.0×10^{-4}	5.0×10^{-3}	1.7017×10^{-4}	8
a_2	SDSA2	2.0	1.0	4.0	2.0094	9
b_0	SDSB0	0.3E-3	-0.01	0.01	0.00022614	10
b_1	SDSB1	0.47	0.2	1.0	0.27561	11
ϕ_{min}	PHIMIN	0.003	0.002	0.005	0.0029775	12
	SNL1:					
C	NLPROP	2.5×10^{-7}	2.4×10^{-7}	2.8×10^{-7}	2.49867×10^{-7}	13
	MISC:					
$\epsilon_{c,0}$	CICE0	0.25				
$\epsilon_{c,n}$	CICEN	0.75				
	FLAGTR	4				

Table 3. As for Table 2, but for parameters used to calibrate the simulation using the source term package of Ardhuin et al (2010), for February through April 1997.

Parameter	Code variable	Initial	Lower bound	Upper bound	Final	*n*
	SIN4:					
z_u	ZWND	10.0				
α_0	ALPHA0	0.0095				
β_{max}	BETAMAX	1.52	1.0	2.0	1.5201	1
p_{in}	SINTHP	2.0				
z_α	ZALP	0.006				

s_u	TAUWSHELTER	1.0	0.0	1.5	0.95649	2
s_0	SWELLFPAR	1				
s_2	SWELLF	0.8	0.5	1.2	0.80014	3
s_1	SWELLF2	-0.018	-0.03	-0.01	-0.018205	4
s_3	SWELLF3	0.015	0.01	0.02	0.014771	5
Re_c	SWELLF4	1.0×10^5	0.8×10^5	1.5×10^5	0.99707×10^5	6
s_5	SWELLF5	1.2	0.8	1.6	1.2085	7
s_6	SWELLF6	0.0				
s_7	SWELLF7	2.3×10^5	0.0	4.0×10^5	2.2554×10^5	8
z_r	Z0RAT	0.04				
$z_{0.max}$	Z0MAX	0.0				
	SINBR	0.0				
	SDS4:					
	SDSC1	0.0				
p	WNMEANP	0.5				
	FXPM3	4.0				
f_{FM}	FXFM3	9.9				
C_{ds}^{sat}	SDSC2	-2.2×10^{-5}	-2.5×10^{-5}	0.0	-2.1541×10^{-5}	9
C_{cu}	SDSCUM	-0.40344	-0.5	0.0	-0.40186	10
B_0	SDSC4	1.0				
C_{turb}	SDSC5	0.0	0.0	1.2	0.41298	11
δ_d	SDSC6	0.3	0.0	1.0	0.26135	12
B_r	SDSBR	0.0009	0.0008	0.0010	0.00090472	13
	SDSBR2	0.8				
p^{sat}	SDSP	2.0				
	SDSISO	2				
C_{ds}^{BCK}	SDSBCK	0.0	0.0	0.2	0.0	14
	SDSABK	1.5				
	SDSPBK	4.0				
	SDSBINT	0.3				
C_{ds}^{HCK}	SDSHCK	0.0	0.0	2.0	0.0	15
Δ_θ	SDSDTH	80.0				
s_B	SDSCOS	2.0	0.0	2.0	2.0	16
	SDSBRF1	0.5				
	SDSBRFDF	0				
	SDSBM0	1.0				
	SDSBM1	0.0				

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

	SDSBM2	0.0				
	SDSBM3	0.0				
	SDSBM4	0.0				
	WHITECAPWIDTH	0.3				
	SNL1:					
C	NLPROP	2.5×10^{-7}	2.4×10^{-7}	2.8×10^{-7}	2.5108×10^{-7}	17
	MISC:					
$\epsilon_{c,0}$	CICE0	0.25				
$\epsilon_{c,n}$	CICEN	0.75				
	FLAGTR	4				

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

Geoscientific Model Development Discussions

Table 4. As for Table 2, but for parameters used to calibrate the simulation using the source term package of Ardhuin et al (2010), for Jan-Dec 1997. The "Delta" value in the seventh column is the estimated change in the (un-normalised) parameter required to produce a 0.1% change in the objective function.

