OCEANS AND ATMOSPHERE
www.csiro.au

User guide for the CSIRO Environmental Modelling Suite (EMS): Optical, biogeochemical, and carbon chemistry models.

Mark Baird, Mathieu Mongin, Karen Wild-Allen, Barbara Robson, Jennifer Skerratt, Farhan Rizwi

GitHub Version 1.2.0.

Local SVN 6539

6th May 2020

Citation

Baird ME, Mongin M Wild-Allen KA Robson B Skerratt J. Rizwi F. 2018. User guide for the CSIRO Environmental Modelling Suite (EMS): Optical, biogeochemical, and carbon chemistry models.

Copyright

Further information, acknowledgments and code access

The code is available from: https://github.com/csiro-coasts/EMS. All conditions for the use of the code can be found on the GitHub archive, as well as acknowledgement of funding sources and authors of the CSIRO EMS code. Thank you

Contents

Figures

Tables

1 Introduction

The CSIRO Environmental Modelling System (EMS) is a modelling tool that allows for the coupling of hydrodynamic / sediment / optical / biogeochemical (BGC) processes. It is has been designed in particular for shallow water systems, where processes at the sediment interface and below affect the water column, and vice versa.

One feature of EMS is the ability to include / exclude biogeochemical processes using the same model executable (or version), but different input configuration files. For example, the same executable can run a model of a tropical region with coral reefs, as well as a temperature region with multiple seagrass species or to just model the optical properties of the water column. The inclusion of model processes is therefore an important component of the model configuration, and makes up a significant portion of this User Guide.

This documentation is designed to be used only for user options that are not detailed in the much more extensive SHOC / COMPAS user guide (https://research.csiro.au/cem/software/ems/ems-documentation/). In particular, details regarding model running, model grids, advection and diffusion schemes, boundary conditions, as well as input / output formats are, are all documented in the SHOC / COMPAS user guide.

For developers considering adding new ecological processes, the original description of the ecological module (The “new” ecological model: user guide by Pavel Sakov) is available on the documentation page.

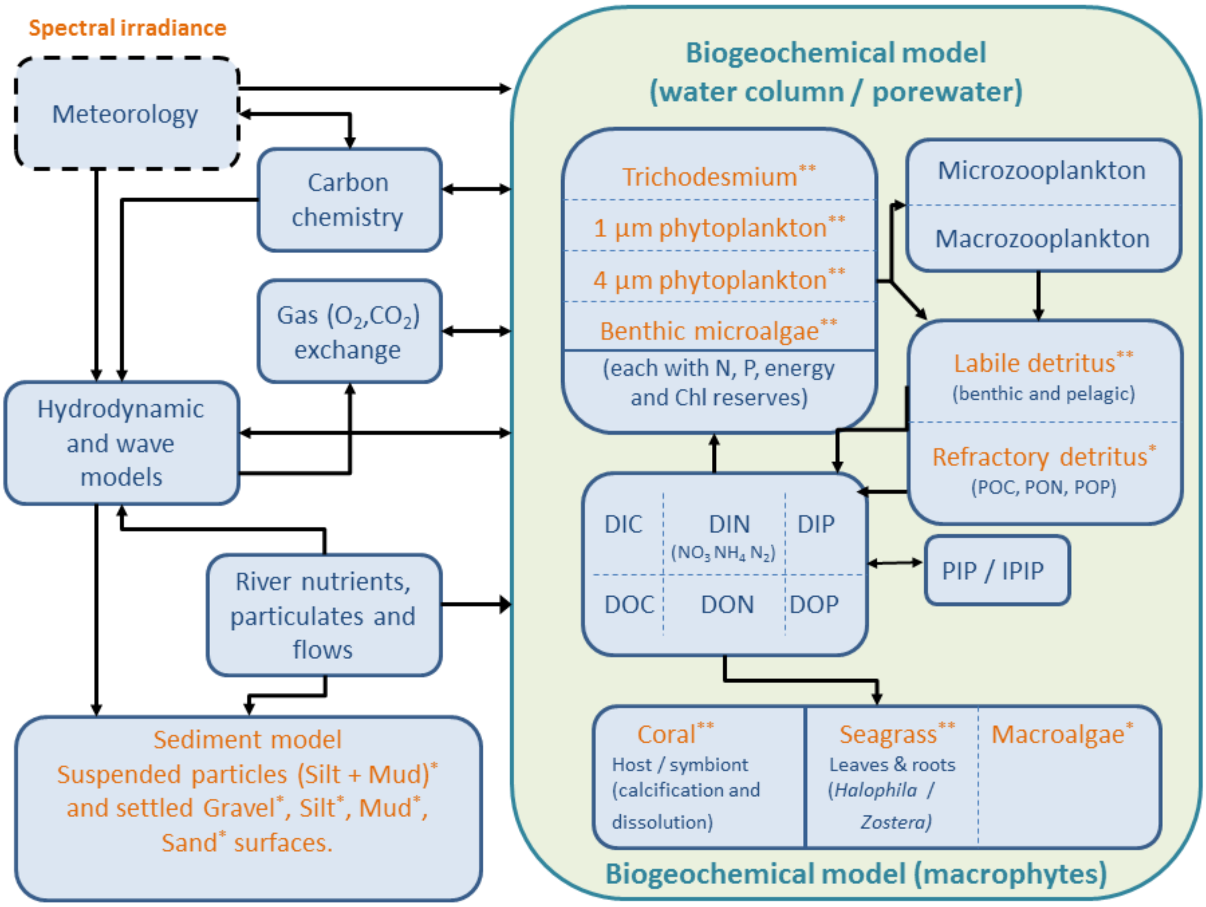


Figure 1 The EMS model suite used for the eReefs project.

