This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 869383

Mechanistic, trait-based model for unicellular plankton

Work package: **WP3**

Deliverable No. (D3.1) – Title**: Mechanistic, trait-based model for unicellular plankton**

Lead Beneficiary (WPX): DTU Aqua

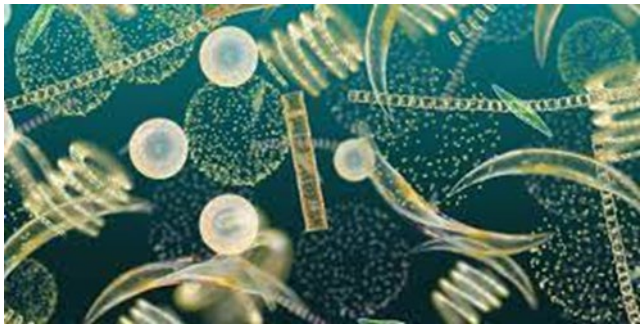
Lead Scientist responsible for the report (Name, Institution): Andre W. Visser (DTU Aqua)

Editors (Name, Institution): Anna Törnroos-Remes (ABO)

Contributors (Name, Institution):

Submission Date: 31/05/2021

Table of Contents



1 Executive Summary

This deliverable supplies a demonstration of the unicellular component (essentially phytoplankton, bacteria and micro zooplankton) of a generalized trait-based plankton model, and is a direct output from Task 3.1 *Generalized trait-based model of plankton*. It outlines and documents the scientific basis of the model and provides a list of parameters and algorithms. The deliverable includes sample output from the model. The model is validated through comparison with observations on a global scale, as well as through sensitivity analysis of parameter values. Finally, the deliverable outlines steps towards integration, in a trait-based model of the pelagic environment that explicitly simulates both the unicellular and multicellular (metazoan and fish) components.

2 Deliverable description from DoA

D3.1 Mechanistic, trait-based model for unicellular plankton (M12, DTU). Code, documentation and validated implementation example for simulating the unicellular component of marine ecosystems. Implementation will be in a vertical water column under a variety of relevant physical forcing scenarios. Model will include algorithms to report the functional diversity unicellular plankton community.

3 Contribution to ECOTIP tasks

This deliverable is a demonstration of a trait-based model for unicellular plankton in the pelagic ocean. This deliverable integrates across several tasks. It is a direct product of Task 3.1 (*Generalized trait-based model of plankton)* and feeds directly into Task 3.2 (*Mechanistic size-based fish community model*) and Task 3.3 (*Bayesian Network to visualize stressors and vulnerabilities*).

With regards Task 3.1, important outcomes for biodiversity and potential regime shifts are net productivity, community size structure, community transfer efficiency and export flux. These aspects will be verified against the output of Task 2.2 (*Understanding the pelagic and benthic processes and their alteration due to environmental change*) and Task 2.3 (*Evidence of altered ecosystems due to changes in biodiversity distribution, invasions and trophic interaction*s) when the full model is completed (MS18). The ultimate ambition of this task is to include trait-based approach into biogeochemical models as a component to Earth System Models.

A more general description of the contribution of this deliverable to ECOTIP tasks can be seen in Table 1.

Table 1: Contribution of D3.1 (Mechanistic trait.-based model for unicellular plankton) to addressing the knowledge gaps identified by ECOTIP

Knowledge gaps to be investigated in ECOTIP	Contribution by deliverable D3.1
Biodiversity in terms of functional (trait) diversity is rarely quantified in general and in the Arctic, despite its demonstrated value in relating changes in biodiversity to changes in ecosystem functions and services	Provides a theoretical basis for the prediction of functional diversity of plankton community (unicellular component), how these relate to ecosystem functions and so to ecosystem services.
Mechanisms behind responses of Arctic biodiversity to multiple stressors are mostly unknown which hampers predictions	Provides a framework where the effects of multiple stressors acting at one level of the ecosystem can propagate to ecosystem function.
The interaction of multiple environmental stressors, functional (trait) diversity and ecosystem vulnerability is unknown which hampers upscaling and extrapolation of the effects	
Non-linear changes and potential ecosystem tipping cascades in Arctic marine ecosystem can result from changes in composition of natural community, but have not yet been demonstrated, nor have their tipping potential estimated.	Provides a means by which tipping elements in the ecosystem can be explored under climate change. Here under numerical experiments and sensitivity analysis can be conducted to map out tipping thresholds and their safe operating space.
The Arctic Ocean is not uniform and the geographic variation both in biodiversity and anthropogenic stressors is unknown and likely to be high – we lack both long-term data series, and pre-historical baseline data.	Regional scale transport matrices when available can be used to characterize geographic variation
Lack of optimized monitoring strategies and significant gaps in knowledge on (particularly) microbial and plankton diversity and across trophic levels, hamper accurate Arctic biodiversity estimates.	Sensitivity analyses conducted on the trait-based model will identify knowledge gaps where targeted observations and monitoring would be invaluable.
Phenology is poorly resolved. There are a few examples of a changing phenology in the Arctic marine environment, such as earlier spring bloom following from an early ice melt. However, we do not know how that affects trophic linkages and food webs.	Provides a basis for the development of an integrated ecosystem model including higher trophic levels (metazoans and fish) and up to marine productivity, wherein the impact of seasonality changes in physical forcing (temperature, sea ice, stratification) can be explored.
Socio-economic changes following from the biodiversity change have not been resolved, although the change in fish communities and production will have large consequences for industry, infrastructure, local economy and life-style of indigenous people.	The key attribute of trait-based modelling is that it provides a theoretical basis for the prediction of ecosystem structure and function. A reliable means of prediction is a prerequisite for developing adaptation strategies. The model its self will be used to provide input to the Bayesian Network model that is seen as an important management tool for socio-economic planning, conservation and sustainable development.
Adequate adaptation strategies have not been proposed to facilitate informed decision-making on local to regional scales regarding biodiversity conservation and sustainable development	

4 Trait-based approach to modelling unicellular plankton

Understanding what shapes the biodiversity of marine plankton communities has been a focal point in evolutionary and functional ecology ever since the seminal work (Hutchinson 1961) who pointed out the apparent paradox; that plankton exhibited a much higher diversity than could be explained by theory (specifically, as might be predicted by the completive exclusion principle). The resolution of this paradox was not so much about re-evaluating the theory, but rather the realization that the environment that different plankton inhabit, is much more varied than it may appear at first glance. In particular, the physical size of plankton cells constrains the effectiveness of various resource acquisition modes (uptake of dissolved organic and inorganic compounds, light harvesting, ability to ingest other organisms), while also affecting other vital processes such as metabolic demands and mortality risk. These constrains laid against the ever-changing gradients in resource supply, means the pelagic environment is much more complex than previously supposed, and is thus able to sustain a richly diverse plankton community. It is against this background that trait-based modelling of marine ecosystems developed (Litchman & Klausmeier 2008, Follows & Dutkiewicz 2011, Litchman et al. 2013, Kiørboe et al. 2018), the conceptual foundations of which can be summarized as:

- The life tasks of organisms are constrained by tradeoffs. Put another way, no organisms can be the best at all things at all times.
- There are key traits (mathematically, these can be thought of as variables) that determine the outcome of tradeoffs (mathematical functions). These key traits are ideally
  (1) General in that they can be used to classify large sectors of the plankton community,
  (2) Functional in that they are engaged in tradeoffs that directly impact the performance of organisms (i.e. their fitness) and
  (3) Observable so that model validation can be conducted.
- Under prevailing environmental conditions, the set of trait values that provides its possessor with highest fitness (growth rate) will supplant those with lower fitness.

In the context of modelling ecosystems, the trait-based approach allow for the representation of whole groups of species in terms of generalized traits and their associated trade-offs.

4.1 Background

The traditional approach to modelling marine ecosystems is through so-called NPZ (Nutrient – Phytoplankton – Zooplankton) models (*e.g*. Fasham et al. 1990, Franks 2002) that depict marine ecosystems as compartmentalized functional types based largely of trophic relationships – phytoplankton require nutrients (and light) to grow, and are eventually eaten by zooplankton who recycle nutrients back into the environment. Such models when rigorously calibrated, can be highly successful in quantifying many aspects of the marine environment from annual cycles of productivity to the biological influence on the ocean' s biogeochemistry (*e.g.* Weitz et al. 2015). The obvious simplicity of the basic NPZ configuration invites the addition of other compartments representing for instance, different classes of nutrients (Nitrogen in its various forms, Phosphorous, Silicon) or other parts of the marine ecosystem such as bacteria, micro-zooplankton, detrital material, and fish. While providing high-resolution results in many areas, NPZ models and their derivatives suffer form 3 major problems

(1) Their constituent compartments generally represent a large variety of different species. For instance, the "Z" box could represent everything from ciliates to copepods, each with quite different feeding preferences, growth rates and predation risks. The calibration parameters represent community level averages that can potentially

vary widely over a seasonal cycle for instance. To resolve this, the temptation then is to split the "Z" compartment into several more specific compartments. This however comes at a cost, as the number of parameters required to drive a system in *n* compartments increases rapidly as n^2.

(2) Calibration parameters tend to be region specific. A model configuration that works well off the coast of California will not work so well in the Bay of Biscay. A consequence of this is that NPZ type models fail at the global scale. More critically, even on a regional scale, NPZ models will fail as global change pushes them outside of their calibration envelop.

(3) Finally, NPZ models are by their design, unable to make predictions about biodiversity. The structure of the ecosystem is hard-wired into the compartments of the model, and while the biomass within these compartments can wax of wane, no novel compartments can be generated.

Within the context of ECOTIP, NPZ models are unsuitable, as they can address neither climate change nor biodiversity. It is with this reasoning in mind, the ECOTIP is using a trait-based approach in in addressing the response of ecosystem services to changing biodiversity and climate. In this, we have developed a new conceptual framework for marine ecosystems that we term the NUM (Nutrients – Unicellular - Multicellular) approach (Serra-Pompei et al. 2020).

While there could be a long discussion as to the relative merits and technical details, the key thing that the NUM approach introduces, and why it can do so much more that traditional NPZ models is that it incorporates additional information in the form of evolutionary theory.

4.2 Conceptual frame work – NUM

Our trait-based NUM model is configured along 3 principle concepts: Firstly organism size is a master trait (Andersen et al. 2016) as it regulates, or is strongly correlated with several key life processes such as metabolic rate (Brown et al. 2004), sensory ability (Martens et al. 2015), offspring size (Neuheimer et al. 2015), uptake kinetics (Pasciak & Gavis 1974, Aksnes & Egge 1991, Armstrong 2008) and predation risk (De Robertis 2002, Kiørboe 2011).