Parameter	Code variable	Initial	Lower bound	Upper bound	Final	Delta	*n*
	SIN4:						
z_u	ZWND	10.0					
α_0	ALPHA0	0.0095					
β_{max}	BETAMAX	1.52	1.0	2.0	1.5194	0.02498	1
p_{in}	SINTHP	2.0					
z_α	ZALP	0.006					
s_u	TAUWSHELTER	1.0	0.0	1.5	0.9339	0.2706	2
s_0	SWELLFPAR	1					
s_2	SWELLF	0.8	0.5	1.2	0.8224	0.0206	3
s_1	SWELLF2	-0.018	-0.03	-0.01	-0.01721	0.00064	4
s_3	SWELLF3	0.015	0.01	0.02	0.01526	0.00042	5
Re_c	SWELLF4	1.0×10^5	0.8×10^5	1.5×10^5	0.9888×10^5	0.2328×10^5	6
s_5	SWELLF5	1.2	0.8	1.6	0.9360	0.3974	7
s_6	SWELLF6	0.0					
s_7	SWELLF7	2.3×10^5	0.0	4.0×10^5	2.2433×10^5	0.7911×10^5	8
z_r	Z0RAT	0.04					
$z_{0.max}$	Z0MAX	0.0					
	SINBR	0.0					
	SDS4:						
	SDSC1	0.0					
p	WNMEANP	0.5					
	FXPM3	4.0					
f_{FM}	FXFM3	9.9					
C_{ds}^{sat}	SDSC2	-2.2×10^{-5}	-2.5×10^{-5}	0.0	-2.1433×10^{-5}	0.0087×10^{-5}	9
C_{cu}	SDSCUM	-0.40344	-0.5	0.0	-0.40194	0.02145	10
B_0	SDSC4	1.0					
C_{turb}	SDSC5	0.0	0.0	1.2	0.0	-	11
δ_d	SDSC6	0.3	0.0	1.0	0.2736	0.0928	12
B_r	SDSBR	9.0×10^{-4}	8.0×10^{-4}	10.0×10^{-4}	8.9788×10^{-4}	0.0951×10^{-4}	13
	SDSBR2	0.8					
p^{sat}	SDSP	2.0					
	SDSISO	2					
C_{ds}^{BCK}	SDSBCK	0.0	0.0	0.2	0.0	-	14
	SDSABK	1.5					
	SDSPBK	4.0					
	SDSBINT	0.3					
C_{ds}^{HCK}	SDSHCK	0.0	0.0	2.0	0.0	-	15
Δ_θ	SDSDTH	80.0					
s_B	SDSCOS	2.0	0.0	2.0	2.0	0.0757	16
	SDSBRF1	0.5					
	SDSBRFDF	0					
	SDSBM0	1.0					

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

Geoscientific Model Development Discussions

	SDSBM1	0.0					
	SDSBM2	0.0					
	SDSBM3	0.0					
	SDSBM4	0.0					
	WHITECAPWIDTH	0.3					
	SNL1:						
C	NLPROP	2.5×10^{7}	2.4×10^{7}	2.8×10^{7}	2.5181×10^{7}	0.1191×10^{7}	17
	MISC:						
$\epsilon_{c,0}$	CICE0	0.25	0.15	0.45	0.2413	0.1285	18
$\epsilon_{c,n}$	CICEN	0.75	0.55	0.85	0.7521	0.2358	19
	FLAGTR	4					