2 Code availability

The source code, and conditions of use, is available from a GitHub archive:

https://github.com/csiro-coasts/EMS.

Model documentation (including the latest version of this document) are available at:

https://research.csiro.au/cem/software/ems/ems-documentation/

3 Model configuration files

EMS is run from a compiled executable (see Section 9) that points to a number of configuration files, all of which are text files. The "Master" configuration is called in the execution line, and contains the names of any other files that are called.

For convenience, instead of using the filename suffix **.txt**, we use the following (noting they are still text files):

- .prm files are text files containing the model parameters.
- .tran files are text files used to run the off line transport model.
- .pss files are text files that list the location of the tracer point source.
- .ts files are text which contains time series of data of model input or outputs.
- .mnc files are text file listing source of multiple netcdf files.

Within these files any line starting with # is a comment line.

3.1 Master configuration file

This is the main file to run the model it contains:

- model run time
- model output and restart files
- hydrodynamic flags and parameters
- grid & bathymetry definition
- model tracer list definition
- model forcing
- model time series outputs
- model boundaries
- vector of bathymetry
- x grid
- y grid

3.1.1 Tracer list

The tracer list contains all the prognostic and diagnostic tracers that are available for output from the model. Most of the flags are self-explanatory, but the "diagn" is subtle.

A prognostic tracer has a tracer flag TRACER.diagn of 0, while a diagnostic tracer can take a value of 1 or 2. Diagnostic variables not altered by advection, diffusion or sediment processes.

In the ECOLOGY model, if diagn = 1 (flux diagnostic), then the diagnostic is the mean flux for the ecological time period (obtained by taking the change in the tracer from the initial value to the end and dividing by the timestep.). If diagn = 2 (mass diagnostic), then the diagnostic is the value of the tracer at the end of the time period.

In the rest of the code (hydro, sediments) diagn 1 and 2 simply the value at the output time, but are not advected of diffused. Therefore tracers which are never given an increment by the ecology, such as TSS, the use of diagn = 1 by the sediment model is not a problem.

Example prognostic hydrodynamic tracer:

```
TRACER0.name                 salt   # SHOC variable name.
TRACER0.long_name            Salinity  # long name in netcdf
TRACER0.units                PSU
TRACER0.type                 WATER SEDIMENT
TRACER0.fill_value           35.0 # default value initial value.
TRACER0.valid_range          0.0 40.0  # gives warning if outside range
TRACER0.advect               1        # this tracer is advected
TRACER0.diffuse              1        # this tracer is diffused
TRACER0.diagn                0        # this tracer is not a diagnostic
tracer
TRACER0.data           data/temp.nc       # if present, overwrite the
                       fill_value, the data in temp.nc are interpolated onto
                       the model grid (see SHOC user guide for specification
                       of temp.nc)
```

Example prognostic dissolved BGC tracer:

```
TRACER23.name DIP
TRACER23.long_name Dissolved Inorganic Phosphorus
TRACER23.units mg P m-3
TRACER23.valid_range_wc 0 1e+35
TRACER23.valid_range_sed 0 1e+35
TRACER23.diagn 0
TRACER23.type         WATER SEDIMENT
TRACER23.fill_value_wc 0.5
TRACER23.fill_value_sed 100 # fill value in the sediment, if present,
overwrite the value_wc in the sediment
TRACER23.inwc 1
TRACER23.insed 1
TRACER23.dissol 1  # dissolved tracer
TRACER23.partic 0
TRACER23.advect 1
TRACER23.diffuse 1
TRACER23.decay 0
TRACER23.svel 0   # sinking rate
TRACER23.resusp_rate 0
```

Example prognostic particulate BGC tracer:

```
TRACER33.name DetR_P
TRACER33.long_name Refractory Detrital Phosphorus
TRACER33.units mg P m-3
TRACER33.valid_range_wc 0 1e+35
TRACER33.valid_range_sed 0 1e+35
```

```
TRACER33.diagn 0
TRACER33.type         WATER SEDIMENT
TRACER33.fill_value_wc 0.01
TRACER33.fill_value_sed 1.1e+03
TRACER33.inwc 1
TRACER33.insed 1
TRACER33.dissol 0
TRACER33.partic 1  # particulate tracer altered by sediment code
TRACER33.advect 1
TRACER33.diffuse 1
TRACER33.decay 0
```

# the following parameter are used in the sediment code to do sinking and resuspension

```
TRACER33.psize 1e-05  # size
TRACER33.b_dens 1e+09  # density
TRACER33.i_conc 2e+08
TRACER33.f_conc 2e+08
TRACER33.svel -2.31e-05
TRACER33.crit_stress 0.2
TRACER33.resusp_rate 0
```

TRACER0.TYPE #which type of variable:

```
WATER    # 3D water column only
SEDIMENT # 3D sediment only
WATER SEDIMENT# 3D water column and SEDIMENT
WC2D# 2D water column
BENTHIC# 2D BENTHIC LAYER (sees both the last layer of the water column and
top layer of the sediment)
```

An option available is the automatic generation of the ecological tracers in the tracer list based on the choice of processes. This development is the key to ecological implementations of RECOM, as the user only specifies the process list and parameter values, not the tracers. For this option, see standard process list (Section 3.2)

3.1.2 Tracer statistics

To calculate a diagnostic tracer from other tracer output, or obtain a time-mean of the tracer of the, the tracerstat option can be used:

```
      TRACER50.name Chl_a_sum
      TRACER50.long_name Chlorophyll_sum
      TRACER50.units mg Chla m-3
      TRACER50.type         WATER SEDIMENT
      TRACER50.valid_range_wc 0 1e+35
      TRACER50.valid_range_sed 0 1e+35
      TRACER50.diagn 2
```

```
TRACER50.tracerstat sum(PhyL_Chl,PhyS_Chl,MPB_Chl,Tricho_Chl)
TRACER50.fill_value_wc 0
TRACER50.fill_value_sed 0
```

The time-means are especially useful when calculating a daily mean of a variable that changes cyclically through the day, such as chlorophyll concentration.