Secondly, size limits the carbon content of an organism, and it is from the limited pool, that an organism must invest in key life tasks. That is, a varying percentages of its carbon mass is devoted to photosynthesis, nutrient uptake, cell structure, biosynthesis and so forth. We term these allocation traits. The organizing principle here is that all unicellular organisms are an assemblage of a small number of common components (*e.g.* organelles, structures, biochemical pathways) that carry out basic cell functions (Figure 1). Diversity is primarily determined by the cell's investment in these components, both in absolute (and hence overall mass) and relative terms (varied cellular functioning). Investments ruled by trade-offs.

Finally, the organism's success is measured in terms of its reproductive performance.

This deliverable focusses on modelling the base of the pelagic food web: the N and the U components of the NUM model framework. This excludes metazoans (e.g. copepods, krill, fish) and these will be incorporated in the next phase. The N and U components of the model spans the base of the pelagic food web, and includes autotrophic phytoplankton, heterotrophic protists and bacteria, diatoms and mixotrophs.

The full NUM model – integrating up to metazoans and fish – will be demonstrated in D3.3 (Mechanistic model of zooplankton) due in M24.

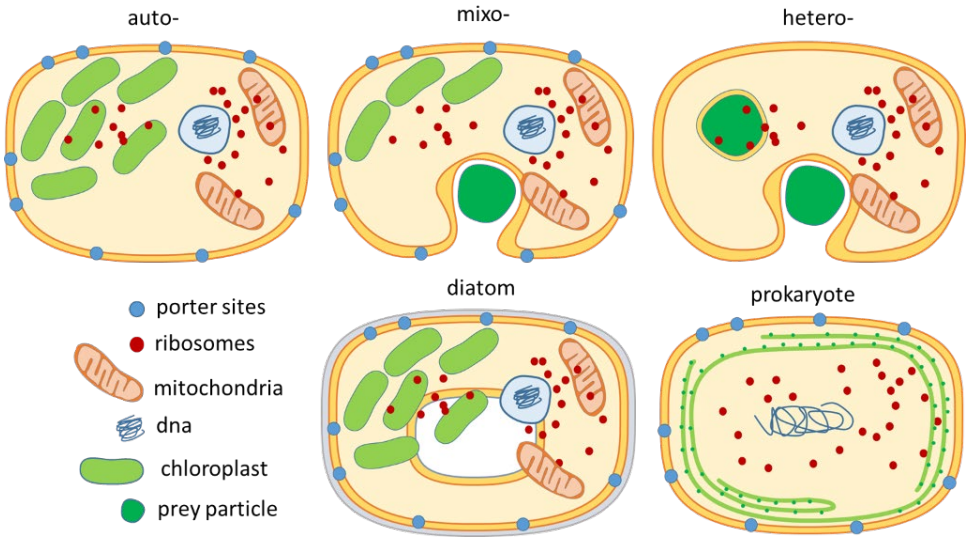


Figure 1: sketch of the various unicellular configurations considered in the NUM model, and their investment in various structures (e.g. porter sites, ribosomes, mitochondria, chloroplasts etc) that facilitate cell functions (e.g. light harvesting, nutrient uptake, phaegotrophy). The main axis for eukaryote generalists follows from autotrophs to mixotrophs and heterotrophs. Diatoms from a separate type, as they are incapable of phaegotrophy (they are obligate phototrophs), and depend on an additional nutrient Si. For completeness, we also include a prokaryote axis that can invest in phototroph (e.g. Prochlorococcus), or heterotrophy fuelled by dissolved organic carbon.

The NUM model splits marine organisms into 2 broad classes; unicellular and multicellular. This is in contrast with traditional modelling frameworks that perform the split along trophic strategy – roughly speaking between plants and animals as one might expect from a traditional terrestrial ecology viewpoint. The rationale for the NUM split for marine ecosystems is that a considerable section of marine ecosystems are composed of so-called mixotrophs – organisms that exhibit both plant-like and animal-like characteristic: they photosynthesis and eat other organisms. Trying to resolve these along a plant-animal model structure has proven unwieldy. In contrast, the unicellular – multicellular split is a clear demarcation between 2 fundamentally different life history requirements. Specifically, unicellular organisms reproduce by cell division, so that daughter cells are roughly the same size as their parent cell. The early life stages of multicellular organisms however, are generally much smaller than reproductive adults, with the mass of an adult being typically 100 to 1000 times that of the juvenile/egg. This growth from small juvenile to large adult means that multicellular organisms have occupy different trophic niches throughout their life time – both in terms of what they eat, and what seeks to eat them.

The NUM model structure allows both the mixotrophic aspects of unicellular organisms, and the ontogenetic aspects of multicellular organisms to be explored. Indeed it allows the trophic state of model organisms to emerge as a model prediction. This is a key feature of the NUM modelling framework, that will be exploited in ECOTIP to provide insights and predictions as to how global change will trigger complex re-organization of marine food webs with eventual impacts of biodiversity and ecosystem services.

4.3 Generalists, Mixotrophs and Diatoms

In our model, we consider size as a master trait, and measure it in terms of the cell's carbon mass which we denote x [µg C]. This includes all the carbon contained in the cell, its cell wall, membranes, nucleus and internal organelles (for eukaryotes), and all other organic material (enzymes, ribosomes etc). The carbon density and C:N ratio of this material falls generally into 2 categories, the cell wall which has a high C density and C:N ratio, and all the other stuff that has a lower density and C:N ration. The carbon contained in the cell wall is generally a minor fraction, but becomes important as the cell becomes small, or contains a large vacuole.

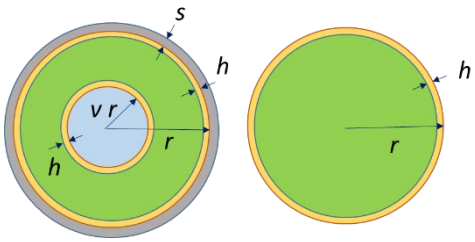


Figure 2 Assumed spherical geometry of cells.

We consider the physical structure of all cells to be essentially spherical, and we use the total carbon mass of the cell as a common metric comparing all cells. Specifically, if c_b [µg C µm^{-3}] as the carbon density of the protoplasm, and c_m [µg C µm^{-3}] as the carbon density of the cell membrane, then

$$x = \frac{4\pi}{3}\left[(r-h_m)^3(c_b-c_m)+r^3c_m\right] \quad (4.1)$$

Where h_m [µm] is the thickness of the cell wall, and r [µm] is the radius of the cell.

The physical structure of a diatom follows similarly, but with a central fluid-filled vacuole separated from the protoplasm by an inner cell membrane. The radius of the cell out to its outer membrane as again denoted r, while its vacuole radius is denoted $v^{1/3}r$ [µm]. That is, v is a measure of the relative volume of the diatom cell that is occupied by the vacuole. Diatoms also have an outer shell composed of mostly of silicate. To provide a consistent comparison with generalist cells, we define x [µgC] for diatoms as the carbon mass of the cell membranes plus all organic material between the inner and outer cell membrane is

$$x = \frac{4\pi}{3}\left[[(r-h_m)^3+(v^{1/3}r+h_m)^3](c_b-c_m)+r^3(1-v)c_m\right] \quad (4.2)$$

The physical radius of a cell r, is an important trait in its own right, as it determines both the diffusive flux of nutrients to the cell, as well as its risk of predation. Equations 4.1. and 4.2 allow us to write a generalized function $r(x,v)$ that relates the physical size of a cell to its carbon mass and vacuole size. While an analytic function is not very elegant, a numerical algorithm can be readily invoked to provide $r(x,v)$.

The structure of the cell also dictates some of the more important constraints on cell stoichiometry. Specifically, while most of the cell structure has a relatively constant C:N ratio, cell membranes are typically low in N content. Thus small or highly vacuolated cells that invest disproportionally in cell membranes, require less N. Writing the N:C ratio as γ_b and γ_m for protoplasm and membranes respectively, then for generalists

$$\gamma_{cell}^{g} = \frac{4\pi}{3x}\left[(r-h_m)^3(c_b\gamma_b-c_m\gamma_m)+r^3c_m\gamma_m\right] \quad (4.3)$$

while for diatoms

$$\gamma_{cell}^{d} = \frac{4\pi}{3x}\left[[(r-h_m)^3+(v^{1/3}r+h_m)^3](c_b\gamma_b-c_m\gamma_m)+r^3(1-v)c_m\gamma_m\right] \quad (4.4)$$

For diatoms there is the additional requirement for silicon (Si) for the construction of the shell. The thickness of the shell (more properly term the frustule) is not constant but depends on several factors size and growth stage. In this work we use the empirical relationship

$$h_s = (2.28 \times 10^{-3})\ r^{0.72} \tag{4.5}$$

(Hansen & Visser 2019) derived from an analysis of several observational sources. This gives reasonable values of shell thickness between 2 and 45 nm consistent with the observed range. The shell thickness gives an estimate of the silicate requirement of the cell

$$\varsigma_{cell} = 4\pi r^2 h_s(r)\sigma_s / x \tag{4.6}$$

where σ_s is the density of Si in the shell [µg Si µm^{-3}].

Table 2. Parameters related to cell structure

symbol	definition	value	references
x	Cell carbon mass	*variable* µg C	Eqs 4.1 & 4.2
r	Cell equivalent spherical radius	*variable* µm	Eqs 4.1 & 4.2
c_b	Carbon density in protoplasm	0.125 (0.1 – 0.16) pg C µm^{-3}	(Strathmann 1967, Raven 1987)
c_m	Carbon density in membranes	0.6 (0.5 – 0.6) pg C µm^{-3}	(Raven 1987)
h_m	Thickness of membranes	0.008 (0.007 – 0.009) µm	(Raven 1987)
h_s	Thickness of diatom shell	*variable* µm	Eq 4.5
γ_b	N:C ratio in cytoplasm	1/6 µg N (µg C)$^{-1}$	(Raven 1987)
γ_m	N:C ratio in membranes	1/32 µg N (µg C)$^{-1}$	(Raven 1987)
γ_{cell}	N:C ratio in cell	*variable* µg N (µg C)$^{-1}$	Eqs 4.3 & 4.4
ζ_{cell}	Si:C ratio for diatoms	*variable* µg Si (µg C)$^{-1}$	Eq 4.6
σ_s	density of Si in diatom shell	10.3 pg Si µm^{-3}	(Cadier et al. 2020)
ρ_b	mass density of cytoplasm	1100 (1030-1100) kg m^{-3}	
ρ_v	mass density of vacuole	1010 kg m^{-3}	(Boyd & Gradmann 2002)
ρ_s	mass density of diatom shell	2200 kg m^{-3}	(Miklasz & Denny 2010)

4.4 Allocation trade-offs

The ability of a cell to perform certain tasks can be viewed in terms of the carbon mass that they allocate to specific "machinery" – organelles, membranes, enzymes etc. Indeed, it can be surmised that the overall rate Q_{task} [day^{-1}] at which particular tasks can be conducted is a monotonically increasing function the carbon allocation x_{task} [µg C]. Specifically, $Q_{task} \rightarrow 0$ as $x_{task} \rightarrow 0$, and $Q_{task} \rightarrow Q_{task,max}$ and $x_{task} \rightarrow x$. The art here is to come up with a scheme whereby the overall performance of a cell can be built up by a manageable number of mechanistically accessible tasks.