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

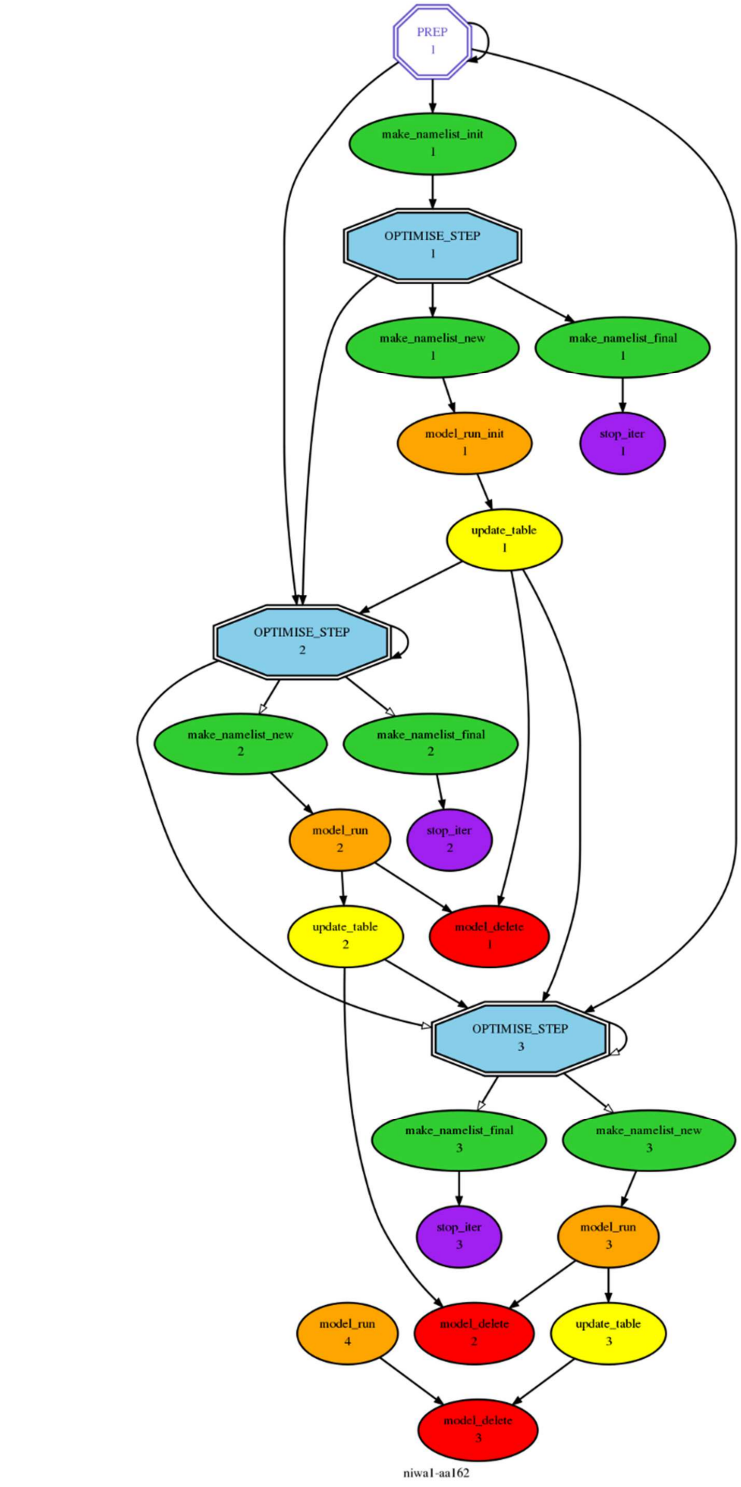

Figure 1: Dependency graph for the Cyclops optimisation suite, showing the first three iteration cycles. Numerals refer to the iteration number. Ellipses represent individual tasks, while the octagonal boxes represent families of tasks, for the case where parallel simulations are allowed. Arrows represent dependency, in that a task at the head of an arrow depends on the task at the tail of the arrow meeting a specified condition (by default, this means completing successfully) before it can start.