3.2 The process list

EMS contains many processes (i.e. phytoplankton growth, gas exchange etc.) that have been develop for projects around Australia and in Chile. To understand how these processes work the reader is encouraged to look at the top of the *.c process file. This section discusses how to implement a sub-set of the EMS processes in a particular model configuration.

The process list file contains the listing of all the BGC processes that are to be run in the model, broken into the 3 vertical zones: water column, epibenthic and sediment. Each of these processes has 2 associated files (a .h and .c file), located in the source code folder at:

```
https://github.com/csiro-
coasts/EMS/tree/master/model/lib/ecology/process_library
```

The process list use is invoked in the configuration file with the statement:

```
PROCESSFNAME     process_list.prm
```

All processes in the source codes (listed in allprocessess.c) are included in the model executable when it is compiled. But processes are only implemented in the model run if they are in the process list. This allows the same executable to be used for multiple applications. Some processes have arguments. In this case, the comments at the top of the *.c file explain the options.

In general, processes with a `_wc, _sed, or _epi` suffix can only be applied in the water column, sediment or epibenthic respectively.

The processes `values_common` and `values_common_epi` are used define variables either in pre-calc, such that they can be used in the other processes, or in post-calc, so they can be outputted. Examples of variables defined include DIN.

Occasionally conflicts can occur with combinations of processes, and the user needs to be careful. For example, if a process is listed twice, the code will attempt to execute the rate process twice. Also, where two alternate versions of a process exists, such as `trichodesmium_grow_wc` and `trichodesmium_spectral_grow_wc`, applying both will have unintended consequences.

Furthermore, a peculiarity of the code is that it checks for the mass balance of model through adding state variable to the sums TN, TP, TC and TO. It does these at the beginning and end of the process. It uses the fact that it is in a particular process to know to add a variable to this sum. So for example, if it is in the process `trichodesmium_spectral_grow_wc,` it adds the nitrogen associated with the trichodesmium structural material and the reserves of nitrogen to TN. However, if `trichodesmium_spectral_grow_wc` is not specified, but there is trichodesmium in the tracer list, and `zooplankton_large_spectral_grow_wc` is called, then the mass balance will fail because TN does not include structural material of trichodesmium, and it is altered by grazing. Usually these conflicts are due to genuine incompatibility of processes. But occasionally it may be necessary to turn off the mass balance (see Section 10.1).

The configuration used in this example contains the following processes:

- **in the water column :**

```
tfactor       # Q10 temperature dependence of some parameters
viscosity     # calculate the viscosity from T and S
moldiff       # calculate the molecular diffusivity from T and S
remineralization  # contain all the remineralization processes.
microphytobenthos_spectral_grow_wc
phytoplankton_spectral_grow_wc(small)
phytoplankton_spectral_grow_wc(large)
trichodesmium_mortality_wc
trichodesmium_grow_wc
phytoplankton_spectral_mortality_wc(small)
phytoplankton_spectral_mortality_wc(large)
zooplankton_mortality_wc(small)
zooplankton_mortality_wc(large)
zooplankton_large_spectral_grow_wc
zooplankton_small_spectral_grow_wc
nitrification_wc
p_adsorption_wc
carbon_chemistry_wc  # OCMIP routines
gas_exchange_wc(oxygen,carbon) # if just one, (gas,dummy)
carbon_chemistry_wc
values_common   # calculate value across the different processes
massbalance_wc
light_spectral_wc
age_wc # perform age tracer experiment from a source
```

Notes:

1. phytoplankton_spectral_grow_wc, phytoplankton_spectral_mortality_wc and zooplankton_mortality_wc contain an argument of either (small) or (large). For these three processes the code is identical, but the parameters vary between small and large.
2. gas_exchange_wc(oxygen,carbon) – Gas exchange is calculated once, and applied to the gases in the argument. If carbon specified, carbon_chemistry must be called above.
3. `Trichodesmium` processes can only be used with the spectrally-resolved processes.
4. Mass balance processes– this is a test within a cell and an ecological time step that the model conserves mass. It’s most appropriate use is to ensure the code is consistent. It does not assess whether the model losses mass across boundaries or during advection or diffusion.

- **in the epibenthic layer (~ benthic layer):**

```
tfactor_epi
macroalgae_spectral_grow_epi
seagrass_spectral_grow_epi(Zostera)
coral_spectral_grow_epi
macroalgae_mortality_epi
seagrass_spectral_mortality_epi(Zostera)
values_common_epi
massbalance_epi
light_spectral_epi
diffusion_epi
```

Notes.