Figure 3 sketches out the major tasks involved in phototrophic growth.

The carbon mass of the organism is allocated to different structures according to

$$1 = \phi_L + \phi_M + \phi_N + \phi_B + \phi_F \tag{4.7}$$

where the ϕ's represent the fraction of the cell's carbon mass that is associated with (*L*) light harvesting, (*M*) cell wall and membranes, (*N*) porter sites and nutrient uptake, (*B*) biosynthesis and (*F*) the ingestion and processing of particulate prey. Different trophic modes of the cell thus depend on investments, e.g. autotroph $\Rightarrow \phi_F = 0$, heterotroph $\Rightarrow \phi_L = 0$.

This becomes an important demarcation between diatoms and generalists in so far as diatoms, because of their rigid shell, are incapable of particle ingestion, and are thus considered obligate phototrophs.

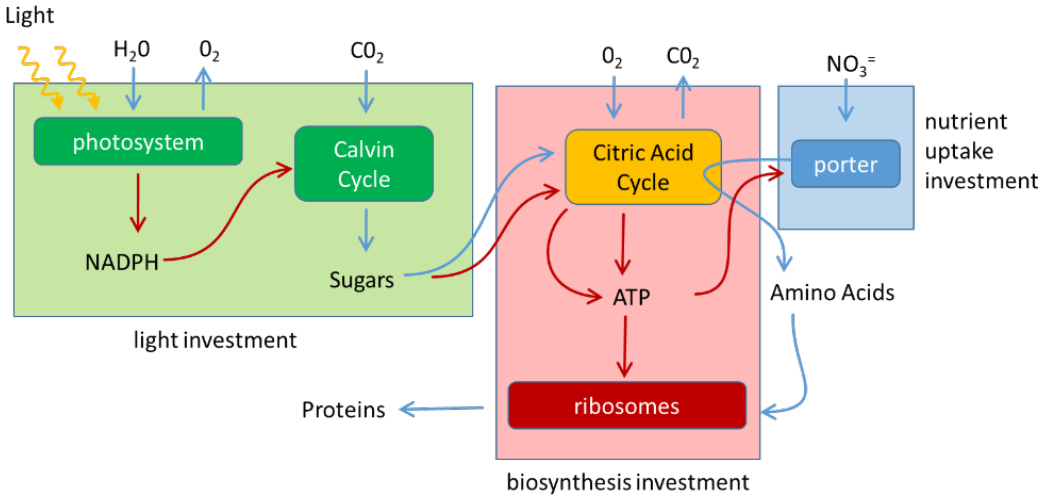


Figure 3: sketch of the energy and material pathways for a phototroph. Light harvesting and the Calvin cycle produce sugars that serve as both a source of fixed carbon and energy. Parallel to this, nutrients are taken up from the environment via porter sites located on the cell wall. These two supply lines feed into biosynthesis, essentially the Citric acid cycle taking place in the mitochondria that produces energy and amino acids (plus all kinds of other products), and ribosomes that weave these into proteins. The production of proteins is essentially cell growth. The rate at which each of these 3 processes run depends on the cell's investment schedule. For any given investment schedule, the overall growth rate cannot exceed that of the slowest process, a concept usually interpreted as Leibnitz's law of the minimum. The optimal investment schedule is that for which all 3 processes ran at the same rate.

4.5 Affinities and Uptake

We take a generalized approach to the uptake of resources (light, nutrients, particulate matter), writing uptake rates in terms of affinities, and maximum uptake rates. Both of these are in general functions of cell size *r* (or exchangeable with cell carbon mass *x*) and investment ϕ. For instance, for a given resource *R*, the affinity A_R is given by

$$A_R = A_{\max,R}(r)\frac{\alpha_R \phi_R x(r)}{\alpha_R \phi_R x(r) + A_{\max,R}(r)} \tag{4.1}$$

where α_R is the affinity gain per unit investment ϕ_R $x(r)$ (Chakraborty et al. 2017, 2020). This reflects a reasonable functionality, in that when investment is low, the affinity is proportional to investment ϕ_R, and goes to zero when investment goes to zero. For high investment however, affinity becomes limited due to some physical or physiological constraint.

The uptake rate of any resource (*R;* a dissolved nutrient, light, or particulate prey) is estimated as:

$$J_R = A_R \cdot R \tag{4.9}$$

4.5.1 Light

For instance, the affinity for light becomes constrained by the cross sectional area of a cell, and further investment in chloroplasts cannot increase photon capture. Indeed, this suggests a functional form for maximum light affinity according to

$$A_{\max,L} = c_L r^2 \tag{4.10}$$

Where c_L is a constant (see below). This scaling tends to show up in meta-analyses conducted across broad ranges of phytoplankton (Edwards et al. 2015). For the purposes of this work, we use a somewhat more mechanistic description based on classic photon absorption dynamics (Duyens 1956, Papageorgiou 1971, Kirk 1975):

$$A_L = c_L r^2 \left[1 - \exp\left(-\kappa \phi_L r\,(1-v)\right)\right] \tag{4.11}$$

where $c_L = \pi\, y$, with y being the photosynthetic quantum yield [µmol C µE^{-1}]. Investment in photosynthesis ϕ_L, can be lined linked to the light attenuation coefficient as:

$$\lambda = ac = (3/4)\ \kappa\phi_L \tag{4.11}$$

(cf. appendix) where an increase in investment, increases the concentration *c* of plastids; the return on investment is governed by the parameter κ [µm^{-1}]. Typical values (Raven 1984) suggest that λ = 0.1 µm^{-1} when about half of the cell's carbon is associated with photosynthesis. This suggests a value of κ = 0.2 µm^{-1}.

Finally, the fixed carbon flux to the cell J_L [µgC day^{-1}] from photosynthesis is then is given by:

$$J_L = yL\pi r^2 \left[1 - \exp\left(-\kappa \phi_L r\left(1-v\right)\right)\right] \tag{4.12}$$

Table 3: Parameters associated with light harvesting

symbol	definition	value	references
y	Photosynthetic quantum yield	0.05 (0.02 – 0.07) µg C µE^{-1}	(Sarthou et al. 2005)
c_L	Photosynthetic affinity constant (= $\pi\, y$)	0.18 µg C µE^{-1}	
a	Cross-sectional area of a chromophore	2270 m^2 (mol chromophore)$^{-1}$	(Raven 1984, 1997)
c	Chromophore number density in cytoplasm	up to 40 mol chromophore m^{-3}	(Raven 1984)
λ	light extinction coefficient within the cytoplasm	order 0.1 µm^{-1}.	(Raven 1984)
κ	Light investment coefficient	0.2 µm^{-1}	(Hansen & Visser 2019)

4.5.2 Nutrients

Likewise the uptake of nutrients becomes limited by diffusion which is ultimately controlled by the linear dimension of the cell (Pasciak & Gavis 1974). This would suggest a functional form for maximum nutrient affinity according to

$$A_{\max,N} = c_N r \tag{4.13}$$

For diffusion limitation for a spherical absorber, the maximum flux is

$$J_N = A_{\max,N} N \approx 4\pi r D N \tag{4.14}$$

suggesting that $c_N = 4\,\pi\, D$ where D is the diffusivity of the dissolved nutrient. This is what might be expected for large cells. For smaller cell, another constraint comes into play, namely the number of porter sites the cell can support on its surface (Aksnes & Egge 1991, Fiksen et al. 2013). Unfortunately, a complete mechanistic description is as yet not available. We turn instead to an analysis of measured uptake rates across a broad range of unicellular plankton and for a number of different nutrients (Figure 4).

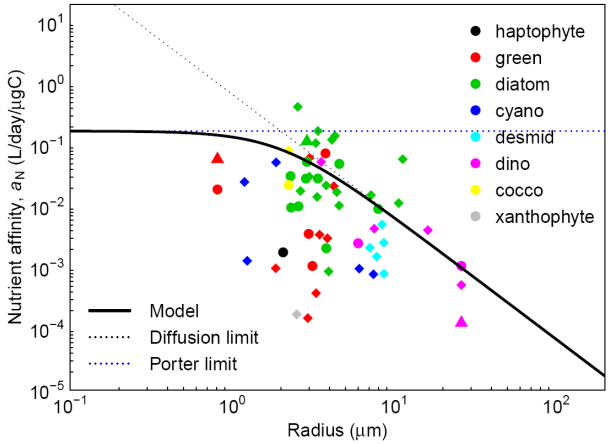


Figure 4: Specific nutrient affinity ($a_N = A_N / x$) as a function of radius. Data from (Edwards et al. 2012). Triangles: ammonium uptakes; circles: nitrate uptake; diamonds: phosphorous uptake. The dotted lines are the theoretical maximum affinity due to diffusion. The transition from diffusion limitation to porter limitation appears at about r = 2 µm.*

A practical affinity that is applied in the model is written

$$A_N = \frac{\alpha_N r(x)^{-2}}{1+(r(x)/r^*)^{-2}} x \tag{4.15}$$

and is used for all dissolved material (nitrates dissolve organic carbon, silicate).

Reconciling (4.13) with (4.16), we have

$$\alpha_N \frac{4\pi r}{3} c_b \approx c_N \tag{4.16}$$

Table 4: Parameters associated with Nutrient uptake

symbol	definition	value	references
D	Diffusivity of dissolved nutrients (N, Si, DOC)	1.62 x 10^{-4} m^2 day^{-1}	(Hansen & Visser 2019)
r^*	Diffusion-porter transition size	2 µm	
α_N	Nutrient affinity scaling factor (mass)	0.68 L day^{-1} (µg C)$^{-1}$ $µm^2$	Andersen
c_N	Nutrient affinity scaling factor (radius)	$c_N = 4\ \pi\ D$	

4.5.3 Phagotrophy

Finally, for phagotrophy (ingestion of prey), empirical evidence from a large range of marine organisms suggests that the volume of water scanned by an organism is roughly 10^6 times its own body volume per day (Kiørboe 2011, Kiørboe & Hirst 2014). This suggests for phagotrophy, the maximum affinity is given by

$$A_{\max,F} = c_F r^3 = \alpha_F x \tag{4.17}$$

where $c_F = 10^6$ day^{-1}. That is, as a general rule of thumb, organisms clear about 1 million times their own body volume per day in search for food. The value for the food affinity factor in terms of mass can then be estimated as:

$$\alpha_F = c_F 3/(4\pi c_b) \tag{4.18}$$

Trophic transfer within the model is assumed to proceed according to the principle that big eats small. While this is not a rigorous in the trophic arrangements for unicellular organisms as it is for metazoans, it does allow for a general framework to be constructed, that can act as a starting point form which more nuanced models can be derived.