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

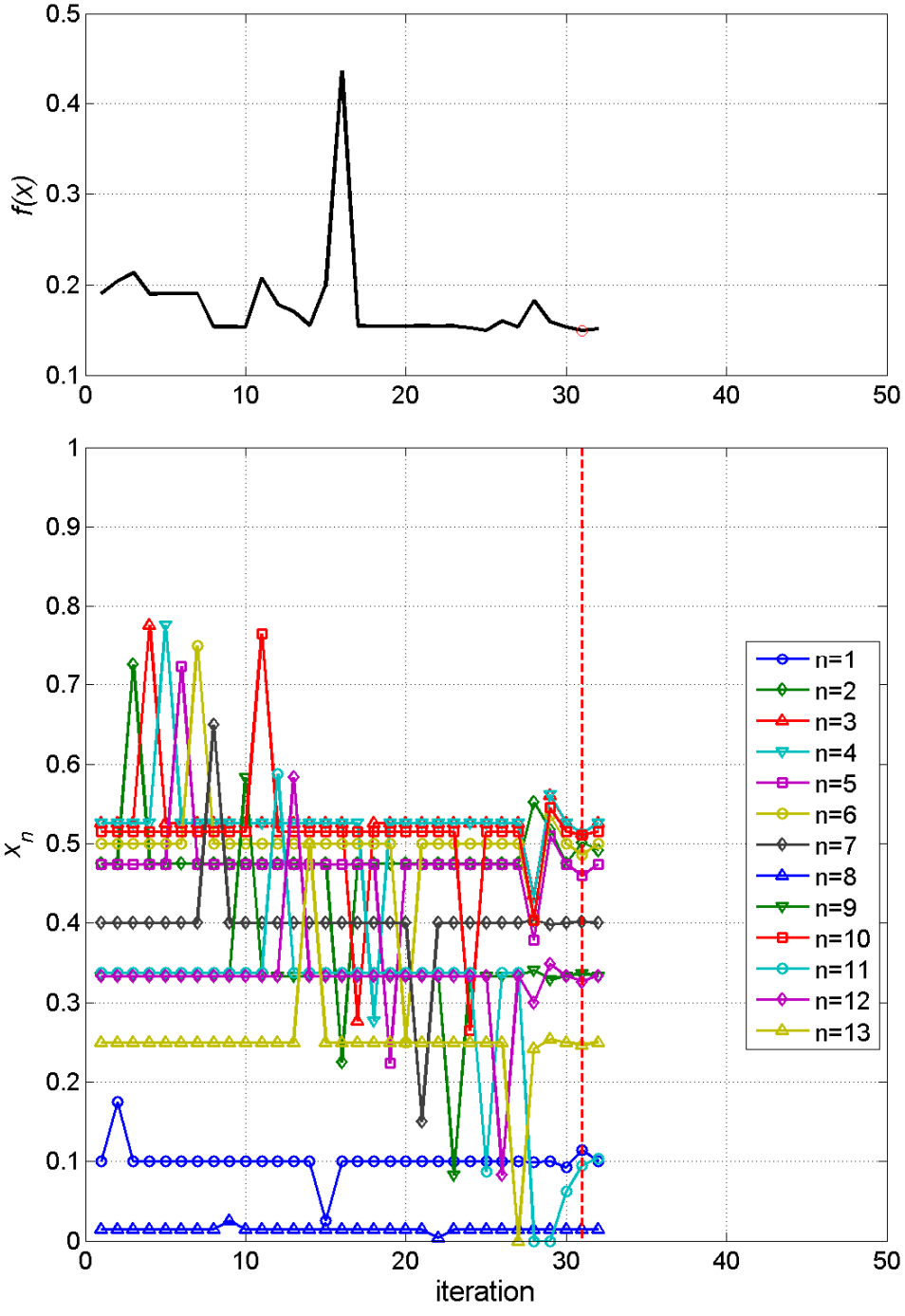

Figure 2: Sequence of objective function values (top) and parameter vector components (bottom) at each iteration in the three month (February – April 1997) ST2 calibration. The red dashed line marks the optimal solution found.