1. `seagrass_spectral_grow_epi` can be called with “Zostera” or “Halophila”. The equations are the same, but “Zostera” uses SG and SGROOT tracers and parameters starting with SG, while “Halophila” uses SGH and SGHROOT tracers and SGH parameters.
2. `diffusion_epi` calculates a diffusive exchange between the sediment and water column based on a constant diffusion coefficient specified by parameters EpiDiffCoeff and EpiDiffDz. It is in additional to the diffusive flux specified in the sediment model by

- **in the sediment layers :**

```
tfactor
viscosity
moldiff
remineralization
light_sed
microphytobenthos_spectral_grow_sed
microphytobenthos_spectral_mortality_sed
phytoplankton_spectral_mortality_sed(small)
phytoplankton_spectral_mortality_sed(large)
trichodesmium_mortality_sed
nitrification_denitrification_sed
p_adsorption_sed
values_common
massbalance_sed
```

3.2.1 Standard process lists

Instead of specifying a list of processes from a file as described above, a number of key words are reserved for standard process lists. For example,

```
PROCESSFNAME     BCG2p0
```

invokes a standard list of processes. When a standard process list is used, the model also generates the tracers that is required by the processes in this list. The list it uses is also written into the execution folder with a filename `processes_*.prm`, where * is the key word. A list of the most commonly used lists is given below (Table 1).

Table 1 Standard process lists

Keyword	Description of process list
BGC2p0	Biogeochemical model used in the eReefs implementation B2p0 with 4 microalgae, 2 zooplankton, 3 seagrass, 1 macroalgae and a simple coral host symbiont model.
BGC3p1	Biogeochemical model used in the eReefs implementation B3p1 with 4 microalgae, 2 zooplankton, 3 seagrass, 1 macroalgae and the coral host symbiont model with coral bleaching.
gas_only	Air-sea gas exchange of carbon and oxygen only.
optics_only	The spectrally-resolved optical model calculations only.
dead_only	All processes in BGC3p1, except those involving living processes, and also including anammox. This includes the spectrally-resolved optical model.

Most of the standard process lists are available through the RECOM interface.

3.3 The parameter file

This file contains a list of the BGC model parameters and their metadata.

```
NPARAMETERS      141

PARAMETER0.name    Tref
PARAMETER0.desc    Reference temperature
PARAMETER0.units   Deg C
PARAMETER0.value   20.000000
PARAMETER0.stderr  0.000000
PARAMETER0.ref     Set by parameter library options.
```

Notes:

1. EMS does read the “units”, and if a time unit other than “seconds” is specified, it will convert to seconds for use in the code.
2. As a matter of principle, all biological parameters are read into the code during the initialisation phase.
3. The top of the bio_spectral.prm file contains a line: “NPARAMETERS 98” which must be manually changed to the number of parameters (remember the count starts at 0).
4. Just because a variable appears in the bio.prm files does not imply it is used in the code. If it is never called by any processes, or a typo exists in the “name”, the code will not use it. Often this results in a default value in the code being used. To ensure the value of the parameter that is implemented is known, **ecology_setup.txt** prints out each time the code uses a parameter value. The self-explanatory statements, are one of the following three:

```
Ecol. parameter forced read, <NAME> = XXX
Ecol. parameter tried to read <NAME> and found XXX
Ecol. parameter tried to read <NAME>, but not in parameter file.
Code default of <NAME>, XXX
```

3.3.1 Standard parameter file

Instead of specifying parameter values from a file as described above, a number of key words are reserved for a set of standard parameter values. For example,

```
BIOFNAME   BCG2p0
```

invokes a standard set of parameter values used in the B2p0 configuration. The list used is also written into the execution folder with a filename `bio_*.prm`, where * is the key word. A list of the most commonly used lists is given below (Table 1).

Table 2 Standard set of parameter values

Keyword	Description of set of parameter values
BGC2p0	Biogeochemical model used in the eReefs implementation B2p0 with 4 microalgae, 2 zooplankton, 3 seagrass, 1 macroalgae and a simple coral host symbiont model.
BGC3p1	Biogeochemical model used in the eReefs implementation B3p1 with 4 microalgae, 2 zooplankton, 3 seagrass, 1 macroalgae and the coral host symbiont model with coral bleaching.

4 Sediment – water column exchange rates

The exchange of dissolved tracers between the water column and sediment occurs because of:

1. Deposition of sediments, entraining the porewater from the bottom layer of the water column (a flux into the sediment).
2. Erosion of sediments, releasing porewater into the water column (a flux out of the sediment).
3. Diffusion in the sediment model (which is slow).
4. Diffusion in the ecology model (implemented in the process `diffusion_epi`).

The sum of processes 1-4 is difficult to calculate from the state variables in the model output, so inline diagnostic calculations are available.

The sediment model calculates the sum of the instantaneous rates of 1-3 of variable `XXX` if the tracer `XXX_fluxsedi_inst` is specified :

```
TRACER196.name              DIC_fluxsedi_inst
TRACER196.long_name         Sed-water sed. DIC flux
TRACER196.units             mg C m-2 s-1
TRACER196.fill_value        0.00e+00
TRACER196.valid_range       -1.00e+35 1.00e+35
TRACER196.diagn             0
TRACER196.advect            0
TRACER196.diffuse           0
TRACER196.type              WC2D DIAGNOSTIC
```

The ecological model calculates diffusion in the ecological model if the process `diffusion_epi` is evoked, and the diagnostic tracer `XXX_sedflux` is specified.

```
TRACER195.name              DIC_sedflux
TRACER195.long_name         Sed-water ecol. DIC flux
TRACER195.units             mg C m-2 s-1
TRACER195.fill_value        0.00e+00
TRACER195.valid_range       0.00e+00 1.00e+35
TRACER195.diagn             2
TRACER195.type              WC2D  ECOLOGY  DIAGNOSTIC
```

At present the only variables available for `XXX_sedflux` are `DIC, DIP, NO3, NH4, alk,` and `Oxygen`. Unless a large resuspension or deposition event has occurred, `XXX_sedflux` will generally be an order of magnitude more than `XXX_fluxsedi_inst.` The total instantaneous sediment-water exchange (positive in the sediment to water direction) is the sum of the `XXX_sedflux` and `XXX_fluxsedi_inst`.