Following a number of theoretical approaches, we define a size preference function according to

$$\theta_{i,j} = \exp\left[-\left(\log\left(\varepsilon\frac{r_j}{r_i}\right)\right)^2 / \left(2\omega^2\right)\right] \tag{4.21}$$

where r_i and r_j are the equivalent spherical radius of the predator and prey respectively, ε is the preferred predator-prey size ratio, and ω is the width of the size preference.

With respect to vulnerability, we treat the two clades, generalists and diatoms slightly differently. While diatoms enjoy some measure of protection from vacuolization (accounted for in the size ratio above), their frustule also provides some protection. We model this by providing a "palatability" factor p.

Writing the biomass of unicellular plankton (both generalists and diatoms) of a particulate type (size class, trait value etc.), as B_i the total food available for a generalist in size class i is

$$F_i = \sum_j p_j \theta_{i,j} \; B_j \tag{4.22}$$

(cf. section 5). In its simplest form, this leads to a potential particulate feeding flux to a mixotrophic generalist of size i of

$$J_{F,i} = \alpha_F \phi_{F,i} x_i F_i \tag{4.23}$$

Table 5: parameters associated with phagotrophy

symbol	definition	value	references
α_F	Food affinity factor (mass)	0.018 L (µg C)$^{-1}$ day^{-1}	
c_F	Food affinity factor (radius)	10^6 day^{-1}	(Kiørboe & Hirst 2014)
ε	preferred predator-prey size ratio (radius)	7 (2-10)	(Cadier et al. 2020)
ω	width of the size preference window (radius)	1/32 µg N (µg C)$^{-1}$	(Raven 1987)
p	palatability	0.6	(Cadier et al. 2020)

4.6 Growth

We assume throughout that the uptake of resources (C, N and Si) are constantly in stoichiometric balance with the requirements of cellular growth. Specifically, our model does not allow for the cells to store unrequited resources. While this is a proven strategy in several unicellular organisms, it plays out on the timescale of hours. Our primary concern here is cellular division rate, and on the time scale of a cell's life span, the uptake of resources must balance its stoichiometric requirements; no more and no less.

With this constraint, net growth follows the usual constraints set out by Liebig's law of the minimum. There are, however a few issues that complicate this calculation. Firstly, uptake of resources require energy, paid for in terms of fixed carbon, *i.e.* as a portion of the incoming carbon flux. In order to maintain energy efficiency, the uptake of inorganic nutrients (N and Si) should be down regulated. Secondly, ingestion of particulate organic matter comes with inorganic nutrients that can be in excess to requirements. These nutrients should be excreted back into the environment. Finally, we assume no imposed limit on photosynthesis when nutrients are limiting; the cell continues to produce carbohydrates, and that which is in excess to requirements (*i.e.* limited by inorganic nutrient supply) is leaked into the environment as dissolved organic carbon (DOC). This leakage of DOC has no direct impact on the model formulation as reported here, but can be important when we implement osmotrophy as an addition unicellular feeding strategy of prokaryotes (*cf*. section 6).

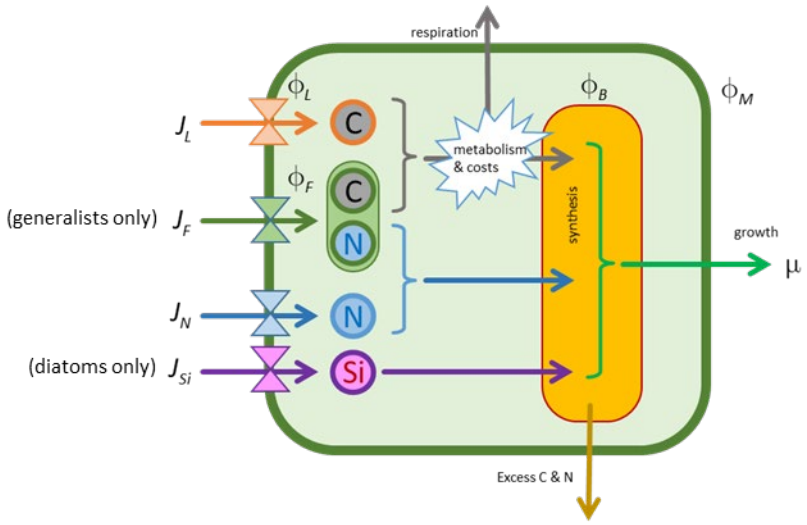


Figure 5: a schematic of the fluxes of carbon and nutrients in a generic cell.

Cell division rate is ultimately dictated by the rate the rate at which resources for biosynthesis enter the cell (Figure 5). Photosynthesis delivers J_L [μg C day^{-1}] fixed carbon (in the form of carbohydrates), nutrient uptake delivers J_N [μg N day^{-1}] fixed nitrogen, and for diatoms, J_{Si} [μg Si day^{-1}] silicon. Particle ingestion by generalists, delivers J_F [μg C day^{-1}] organic carbon which also delivers $J_F \gamma_{cell}$ [μg N day^{-1}] organic nitrogen. The level of J_L and J_F depend on investments ϕ_F, ϕ_L and ϕ_M. Out of the gross carbon flux (J_L for diatoms, $J_L + J_F$ for diatoms), certain energetic costs need to be paid. These include base metabolic costs (including cost of biosynthesis), as well as specific costs for photosynthesis, and nutrient uptake. These latter costs are not insubstantial, and constitute about 1 mole ATP per mole nutrient. Net resources are brought together in biosynthesis to produce stoichiometrically balanced organic

matter, with the net growth rate determined by Liebig's law of the minimum (see below). There is a process of down regulation (see below) that ensures efficient resource management (as might be expected from evolutionary arguments).

4.6.1.1 Biosynthesis

Biosynthesis proceeds on the simple premise that with full access to resources, (fixed carbon and nutrients) the rate at which new organic matter such as proteins are produced is proportional to ϕ_B, the cell's carbon mass that is devoted to biosynthesis machinery (e.g. ribosomes). The current implementation of the model assumes that this allocation is constant.

Future implementations will address the sensitivity of the model to this assumption, and explore if this can be treated as a variable trait value.

4.6.1.2 Liebig's Law of the Minimum

The net growth rate is ultimately dictated by the rate of supply of the minimum – a concept often referred to as Liebig's law of the minimum.

The net growth rate of a cell is determined by the rate at which it can assemble a stoichiometrically balanced replicate of itself. For a diatom, the relevant compounds are C, N and Si. Organic carbon comes for instance from photosynthesis, while mineral nutrients N and Si need to be taken up from the environment.

Let us consider a cell (generalist of diatom) which has a flux of carbon J_L [µg C day^{-1}] coming in from photosynthesis, a flux of carbon J_F [µg C day^{-1}] coming in from phagotrophy, a nitrogen flux of J_N^* [µg N day^{-1}] and a silicate flux J_{Si}^* [µg Si day^{-1}] both coming from diffusive uptake. The cell has a N:C ratio γ_{cell} [µg N (µg C)$^{-1}$] and a Si:N ratio ς_{cell} [µg Si (µg C)$^{-1}$].

Note that the J_L and J_F represent carbon fluxes while J_N^* and J_{Si}^* represent fluxes of specific nutrients. The latter can be converted to potential carbon fluxes as

$$J_N = J_N^* \gamma_{cell}^{-1} \text{ and } J_{Si} = J_{Si}^* \varsigma_{cell}^{-1} \tag{4.24}$$

Further, the ingestion of prey by mixotrophs-heterotroph generalists brings in not only carbon but nutrients as well. If γ_{prey} is the N:C ratio of prey, then the flux of nutrient to the cell associated with phagotrophy is

$$J_{N,F}^* = J_F \gamma_{prey} \tag{4.25}$$

The net potential accumulation rate of biotic material for the cell is given by Liebig's Law of the minimum. For a generalist this can be written as

$$J_C = \min\left\{J_L(1-\beta_L) + J_F(1-\beta_F) - J_N^*\beta_N - J_R,\ \left(J_N^* + J_F\gamma_{prey}\right)\gamma_{cell}^{-1}\right\} \tag{4.26}$$

while for a diatom

$$J_C = \min\left\{J_L(1-\beta_L) - J_N^*\beta_N - J_{Si}^*\beta_{Si} - J_R,\ J_N^*\gamma_{cell}^{-1},\ J_{Si}^*\varsigma_{cell}^{-1}\right\} \tag{4.27}$$

where the β's represent the cost in terms of carbon for the various processes. A cell working on this principle would however, be quite inefficient as it would pay costs of processes that are surplus to requirements. For instance, if carbon were limiting, then J_N and J_S would not only be supplying more N and Si than would be required, but also paying unnecessary energetic costs.

4.6.1.3 Down Regulation

We introduce into Liebig's law of the minimum, three reduction factors δ_C, δ_N and δ_{Si}, such that the diatom will not take up more nutrient and silica than what is required for stoichiometric balance. The rationale for this is to to ensure diatoms do not take up (and pay the cost for) a lot of nutrients they cannot use. This is the concept of down-regulation. We also assume that cells also downregulate carbon fixed by photosynthesis, but leave open the possibility that unrequited fixed carbon is excreted by otherwise nutrient limited cells. This is thought to be an important source of DOM in the marine environment.

We solve the down regulation problem for the three reduction factors by setting all expressions in Liebig's law equal, thereby assuming co-limitation of the three elements.

$$\begin{aligned}
&\delta_L J_L - J_R - \delta_L \beta_L J_L - \delta_N \beta_N J_N - \delta_{Si} \beta_{Si} J_{Si} = \delta_N R_{CN} J_N \\
&\delta_L J_L - J_R - \delta_L \beta_L J_L - \delta_N \beta_N J_N - \delta_{Si} \beta_{Si} J_{Si} = \delta_{Si} R_{CSi} J_{Si} \\
&\delta_N R_{CN} J_N = \delta_{Si} R_{CSi} J_{Si}
\end{aligned} \tag{4.28}$$

As three equations with three unknowns cannot be solved immediately, we solve this system over three times: when carbon is limiting $\delta_L = 1$, when nitrogen is limiting $\delta_N = 1$, when silica is limiting $\delta_{Si} = 1$. Thereby we have three different solutions for the three reduction factors. The condition when $\delta_1 = 1$ and δ_2, $\delta_3 < 1$ occurs when δ_1 is limiting. As only one substrate will be limiting, only one of the three solutions fulfil the previous condition at each set of traits under each set of environmental conditions. In case one of the nutrients are limiting and $\delta_L < 1$ it is manually set it to 1, as the photosynthetic uptake cannot be reduced. The value $1-\delta_L$ would determine the leaked carbon that cannot be used for synthesis because of the limited nutrient availability.