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

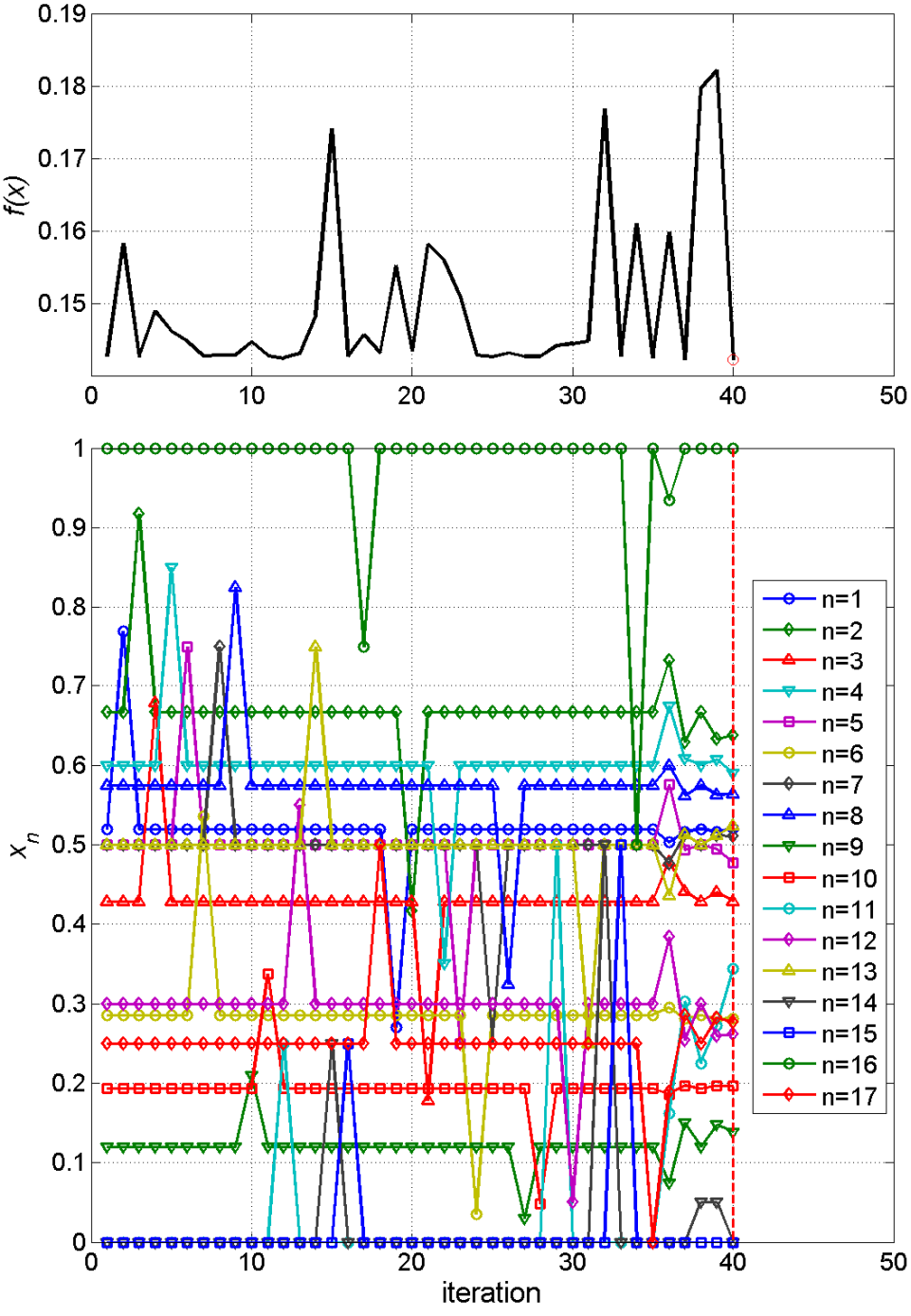

Figure 3: Sequence of objective function values (top) and parameter vector components (bottom) at each iteration in the three month (February – April 1997) ST4 calibration. The red dashed line marks the optimal solution found.