5 Optical model configuration

The spectrally-resolved optical model calculates the inherent optical properties (IOPs) from the optically-active state variables, and then uses the IOPs to calculate the apparent optical properties such as the light field and remote-sensing reflectance.

The optical model is configured using parameter variables specified in **bio.prm**, through option calls to light processes in the process list models, and through tracer list specification. The relationships used to relate state variables to IOPs (such as the pigment-specific absorption coefficient of each of the pigments) are specified in the file `bio_opt.c`:

`https://github.com/csiro-coasts/EMS/blob/master/model/lib/ecology/bio_opt.c.`

The wavelengths (centre and bandwidths) used are specified in the parameter file using:

```
PARAMETER4.name    Light_lambda
PARAMETER4.desc    Wavelengths of light
PARAMETER4.units   nm
PARAMETER4.value   290.000000 310.000000 330.000000 350.000000 370.000000
390.000000 410.000000 430.000000 440.000000 450.000000 470.000000 490.000000
510.000000 530.000000 550.000000 570.000000 590.000000 610.000000 630.000000
650.000000 670.000000 690.000000 710.000000 800.000000
PARAMETER4.stderr  0.000000
PARAMETER4.ref     Approx. 20 nm resolution with 10 nm about 440 nm. PAR
(400-700) is integral of bands 2-22.
```

The 'hard' coded shortest wavelength edge is 140 nm, and the longest edge at 8000 nm. The spectral 'grid' that results from this specification is output into **ecology_setup.txt**:

```
Centre of waveband (edge):
290.00 (140.00 - 300.00)
310.00 (300.00 - 320.00)
330.00 (320.00 - 340.00)
350.00 (340.00 - 360.00)
370.00 (360.00 - 380.00)
390.00 (380.00 - 400.00)
410.00 (400.00 - 420.00)
430.00 (420.00 - 435.00)
440.00 (435.00 - 445.00)
450.00 (445.00 - 460.00)
470.00 (460.00 - 480.00)
490.00 (480.00 - 500.00)
510.00 (500.00 - 520.00)
530.00 (520.00 - 540.00)
550.00 (540.00 - 560.00)
570.00 (560.00 - 580.00)
590.00 (580.00 - 600.00)
610.00 (600.00 - 620.00)
630.00 (620.00 - 640.00)
650.00 (640.00 - 660.00)
670.00 (660.00 - 680.00)
690.00 (680.00 - 700.00)
710.00 (700.00 - 755.00)
800.00 (755.00 - 8000.00)
```

For this particular optical grid, and the clear sky spectrally-resolved solar radiation, approximately 40 % of the energy of the sun is between 400 and 700 nm. Note that in this configuration we have used a 20 nm resolution, but added an extra wavelength at 440 nm to better resolve absorption by photosynthetic pigments.

In the tracer list, the inclusion of tracers with tracer names of R_*** results in the optical model calculating remote-sensing reflectance, R_{rs}, at wavelength *** nm:

```
TRACER93.name              R_640
TRACER93.long_name         Remote-sensing reflectance @ 640 nm
TRACER93.units             sr-1
TRACER93.fill_value        0.00e+00
TRACER93.valid_range       0.00e+00 1.00e+35
TRACER93.diagn             2
TRACER93.type              WC2D  ECOLOGY  DIAGNOSTIC
```

Further, a number of simulated satellite products can be outputted if the required remote-sensing reflectance at the required wavelengths, and the key tracer name are specified (with similar options to the Remote-sensing reflectance above):

Table 3 Simulated observations

Tracer name	Description of simulated variable	wavelength (nm)	Source
OC3M	Three-band chlorophyll algorithm calibrated for the MODIS sensor.	443, 488, 547	
OC3V	Three-band chlorophyll algorithm calibrated for the VIIRS sensor.	443, 486, 551	
OC4Me	Four-band chlorophyll algorithm calibrated for the MERIS and Sentinel3A sensors	443, 488, 510, 560	
Secchi	Secchi depth calculated from vertical integral of attenuation of light at 488 nm.	488	Lee et. al (2016).
TSSM	Turbidity based on red reflectance – local GBR algorithm.	645	Petus
KD490M	Two band vertical attenuation at 490 nm.	488, 547	
nFLH	Normalized Fluorescence Line Height based on MODIS wavelengths.	667, 678, 748	
Turbidity	Turbidity calculated from total scattering at 590 nm.	590	
Fluorescence	Chlorophyll concentration 'scaled' by processes that reduce fluorescence such as self-shading and photochemical quenching.	Integral of photosynthetic wavelengths.	

Finally, the default calculation of the zenith angle is on the time zone (in below example +10) specified at the top of the master configuration file:

```
TIMEUNIT         seconds since 1990-01-01 00:00:00 +10
```

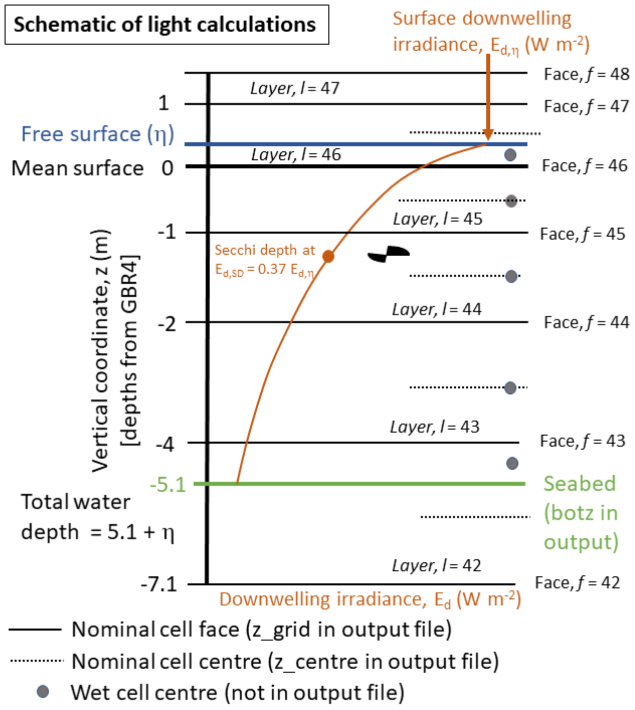
However, if the option:

```
PROJECTION       geographic
```

is invoked, the zenith angle is calculated based on the longitude (as well as the hour, day, and latitude).