Table 6: Parameters related to respiration, costs and down regulation

symbol	definition	value	references
β_0	Basal respiration	0.04 day^{-1}	(Chakraborty et al. 2017)
β_L	Cost of light harvesting	0.08 µg C (µg C)$^{-1}$	(Raven 1987)
β_N	Cost of N uptake	0.45 µg C (µg N)$^{-1}$	
β_{Si}	Cost of Si uptake	0.45 µg C (µg Si)$^{-1}$	
β_F	Cost of food uptake	0.45 µg C (µg C)$^{-1}$	

4.6.1.4 Cell division rate

With down regulation in place (essentially specifying co-limitation on resources) the net cell division rate is given by

$$\mu = \mu'_{max}\phi_B \frac{J_C}{J_C + \mu_{max}\phi_B x} \tag{4.29}$$

where for a generalists (after down regulation)

$$J_C = \delta_L J_L(1-\beta_L) + J_F(1-\beta_F) - \delta_N J_N^* \beta_N - J_R \tag{4.30}$$

while for a diatom

$$J_C = \delta_L J_L(1-\beta_L) - \delta_N J_N^* \beta_N - \delta_{Si} J_{Si}^* \beta_{Si} - J_R \tag{4.31}$$

The rationale for the form of μ is that the maximum biosynthesis rate is limited by the investment in biosynthesis machinery, and saturates at some maximum growth rate. In the simplest form that we implement here, we set ϕ_B = 0.5, and $\mu'_{max} = \mu_{max}\,\phi_B$ = 2 day^{-1}; a reasonable estimate for a large class of phytoplankton species.

In summary, the specific growth rate [day^{-1}] for a unicellular plankton of type *i* (size class, generalist, diatom, trait value) is given by

$$\mu_i = \mu_{max}\frac{J_{C,i}}{J_{C,i} + \mu_{max} x_i} = \mu_{max} f_i \tag{4.32}$$

where f_i is a convenient expression for the fraction of maximum growth rate that can be realized:

$$f_i = \frac{J_{C,i}}{J_{C,i} + \mu_{max} x_i} \tag{4.33}$$

4.6.2 Mortality and Losses

Mortality due to grazing from generalists is explicitly represented in the model. Specifically, following from the grazing

$$m_i = p_i \sum_j A_{F,j} H(v_j)\theta_{i,j} B_j (1-f_j)/x_j \tag{4.34}$$

Higher trophic level as imposed for instance by zooplankton grazers are represented by

$$m_{h,i} = m_{h,0}(1 - H(v_i)p_i)\ Z\ r_i^{-3/4} \tag{4.35}$$

where $m_{h,0}$ is a grazing constant, Z is the and abundance of metazoan grazers [measured in biomass abundance; μg C m^{-3}], and the $r^{-3/4}$ term represents decreasing predation risk with size (Kiørboe & Hirst 2014).

The last class of mortality to consider is that due to virus and pathogens. For this we assume a quadratic mortality rate of the form

$$m_{v,i}B_i = g_v B_i^2 \tag{4.36}$$

In this case we assume the rate is independent of cell size or type. The rationale here is that particular cell types or sizes that become abundant will attract the attention of specialized virus/pathogens that will rapidly multiply resulting in a faster than linear mortality rate.

The final loss term to consider is that due to sinking. This is after all one of the key processes contributing to the biogeography and seasonality of ocean productivity is the sinking out of detrital material from the surface ocean. The sinking speed can be estimated form Stokes' law, namely, a particle with radial dimension r and density ρ, will sink with a speed w given by

$$w = \frac{2}{9}\frac{gr^2}{\eta}\frac{\rho - \rho_w}{\rho_w} \tag{4.37}$$

where ρ_w is the density of sea water, η is its viscosity, and g is the acceleration due to gravity.

The excess density for any cell can be calculated from its geometry and the density of material within the cell. Specifically for diatoms

$$\rho = (1-v)\rho_b + v\rho_v + \frac{h_s}{3r}\rho_s \tag{4.38}$$

while for generalists, cell density is simply given by $\rho = \rho_b$. Sinking speed is really only relevant for diatoms, and then only when they are stressed. Indeed, diatoms in good health and growing rapidly appear to remain neutrally buoyant, even though the same cells when killed, sink rapidly in accordance with Stokes law. Precisely what mechanism is behind this remain elusive (some manipulation of the ion content of the vacuole sap perhaps). Be that as it may, we introduce a scaling factor to estimated sinking speed according to realized growth according to

$$w = \frac{2}{9}\frac{gr^2}{\rho_w \eta}\left[(1-v)\rho_b + v\rho_s + \frac{h_s}{3r}\rho_s - \rho_w\right]\left[\frac{\mu_{max} - \mu}{\mu_{max}}\right] \tag{4.39}$$

so that $w \to 0$ as $\mu \to \mu_{max}$

Finally, detrital material (POM) also sinks. While in general this is a complex function of particle aggregate size and density (a topic to be taken into account at a later date; see section 0), for the purposes here, we assume a uniform sinking speed of w_{p0} = 5 m day^{-1}

Table 7: Parameters associated with growth, mortality and losses

symbol	definition	value	references
μ_{max}	Maximum growth rate	2 day^{-1}	(Chakraborty et al. 2017)
ϕ_B	Investment in biosynthesis (carbon ratio)	0.5	(Raven 1987)
g_v	Coefficient for quadratic mortality due to viruses and pathogens	0.01 $(mg\ C\ m^{-3})^{-1}\ day^{-1}$	
$m_{h,0}$	Higher trophic level grazing coefficient	0.1	
g	Acceleration due to gravity	9.8 $m\ s^{-2}$	
η	Kinematic viscosity of sea water	$1.3 \times 10^{-6}\ m\ s^{-2}$	
ρ_w	mass density of sea water	1028 $kg\ m^{-3}$	
ρ_b	mass density of cytoplasm	1100 (1030-1100) $kg\ m^{-3}$	
ρ_v	mass density of vacuole	1010 $kg\ m^{-3}$	
ρ_s	mass density of diatom shell	2200 $kg\ m^{-3}$	
w_{p0}	Sinking speed of particulate detrital material	5 $m\ day^{-1}$	

4.6.3 Temperature effects

Temperature effects are incorporated on vital rates in Q_{10} formulations. Specifically, if J_0 is the rate of a process at temperature T_0, then the rate $J(T)$ at any other temperature T is given by

$$J(T) = J_0 Q_{10}^{(T-T_0)/10} \tag{4.40}$$

In this, we choose to place temperature dependence on key processes controlling growth rather than on growth directly (Serra-Pompei et al. 2019). Specifically, uptake rates associated with photosynthesis have a $Q_{10} = 1$ (i.e. are invariant with temperature); those associated with the uptake of dissolved material from the environment have a $Q_{10} = 1.5$; while those associated with respiration and have a $Q_{10} = 2$.

4.6.4 Optimization

The organism's success is measured in terms of its reproductive performance. In the simplest instance, this

$$g = \mu - x \cdot m \tag{4.41}$$

somewhat different rule (Gilliam's rule) for some implementations and define the parameter

$$g' = \frac{\mu}{m} \tag{4.42}$$

a parameter which essentially measures the expected lifetime income of carbon. In either case, the optimal trait configuration is that for which g (or g') is maximized.

5 Ecological Model

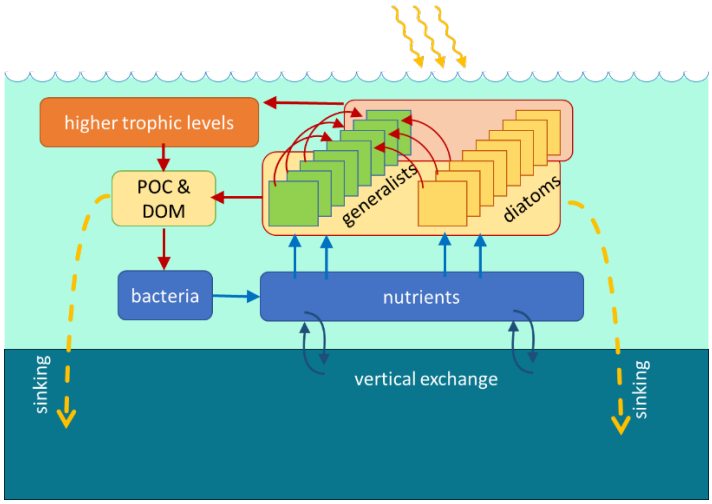


Figure 6: Integrating the unicellular model into an ecological setting.

To complete the model, we need a representation of the physical environment where the key drivers are light, temperature, nutrients (both nitrate and silicate), some estimate of vertical mixing and how it is effected by stratification, and

5.1 Water Column and Community Structure

The physical environment is represented by a 1-D hydrodynamic model that describes the dynamics of nutrients (N and Si), detritus *P*, and the biomass of unicellular plankton **B**. For the purpose of this description, **B** is considered a vector $\mathbf{B} = [B_1\ B_2\ \ldots\ B_i\ \ldots B_{\mathrm{Imax}}]$, that contains the biomass per unit volume of ocean (i.e. µg C m^{-3}) for both generalists and diatoms within specific size bins and allocation schedules at a given point in time and at some location. Other vectors of the same shape are the primary traits; radial size **r**, carbon mass **x**, vacuole size **v**, light investment ϕ_L and phagotrophy investment ϕ_F.