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

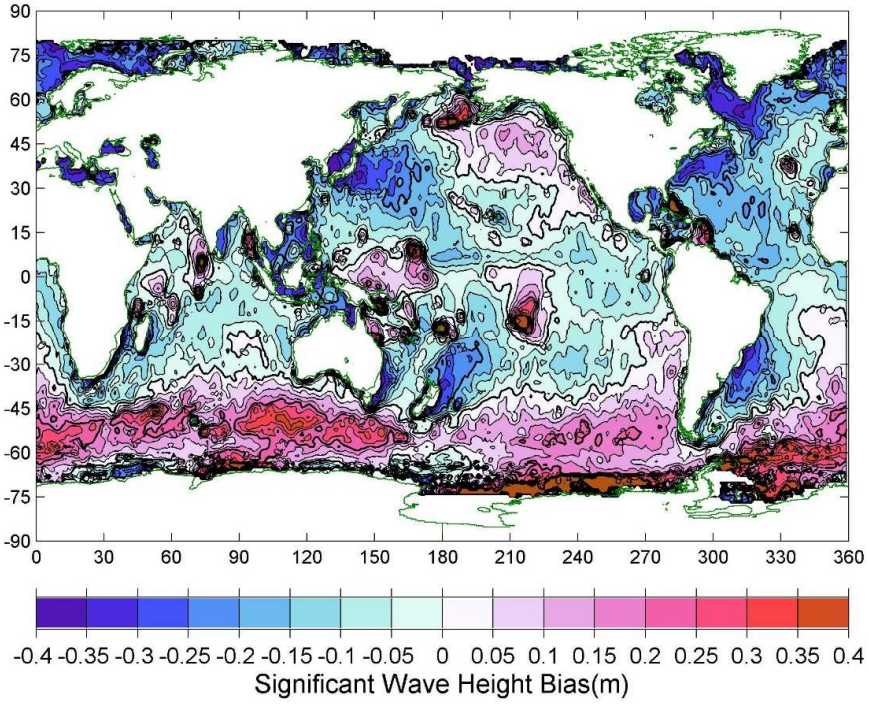

Figure 4: Bias in significant wave height from the hindcast compared with satellite altimeter measurements, over the period August 1991 – December 2016.

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-185
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 21 September 2017

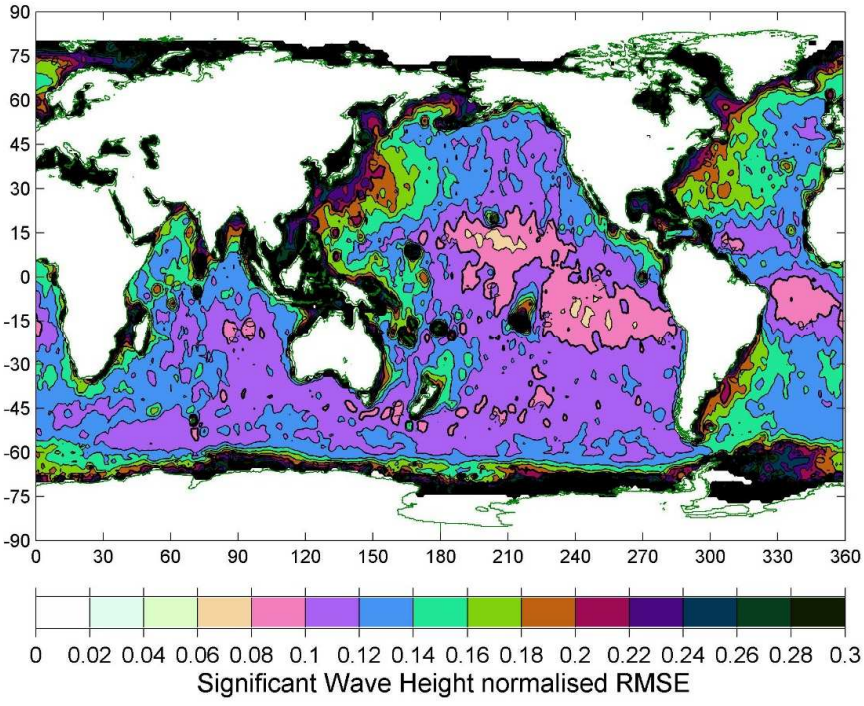

Figure 5: Normalised root-mean-square error in significant wave height from the hindcast compared with satellite altimeter measurements, over the period August 1991 – December 2016.