5.1 Outputs of optical variables

The output of optical variables is complicated by the calculation of downwelling irradiance (PAR_z) at the cell faces, but PAR at the cell centres. Further, cell centres of the partially-filled surface and bottom layers deviate from the nominal cell centre (z_centre in output files). The schematic below (Figure 2) illustrates these complications and gives equations that can be calculated from the output. None of these complications affect the output of 2D variables (such as remote-sensing reflectance) which are calculated on the moving free surface at the output time.



Schematic of light calculations
Surface downwelling irradiance, $E_{d,\eta}$ (W m^{-2})
Layer, l = 47
Face, f = 48
Face, f = 47
1
Free surface (η)
Layer, l = 46
Mean surface 0
Face, f = 46
Layer, l = 45
-1
Face, f = 45
Secchi depth at $E_{d,SD}$ = 0.37 $E_{d,\eta}$
Vertical coordinate, z (m) [depths from GBR4]
-2
Layer, l = 44
Face, f = 44
-4
Layer, l = 43
Face, f = 43
-5.1
Seabed (botz in output)
Total water depth = 5.1 + η
Layer, l = 42
-7.1
Face, f = 42
Downwelling irradiance, E_d (W m^{-2})
Nominal cell face (z_grid in output file)
Nominal cell centre (z_centre in output file)
● Wet cell centre (not in output file)

Quantities in model output (name in RED)

Downwelling irradiance (PAR_z) at cell face *f-1 is* calculated from the light at the face above and considering attenuation by constituents in layer *l* = *f-1* at a rate $K_{d,l}$:

$$PAR\_z(f-1) = E_{d,f-1} = E_f e^{-K_{d,l}(z(f)-z(f-1))}$$

PAR_z is outputted for each layer, and is the value at the *top* of each layer. EpiPAR_sg, which is a 2D variable, is PAR_z when the bottom face is the seabed.

Volume-averaged downwelling irradiance for layer *l* (PAR):

$$PAR = E_{d,l} = \frac{E_f\left(1 - e^{-K_{d,l}(z(f)-z(f-1))}\right)}{K_{d,l}\,(z(f) - z(f-1))}$$

For lowest cell face, *f*, at which at 488 nm, $E_{d,488} < 0.37\ E_{\eta,488}$, Secchi depth (Secchi):

$$Secchi = -z(f+1) + \log\left(\frac{\exp(-1)\,/E_{f+1,488}}{-K_{d,l,488}}\right)$$

Quantities that can be calculated from model output

The downwelling irradiance at depth z′, found within layer *l*, is given by: $PAR_{z'} = PAR\_z(l)\ e^{-K_{d,l}(z(l)-z')}$

Volume-averaged scalar irradiance for layer *l* (not outputted), with total absorption a_T, used in photosynthesis calculations:

$$E_{s,l} = E_{d,l}\,K_{d,l}/a_T$$

Figure 2 Schematic illustrating the vertical grid of the model and the locations on this grid that the optical properties have been calculated.

5.2 Moonlight

The optical model includes the calculation of moonlight parameters using orbital calculations in the hydrodynamic model, and the USGS ROLO spectral irradiance model (https://astrogeology.usgs.gov/moon-cal/modeling/irradiance.php , Kieffer, H. H and T. C. Stone (2005) The spectral irradiance of the moon. The Ast. J. 129: 2887-2901).

In order to invoke the model, the third argument to the process `light_spectral_wc` must be ‘M’. In order for moonlight calculations to be undertaken, `PROJECTION` must be `geographic`.

The output variables for moonlight are:

Moonlight	Moonlight (downwelling PAR-integrated irradiance)
Lunar_zenith	Moon zenith (0 overhead; π/2 at horizon)
Lunar_phase	Moon phase (0 new moon; π/4 first quarter; π/2 full moon; 3π/4 last quarter; 0 new moon)
Moon_fulldisk	Moonlight brightness (PAR-integrated) if looked at directly.

The full disk reflectance includes the effect of moon phase on irradiance, but the zenith angle.

The USGS ROLO model include the effect of Earth-Sun distance, Earth Moon distance, spatially varying albedo of the moon (i.e. the craters are darker than the plains), the point of the Moon facing the Earth, phase of the moon, time of day, point on the Earth (latitude).