Other relevant quantities describing the unicellular community can be readily calculated. For instance the number densities in each class, allocation schedule and size bin is given by **C** = **B**/.**x** (/. Represents element-wise division). That is

$$\mathbf{C} = \mathbf{B} / .\mathbf{x} = B_i / x_i \tag{5.1}$$

In general, the size of these vectors is $n_{\max} = n_{size} \times n_{types} \times n_{allocation} \times n_{vacuole}$ which can rapidly become a large number. However, there are parts of their general trait space that is unoccupied by mutual exclusion. For instance, because of their shell, diatoms cannot engage in phagotrophy, thus they are obligate phototrophs with $\phi_F = 0$. Likewise, we assume that generalists do not contain vacuoles; $v = 0$. We also assume that the carbon mass investment in nutrient uptake is small. There is a large energetic cost, but to all intents and purposes, $\phi_N = 0$ for both diatoms and generalists. Finally, we also assume that a constant fraction of cell carbon mass is devoted to biosynthesis, ϕ_B = constant for both diatoms and generalists. The last cellular carbon allocation that which goes into the cell membranes ϕ_M is a function of cell size and vacuolation. Indeed, a relatively robust estimate can be written:

$$\phi_M = 4\pi r^2 h_m (1 + v^{1/3}) c_m / x \tag{5.2}$$

which is valid for both diatoms and generalists (v → 0).

The constraint on allocations in light can thus be conveniently written as

$$\phi_L = 1 - \phi_B - \phi_M(x,v) - (1 - H(v))\phi_F \tag{5.3}$$

where *H* is the Heavyside function. In essence, this reduces trait space to a single variable, *v* for diatoms and ϕ_F for generalists.

The size of the vectors thus reduces to $n_{\max} = 2 \times n_{size} \times n_{traits}$ or about 200 for a reasonable coverage of trait space. In addition to the vectors already introduced: **B**, **C**, **x**, **r**, **v** and ϕ_F, there are additional vectors of the same size that can be used to describe important cell properties. Specifically γ_i to represent the N:C ratio in the cell (Eq 4.3 and 4.4) , ζ_i to represent Si:C ratio (Eq 4.6), and sinking speed w_i (Eq 4.39) .

The most general description of the system can be written in terms of the following equations

$$\begin{aligned}
\frac{\partial N}{\partial t} &= -\sum_i \mu_i B_i \gamma_i + \sum_i E_i B_i + \lambda_r P_O + \frac{\partial}{\partial z}\left(K \frac{\partial N}{\partial z}\right) \\
\frac{\partial Si}{\partial t} &= -\sum_i \mu_i B_i \zeta_i + \lambda_r P_{Si} + \frac{\partial}{\partial z}\left(K \frac{\partial Si}{\partial z}\right) \\
\frac{\partial B_i}{\partial t} &= \mu_i B_i - M_i - g_v B_i^2 - m_{h,i} B_i + \frac{\partial}{\partial z}\left(K \frac{\partial B_i}{\partial z}\right) - w_i \frac{\partial B_i}{\partial z} \\
\frac{\partial P_O}{\partial t} &= m_v \sum_i B_i^2 + (1 - \sigma_g)\sum_i M_i - \lambda_r P_O + \frac{\partial}{\partial z}\left(K \frac{\partial P_O}{\partial z}\right) - w_P \frac{\partial P_O}{\partial z} \\
\frac{\partial P_{Si}}{\partial t} &= \sum_i M_i \zeta_i - \lambda_r P_{Si} + \frac{\partial}{\partial z}\left(K \frac{\partial P_{Si}}{\partial z}\right) - w_P \frac{\partial P_{Si}}{\partial z}
\end{aligned} \tag{5.4}$$

where the state variables are:
N [µg N m^{-3}], the concentration of dissolved inorganic nitrogen in the environment
Si [µg Si m^{-3}], the concentration of dissolved inorganic silicon in the environment
B_i [µg C m^{-3}], biomass of different components (*i*) if the living unicellular community
P_O [µg C m^{-3}], concentration of particulate organic matter
P_{Si} [µg C m^{-3}], concentration of silicate within the particulate organic matter pool.

The key processes represented here are
(1) The uptake of nutrients (J_N, J_{Si}) from the environment which, within the constraints of Liebig's law fuels cell division (μB).
(2) Excess nutrients not required by generalists are excreted ($E B$) back to the environment.
(3) Nutrients are also released back to the environment by the remineralization of particulate matter ($\lambda_r P_O$)
(4) Vertical mixing (with turbulent eddy diffusivity *K*) operates on vertical distributions of nutrients
(5) Cells (both generalists and diatoms) are grazed by generalists according to the size and palatability preference function

(6) Unresolved mortality due to viruses and higher trophic levels (zooplankton to fish) are simulated as quadratic mortality terms
(7) Living plankton cells can sink with a velocity w_i [m / day]. This only applies to large diatoms, and then only when they are limited by *Si*. (section 4.6.2)
(8) The particulate matter pool has an organic component (P_O) and a mineral component (P_{si}).

To complete the system of equations we also need boundary conditions. At the surface (z=0) there are a zero flux condition:

$$\frac{\partial}{\partial z}(N, Si, B_i, P_0, P_{Si})\Big|_{z=0} = 0 \tag{5.5}$$

At the bottom of the simulated water column (z = h), the conditions become

$$\begin{aligned} K\frac{\partial}{\partial z}(N, Si)\Big|_{z=h} &= \upsilon\left[(N_0 - N(h)), (Si_0 - Si(h))\right] \\ K\frac{\partial}{\partial z}(P_O, P_{Si})\Big|_{z=h} &= -w_p\left[P_O(h), P_{Si}(h)\right] \\ K\frac{\partial}{\partial z}(B_i)\Big|_{z=h} &= -w_i B_i(h) \end{aligned} \tag{5.6}$$

The physical system as written here is consistent with a vertical water column model such as the General Ocean Turbulence Model (GOTM), with multiple levels in the vertical dimension. A simplified version that can be used for diagnostic work, is a 2 layer ocean model. The corresponding equations for this system are

$$\begin{aligned} \frac{\partial N}{\partial t} &= -\sum_i \mu_i B_i \gamma_i + \sum_i E_i B_i + \lambda_r P_O + \upsilon(N_0 - N) \\ \frac{\partial Si}{\partial t} &= -\sum_i \mu_i B_i \zeta_i + \lambda_r P_{Si} + \upsilon(Si_0 - Si) \\ \frac{\partial B_i}{\partial t} &= \mu_i B_i - M_i - g_v B_i^2 - m_{h,i} B_i - \frac{w_i}{H} B_i \\ \frac{\partial P_O}{\partial t} &= m_v \sum_i B_i^2 + (1-\sigma_g)\sum_i M_i - \lambda_r P_O - \frac{w_P}{H} P_O \\ \frac{\partial P_{Si}}{\partial t} &= \sum_i M_i \zeta_i - \lambda_r P_{Si} - \frac{w_P}{H} P_{Si} \end{aligned} \tag{5.7}$$

Where *H* is the depth of the surface layer, N_0 is the nitrogen and Si_0 is the silicate concentration in the deep ocean. The parameter υ represents the turbulent exchange of dissolved material between the surface and deep layers. It is approximated by $\upsilon \approx K / H^2$ and has units day^{-1}. It varies quite significantly over seasonal cycles, particularly in temperate latitudes where stratification and wind mixing have pronounced annual variations (Cadier et al. 2020).

Within the context of ECOTIP, the model will be imbedded in a transport matrix representation of the physical system (*cf* section 0)

5.2 Examples of results

Verification of the model can be seen in a few representative examples. For instance, Figure 7 shows the abundance of generalists and diatoms in trait space (mass *x*: from 10^{-8} µgC to 10 µg C, ϕ_L: light investment for generalists and *v*: vacuole cell volume fraction for diatoms). This is for typical winter conditions.

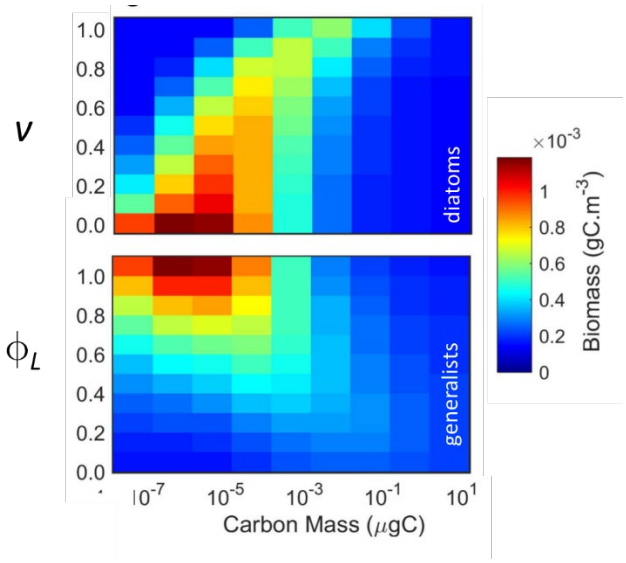


Figure 7: The abundance of generalists (upper panel) and diatoms (lower panel) in a head-to-head competition under winter conditions (low light; L = 46 µE m^{-2} s^{-1}, high nutrients; N = 12 mmol N m-3, Si = 7 mmol Si m^{-3}, low grazing pressure; Z = 5 mg C m^{-3}, and high vertical mixing rate u = 0.5 day^{-1})

This show that under winter conditions, the unicellular community is dominated by relatively small (10^{-6} µg C) cells with small vacuole volumes and high light (and therefor low phagotrophy) investments. Under winter conditions, light tends to be the limiting factor (hence high investment) and there is little benefit to be had in investing in a large vacuole (plentiful nutrients and little predation pressure) or in mixotrophic abilities (mineral nutrients plentiful, and low abundance of prey in the appropriate size class). Note that cells in a particular size class with *v* = 0 and ϕ_L =1 are functionally nearly identical.

In contrast, during spring time conditions (Figure 8) as the water column stabilizes and the spring bloom develops (note the total biomass of unicellular plankton goes up by about an order of magnitude), diatoms with larger vacuoles appear as do mixotrophs with more investment in phagotrophic abilities.

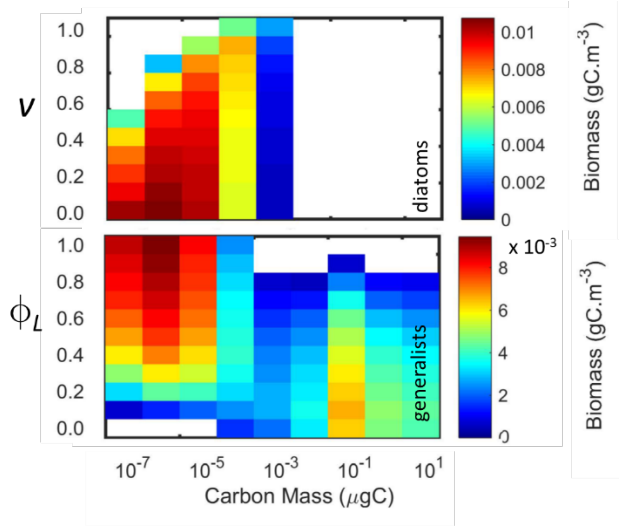


Figure 8: As for Figure 6, but for spring conditions (high light; L=230 µE m^{-2} s^{-1}, high nutrients N= 12 mmol N m^{-3} Si = 7 mmol Si m^{-3}, increasing metazoan grazer biomass Z = 10 mg C m^{-3} and increasing water column stability D= 0.2 d^{-1}) . Regions of trait space in white is not viable.

Note the trophic cascade where a large generalists (mass 10^{-1} µg C) appear that are very nearly 100% heterotrophic.

5.3 Biodiversity Algorithm

A key research question in ECOTIP is to relate expected changes in biodiversity in the Arctic marine environment to important ecosystem services (e.g. carbon sequestration and fisheries production) that local societies rely on. Central to this task is predicting changing biodiversity patterns under climate change. This is where the predictive power of trait based models come into play.

The ability of trait-based models to say something about diversity can be seen in some of the typical out puts that these models provide. Figure 7 and Figure 8 illustrate this. These represent the abundance of organisms (measured in biomass) exhibiting particular trait combinations. A relatively simple algorithm can be setup as

$$\Gamma = \sum_i \psi_i^2$$

where ψ_I is the portion of the entire abundance that is found in the trait bin *i*. This is based on the Simpson index. Similarly we can define

$$\Xi = -\sum_i \psi_i \ln \psi_i$$

based on the Shannon index.

Both of these metrics are in terms of traits*, i.e.* measure trait diversity and are related to but not identical to species diversity. However, trait diversity is more closely related to functional diversity, which is seen as a more robust measure of ecosystem state.

5.4 Validation

The components and parameter settings of the model have been validated through a comparison between results and observations. An example can be found in Figure 9.

The comparison in between model prediction and observation were relatively good in so far as that they provide consistent values and trends. Given the added uncertainty in the physical representation (stratification, mixed layer depth, nutrient concentrations, turbulence, light attenuation coefficient), it seems the ecological model captures the key aspects of the system.

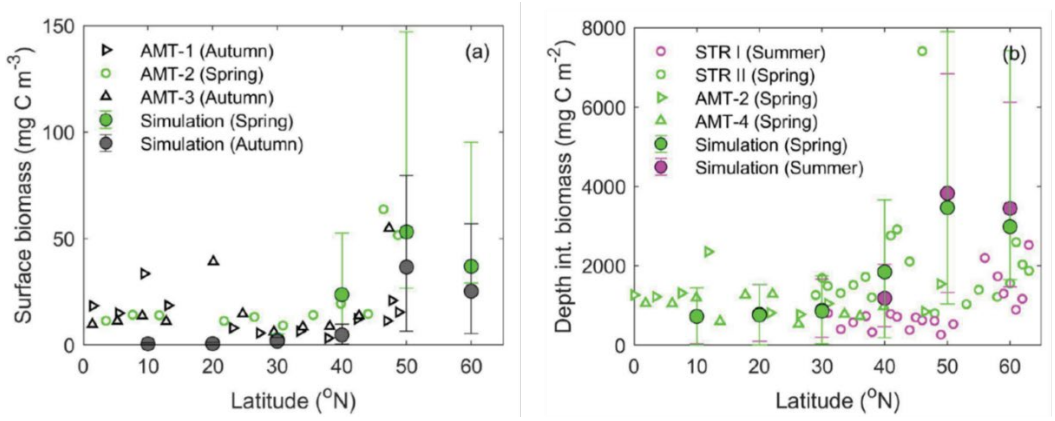


Figure 9: A comparison between observations and model predictions for meridional transects in the North Atlantic; surface (a) and depth integrated (b). Simulations and observations correspond to different seasons. Observations are taken from three Atlantic Meridional Transect (AMT) cruises between Great Britain and the Falkland Islands taking place in 21 September-24 October 1995 (AMT-1), 22 April-22 May 1996 (AMT-2) and 16 September-25 October 1996 (AMT-3) (Marañón et al. 2000). Depth-integrated biomass data are taken from two different sources: (1) two research cruises STRATIPHYT I during July–August 2009 and STRATIPHYT II from April–May 2011 in the Northeast Atlantic Ocean (Marañón et al. 2001, Mojica et al. 2015). Simulations are from (Chakraborty et al. 2020)

As an added exercise, a sensitivity analysis has also been conducted (Figure 10). Key model parameters (those with the highest level of uncertainty) were varied ± 50% of their mean value. These key model parameter included the affinities for light, nutrients and food (α_L, α_N, α_F), the cost of resource acquisition (β_L, β_N, β_F), and the predation by higher trophic levels ($m_{h,0}$). The model output parameters explored were integrated primary production, carbon export, and trophic transfer efficiency where we tracked the % change in predicted value with ± 50% compared to the standard.

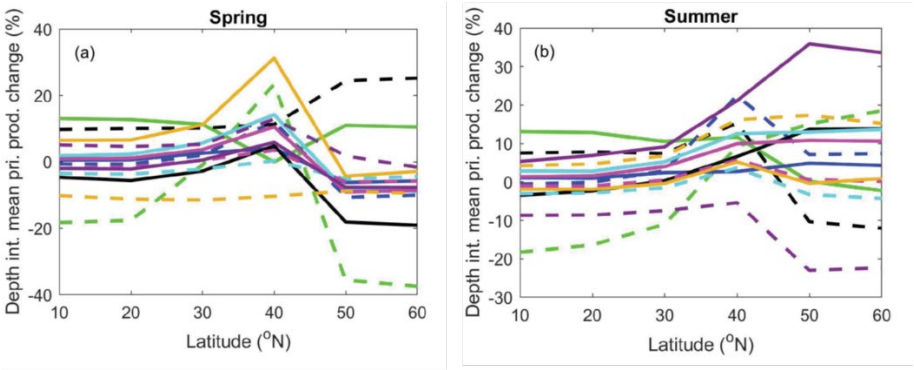


Figure 10: Example of sensitivity of predicted depth integrated primary productivity along a meridional transect during (a) spring and (b) summer condition for various parameters. Sensitivity is expressed as % change of predicted value with ± 50% parameter variation with regards the value predicted at the mean parameter value.

While there is obvious sensitivity to the parameter values, this sensitivity does not exceed the range of the parameter variation. This indicates that over a broad class of parameter settings, the model is well-behaved in that it does not exhibit any rapid (non-linear) transitions due to parameter settings alone.

It should also be noted that the values of parameters, in particular the affinities (α_L, α_N, α_F) are much more constrained by physical laws than the ± 50% variance that they are subject to here. The more critical aspects of the model that are crudely represented, the export flux, and the trophic transfer to higher trophic levels, will be

addressed in explicit representations; namely through modelling metazoan grazer and particle dynamics (see section 6)

6 Steps towards integration

This work represents the first step in an integrated trait-based model of the marine plankton community. The next major push is to implement a metazoan component. As discussed above, this will entail a specific representation of multicellular organisms. In order to get a closed representation capable of describing key ecosystem services, also need to consider the smaller scale – namely bacteria and particles.

6.1 Multicellular component

In keeping with the unicellular model, the multicellular component of the ecosystem will be modelled with size as a master trait. The rational for this structure applies here too; size regulates metabolic processes, trophic arrangement, as well as life history attributes. Indeed, it can be argued that size is an even stronger structuring agent for multicellular organisms as it is for unicellular.

The integrated NUM model is illustrated in Figure 11. This model structure allows for an explicit representation of higher trophic levels, as well as providing important source terms and mechanisms involved in the export of biogenic material (detritus, fecal pellets, deadfall) from the surface ocean into the oceans' interior and eventually to the benthos. The model design has been implemented (Serra-Pompei et al. 2020) but without diatoms or a complete description of POC dynamics.

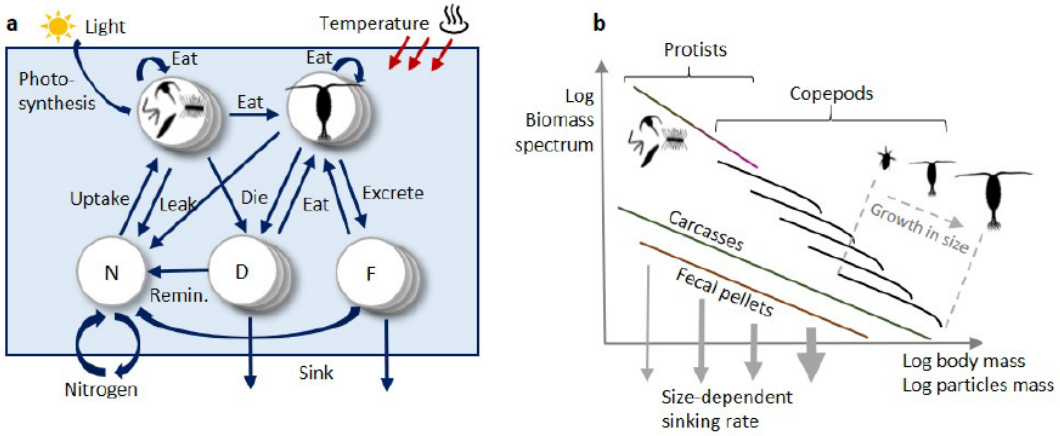


Figure 11: Integrated NUM model structure

The general concept is that all multicellular organisms grow from an egg to an adult, where the egg is a fixed mass of the adult. The rate of egg production depends on adult feeding rate, and the development through life stages is likewise food dependent. Food is provided both by the unicellular components as well as from smaller multicellular components of the plankton community. This structure is particularly relevant for copepods, the most ubiquitous and arguably most important class metazoan grazer found in the world's oceans.

With size as a key trait (either egg or adult) the only other trait considered here is feeding mode. Specifically, whether the copepods are ambush predators, or engage in active feeding (feeding current of cruise predator). This distinction implicates an important trade-off; ambush predation is low risk and low cost, but is also inefficient, active feeding is more efficient, but carries higher risk and cost (Mariani et al. 2013, Kenitz et al. 2017).