6 Carbon chemistry model

To implement the carbon chemistry model requires that the tracers temperature (`temp)`, salinity (`salt)`, total alkalinity (`alk)`, dissolved inorganic carbon (`DIC`) and pH (`PH)` be specified. Additional diagnostic variables that can be produced by the carbon chemistry model are calcite saturation, aragonite saturation, surface partial pressure of carbon dioxide, bicarbonate ion concentration, carbonate ion concentration and atmospheric CO_2 concentration. The atmospheric CO_2 concentration can either be set to a constant using the parameter file:

```
PARAMETER3.name    xco2_in_air
PARAMETER3.desc    Atmospheric CO2
PARAMETER3.units   ppmv
PARAMETER3.value   396.480000
PARAMETER3.stderr  0.000000
PARAMETER3.ref     Mean ‘13,Mauna Loa: http://co2now.org/current-co2/co2-now/
```

Or by using a resetting spatially-resolved tracer:

```
TRACER77.name            xco2_in_air
TRACER77.long_name       Atmospheric CO2
TRACER77.units           ppmv
#TRACER77.fill_value      400.0
TRACER77.valid_range     0.00e+00 1.00e+35
TRACER77.advect          0
TRACER77.diffuse         0
TRACER77.diagn           0
TRACER77.partic          0
TRACER77.dissol          0
TRACER77.type             WC3D   SEDIMENT ECOLOGY  PROGNOSTIC
TRACER77.eco_flag            0
TRACER77.reset_file  forcing_bgc/global_co2_merged_2016.ts
TRACER77.reset_dt 30 days
```

The Ocean-Carbon Cycle Model Intercomparison Project (OCMIP) numerical methods are used to quantify air-sea carbon fluxes and the carbon dioxide system equilibria in seawater. The OCMIP procedures quantify the state of the CO_2 system using two prognostic variables, the concentration of dissolved inorganic carbon, C_T, and total alkalinity, A_T. The value of these prognostic variables, along with salinity and temperature, are used to calculate the partial pressure of carbon dioxide, $p\mathrm{CO_2}$, in the surface waters using a set of governing chemical equations that are solved using a Newton-Raphson method with literature-derived constants. One alteration from the typical implementation of the OCMIP algorithm is that we increased the search space for the iterative scheme from ±0.5 pH units (appropriate for global models) to ±2.5. The air-sea flux of carbon dioxide is calculated based on wind speed data (requiring process `gas_exchange_wc(carbon)`), and the air-sea gradient in $p\mathrm{CO_2}$, using an empirical net flux relationship.

7 Ecological integration options

In the model configuration file .prm or .tran file: ECOLOGY options are:

```
DO_ECOLOGY 1      ## DO ECOLOGY FLAG SET TO 1 TO run ecology
ECOLOGY_DT 1200   # ECOLOGY time step
```

Note: make sure that the ecology timestep is the same length or less than the shortest timestep in the physics or forcing data. If this is not true, the model will change `ECOLOGY_DT` from what is specified in the file, which will be noted in the **runlog.**

Specify the other configuration files:

```
biofname          in/bio_spectral.prm  # file that contain model parameters
processfname      in/processes_spectral.prm # file that contains the list of
                  which processes.
```

Specify integration checks:

```
check_nans 0      # test to check is there is any NaN in the state variable
array , computationally expensive so only used during debugging.
```

```
check_negs 0 # test to check is there is any <0 in the model , expensive so
only used during debugging.
```

Both `check_nans` and `check_negs` cause the model to stop, and output diagnostic information to the **runlog**. `check_negs` is will often fail with dissolved oxygen, that is allowed to go slightly negative.

Miscellaneous options:

```
internal_tracers 1  # legacy
mandatory_water  0 # legacy
mandatory_sediment 0 # legacy
```

Light forcings:

The short wave solar radiation (i.e up to about 5000 nm, or which approximately 40% is PAR, see Section 4) can be specified from a file:

```
LIGHT             in/light.ts    # get the light from a .ts text file
LIGHT_INPUT_DT          1 days
ALBEDO_LIGHT            0.2# legacy
```

Or from an internal tracer:

```
LIGHT             swr     # get the light from the internal swr hydro tracer
(form meteorological forcing)
```

If it is coming from an internal tracer, the tracer can either be from the fully-coupled model, or in the transport model. For convenience, if the following diagnostic tracer is specified:

```
TRACER120.name              PAR_z
TRACER120.long_name         Downwelling PAR at top of layer
TRACER120.units             mol photon m-2 s-1
```

```
TRACER120.fill_value      0.00e+00
TRACER120.valid_range     0.00e+00 1.00e+35
TRACER120.fill_value_sed  0.00e+00
TRACER120.valid_range_sed 0.00e+00 1.00e+35
TRACER120.advect          0
TRACER120.diffuse         0
TRACER120.diagn           2
TRACER120.partic          0
TRACER120.dissol          0
TRACER120.type             WC3D  SED  ECOLOGY  DIAGNOSTIC
TRACER120.psize           0.00e+00
TRACER120.b_dens          0.00e+00
TRACER120.i_conc          0.00e+00
TRACER120.svel            0.0
TRACER120.cohesive        1
TRACER120.calcvol         0
TRACER120.floc            0
TRACER120.resuspend       1
TRACER120.deposit         1
TRACER120.eco_flag            0
```

then the surface value of PAR_z will be the surface PAR radiation (see Section 4).

Numerical scheme for ecology integration

The model uses an adaptive time step to solve the ecological ordinary differential equations (ODEs). The model configuration file allows specification of the adaptive integrator used, and the precision to which it is applied.

```
integrator dopri5            #  BGC integrator scheme
integration_precision 1e-5   # BGC integrator precision
```

where the above options are the default. Integrator schemes include (`dopri8,dopri5,euler1`). The adaptive schemes use an approximation of the solution through the time-step (for `dopri5`, four and five time-steps), and use the difference in solution at the end to determine if the precision requirement has been meet. Only the solution at the end of the time step is kept.

By-passing ecological integration

The ecological integration is not undertaken if:

- The cell is not wet. Note whether a cell is ‘wet’ or ‘dry’ can change in time due to surface elevation changes.
- If the adaptive integrator fails to converge after NMAX = 2000 steps (this is hardcoded), the ODEs for that timestep are set to zero, the tracer ECOLOGY_ERROR is incremented. In this case the updating excludes the whole column (wc, epibenthic and sed).
- If the master configuration file lists the cell below “EXCLUDE_BGCSED”. For example:

  ```
  EXCLUDE_BGCSED 2
  (169,15)-(170,15)
  7 11
  ```

8 Pre-processing tools for BGC

[TO BE COMPLETED]

9 Running simulations

Simulations can be run either fully coupled to one of the two hydrodynamic models available in the Environmental Modelling Suite (SHOC – Sparse Hydrodynamic Ocean Code; COMPAS – Coastal Ocean Marine Prediction Across Scales), or in an off-line mode where the current, temperature and salinity fields from an earlier hydrodynamic run are used to force the BGC model.

9.1 Fully-coupled mode

By convention, model configuration files that are run in the fully-coupled mode have endings `.prm`.

To execute the model in the fully coupled SHOC model:

```
./shoc -p test_estuary_spectral.prm
```

To execute the model in the fully coupled COMPAS model:

```
./compas -p test_estuary_spectral.prm
```

9.2 Off-line transport mode

There are two transport models available, one in SHOC and one in COMPAS.

This is just a quick description of the off-line transport model, more details are available in the SHOC user guide.

The following files has been saved when running the hydrodynamic model:

```
file8.name              out/test_estuary_tran.nc
file8.filetype          sparse
file8.tinc              1 hour
file8.bytespervalue     4
file8.vars              eta salt temp u1mean u2mean wmean Kzmean swr
```

The transport model then reads test_estuary_tran.nc and runs the BGC and sediment in an off line version. This file need to be copied across to the trans folder (so it does not get deleted).

Instead of using a test_estuary_spectral.prm as a main parameter file, we use a stripped version called test_estuary_spectral.tran (grid and model forcing are removed only replaced by transport model options, all the ecology tracer list sediment ocean boundary stay the same, expect the flag advection and diffusion for temp and salt need to be set to 0 as they are read from files).

If the model source code has been configured with the `--enable-omp` option during configuration before compiling (see SHOC user guide):

```
./conf/configure --enable-omp
```

Then the BGC model can be run using more than 1 CPU using the following option specified in the `test_estuary_spectral.tran`

```
eco_omp_num_threads 12  # 12 cpus in this case
```

To run the transport model in the two models:

```
./shoc –t trans/test_estuary_spectral.tran
./compas –t trans/test_estuary_spectral.tran
```

10 Computational diagnostics

The model outputs a few files during execution that are informative. Files written during the initialization are:

1. **setup.txt** – describes the model configuration, including tracers, boundary conditions etc. This is from the transport model.
2. **sedlog.txt** – outputs at the beginning the sediment configuration, plus any errors in the sediment model (requires VERBOSE_SED to be activated in the in.tran or in.prm file).
3. **ecology_setup.txt** – describes the optical model characteristics, and lists the biological parameters read and used, as well as the processes. Arguments given in the process list, such as gases calculated in gas_exchange_wc etc. are also listed.

Files written during the initialization and model integration:

4. **diag.txt** – this file is written to each ecological time step, and updates computing performance and expected finish time.
5. **runlog** – outputs during runtime warnings and errors.

The files **ecology_setup.txt** and **setup.txt** are automatically copied to the OUTPUT folder at the completion of the model initialization.

10.1 Mass balance

Like all biogeochemical models, maintaining mass balance is an important requirement of the biogeochemical model. The **Biogeochemical Science Guide** details the sink / source equations that are conserved. The model code contains checks for the conservation of carbon, nitrogen, phosphorus and oxygen. This check involves calculating the mass before and after each ecological step. The difference between the two is calculated. If the difference exceeds a relative tolerance (which is printed in **ecology_setup.txt**, and is calculated as the mass difference over the original value), then the model prints an error to the **runlog**. The hardcoded value for the relative tolerance is 10^{-9} for the water column and sediment, and 10^{-6} for the epibenthic. The exception to the above calculation is oxygen, for which the mass difference is divided by the minimum of the original mass and 8000 mg m^{-3}.

In order to implement the mass balance the process list must contain `massbalance_wc`, `massbalance_epi` and `massbalance_sed` in the process list.

To determine mass budgets, including external inputs, circulation etc. see the SHOC manual region files.

11 RECOM / test case

[TO BE COMPLETED]

age.ts : time series file for tracer age concentration at the river boundary (set to 0 in this example)

dicriver.ts : time series file for tracer DIC and alkalinity concentration at the river boundary

gbr_gravel_600x180.txt : text file with gravel data to initialize the model

input.nc : initial condition file created using ./shoc –g prm/bgc/prm data/input.nc

met.ts : time series file for meteorological forcing (time varying but spatially constant across the model domain)

passive0.ts : point source data file that contain NO3 and passive tracer fluxes need to be in

data.txt : text file with geographic short wave radiation bottom absorption ~ spatially variable parameter.

flow.ts : time series file with river flow temperature and salinity condition at the river boundary.

gbr_mud_600x180.txt : text file with mud data to initialize the model

nutrient_loads.ts : time series file with nutrient and sediment concentration at the river boundary (flow is also present but not used in fully coupled, only when used in transport mode).

dicocean.ts : time series file for tracer DIC and alk concentration at the river boundary

forcing.ts : time series file for meteorological forcing (time varying but spatially constant across the model domain)

gbr_sand_600x180.txt : text file with sand data to initialize the model

met2.ts : time series file for meteorological forcing (time varying but spatially constant across the model domain)