This model is able to reproduce many features of the seasonal succession of plankton communities observed on shelf seas (Figure 12)

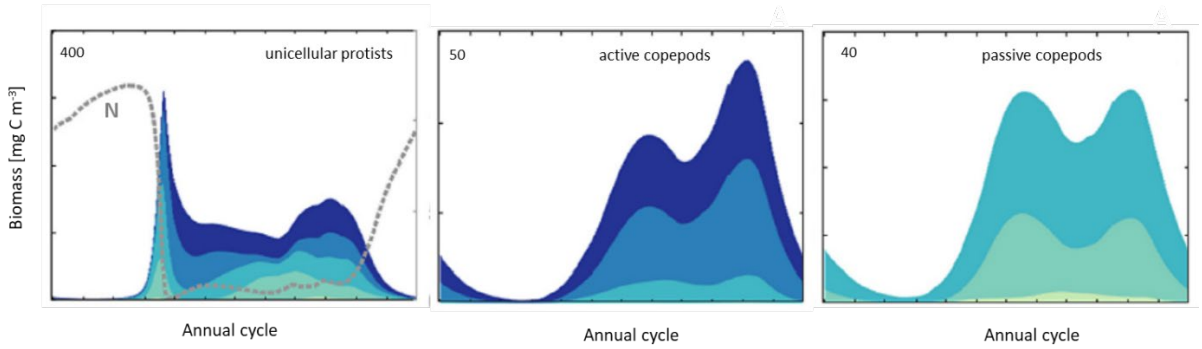


Figure 12: NUM model prediction of the annual cycle plankton community structure of a temperate shelf sea over a seasonal cycle (January to December). The darker colors indicate larger organisms. See Serra-Pompei et al. 2020 for details. The biomass of unicellular protists are estimated using the generalist component of the NUM model.

One of the more important functions of metazoans not implemented here is the effect of vertical migration, both the daily migration exhibited by large groups of metazoans around the world, as well as the seasonal migrations typified in the life history strategies of high latitude copepods. Both of these can have significant impact of the export of biogenic material from the surface ocean (Jónasdóttir et al. 2015, Hansen & Visser 2016, Pinti & Visser 2019, Brun et al. 2019)

6.2 Export Flux and the Biological Pump

One of the key attributed of the NUM model framework is that it is size based. This suggests a natural extension to a size-based representation of POC dynamics. The size of detrital particles are after all a key determinant of their sinking rate (cf Eq 4.37), and by extension the rate at which POC leaves the surface mixed layer.

The extension of NUM into particle dynamics is based on a 2 dimensional state space representation of POC, with particle size and density as the 2 main axes. These also constitute the primary particle characteristics that govern sinking speed.

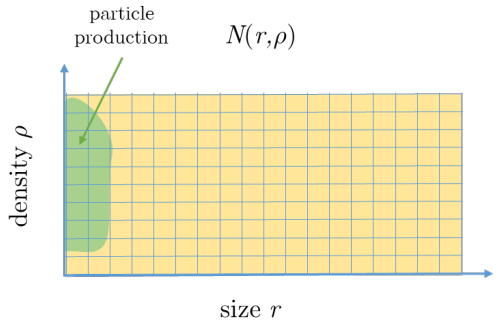


Figure 13: Two dimensional state space for particle dynamics. Size is conveniently given as a logarithmic scale while density is a linear scale.

The size, density and production rate of primary particles is set by the NUM model. Sinking rates and their dependence on ballasting by, for instance the silicate shell of diatoms was addressed above. The characteristics of faecal pellets can likewise be estimated from the diet and size of the metazoans producing them.

The dynamics of POC are determined by the supply of primary particles, the processes by which particles coagulate into larger particles, and the loss of particles out of the surface mixed later by sinking. The collision rate of particles is goverened by 3 physical mechanisms; Brownian motion, differential settling and turbulent collisions (Burd & Jackson 2009).

The central simplify assumption that we use in this work is that when 2 particles of any given size r_i and r_j combine to produce a larger particle, the size of that particle is given by:

$$r_{i\circ j} = (r_i^a + r_j^a)^{1/a}$$

where a is a constant (Jackson 1998). This simplification is based on self-similarity; that the way in which particle pack together is largely scale independent. Note that a=3 corresponds to space filling packing; where the volume of the product particle is exactly the sum of the volumes to the two parent particles. In nature, a is typically 2.7 as can be deduced from macro properties of particle size spectra. A factor $a < 3$ means that on each collision, the volume of the particle increases over the sum of the volume of the two parent particle, with an entrainment of water to fill the extra volume. With this rule in place, it is likewise possible to estimate the density of a product particle. In particular, if the mass of these particles is m_i and m_j respectively, then the mass of the combined particle is

$$m_{i\circ j} = m_i + m_j + \rho_w \frac{4\pi}{3}(r_{i\circ j}^3 - r_i^3 - r_j^3)$$

i.e. the sum of the mass of the 2 parent particles plus a small additional mass due to the incorporation of water (density ρ_w) that fills the extra particle volume. This can be rewritten in terms of particle density as

$$\rho_{i\circ j} = \rho_i \frac{r_i^3}{r_{i\circ j}^3} + \rho_j \frac{r_j^3}{r_{i\circ j}^3} + \rho_w \left(1 - \frac{r_i^3}{r_{i\circ j}^3} - \frac{r_j^3}{r_{i\circ j}^3}\right)$$

where ρ_i and ρ_j the density if the patent particles.

An extremely useful transformation is to write $z_i = \ln(r_i)$. Introducing the function

$$\xi(z_i, z_j) = (1 + e^{(z_i - z_j)a})^{1/a}$$

then the simplified collision product rule follow

$$z_{i\circ j} = z_i + \ln(\xi(z_j, z_i)) = z_j + \ln(\xi(z_i, z_j))$$
$$y_{i\circ j} = z_i \xi(z_j, z_i)^{-3} + z_j \xi(z_j, z_i)^{-3}$$

where $y = \rho - \rho_w$. This represents an efficient means of describing POC dynamics.

With regards integrating the POC dynamics into a full water column representation, we can note an additional simplifying aspect; production and coagulation are processes that primarily effect dynamics in the surface ocean, while degradation, consumption and fragmentation are the main controlling factors in the deep ocean.

This aspect of the NUM model is under development.

6.3 Transport Matrices

Transport matrices (Primeau 2005, Khatiwala 2007) are an efficient means of representing the circulation of the ocean in ecosystem models. They are essentially the product of sophisticated geophysical fluid dynamic models that condense the output, focussing on the rate at which dissolved and suspended material is transported between different locations. They have been typically used for global biogeochemical studies of the ocean (Kwon et al. 2011, DeVries & Weber 2017, Kvale et al. 2017, DeVries & Holzer 2019), but can also be turned towards more regional and ecologically focussed studies.

Essentially, the effect of ocean circulation on a tracer distribution can be written as

$$\frac{\partial \mathbf{c}}{\partial t} + \mathbf{T}\mathbf{c} = \mathbf{q}(\mathbf{c})$$

Where **c** is the concentration of a particular tracer, **T** is the transport matrix, and **q** represents any sources or sinks of **c**. The simplicity of this equation belies a rather elaborate computational scheme.

There is a growing number of transport matrices available as products from GCM's. Some are now being provided for climate change scenarios, and will be made available to ECOTIP form our H2020 project cluster partners.

We have, in the meantime implemented a transport matrix model on the global scale. This is currently being used to investigate the contribution of North Atlantic lipid pump (Jónasdóttir et al. 2015, Visser et al. 2017) to the sequestration of carbon in the deep ocean. Importantly, this allows us go beyond carbon flux estimates and start to disentangle the importance of various processes to maintaining the oceans huge carbon reservoir.

In particular, the spatial distribution of sequestered carbon **c*** from a particular source distribution **q** at steady state can be readily estimated as

$$\mathbf{c}^* = \mathbf{T}^{-1}\mathbf{q}(\mathbf{x}, z)$$

with **c***(z=0)=0, i.e. the surface concentration goes to zero to reflect the emission of sequestered carbon back to the atmosphere. The source term here **q** is a function of geographical location and depth**.** Again, the simplicity of this

equation some numerical complexity and involved the inversion of a large but sparse matrix. Depending of the resolution of the transport matrix involved, this can take between 2 to 20 minutes on a laptop computer.

In Figure 14 we present results for two separate population of deep overwintering copepods.

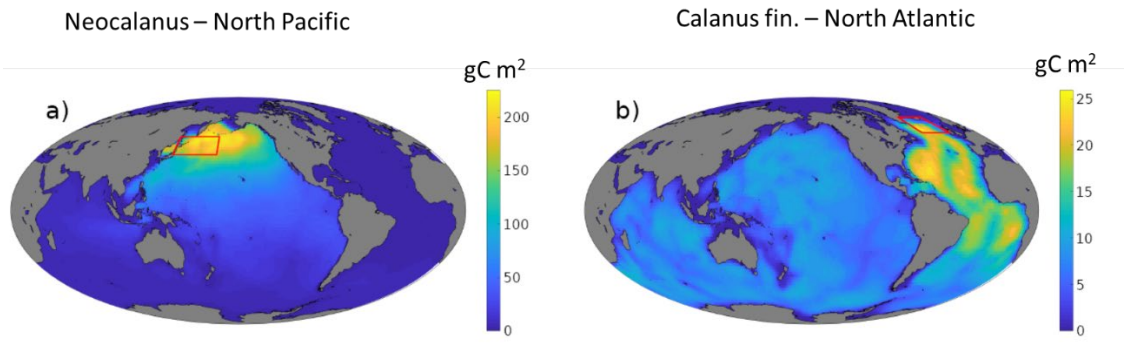


Figure 14: a comparison of the carbon sequestration due to the lipid pump (a) for Neocalanus species in the North Pacific, and (b) for Calanus finmarchicus in the North Atlantic. The maps are of the vertically integrated carbon from each of these sources. While the Neocalnus has a higher regional footprint, carbon from C. finmarchicus is very nearly global in extent. The total carbon sequestered for each population is nearly the same at about 5 Pg C each (the total carbon sequestered in the ocean due to all oceanic pump is about 4000 Pg C).

The model code for this example is included in the Appendix.

7 References

Aksnes DL, Egge JK (1991) A theoretical model for nutrient uptake in phytoplankton. Mar Ecol Prog Ser:65–72.

Andersen KH, Berge T, Gonçalves RJ, Hartvig M, Heuschele J, Hylander S, Jacobsen NS, Lindemann C, Martens EA, Neuheimer AB, Olsson KH, Palacz A, Prowe AEF, Sainmont J, Tarving SJ, Visser AW, Wadhwa N, Kiørboe T (2016) Characteristic sizes of life in the oceans, from bacteria to whales. Annu Rev Mar Sci 8:217–241.

Armstrong RA (2008) Nutrient uptake rate as a function of cell size and surface transporter density: A Michaelis-like approximation to the model of Pasciak and Gavis. Deep-Sea Res Part Oceanogr Res Pap 55:1311–1317.

Boyd C, Gradmann D (2002) Impact of osmolytes on buoyancy of marine phytoplankton. Mar Biol 141:605–618.

Brown JH, Gillooly JF, Allen AP, Savage VM, West GB (2004) Toward a metabolic theory of ecology. Ecology 85:1771–1789.

Brun P, Stamieszkin K, Visser AW, Licandro P, Payne MR, Kiørboe T (2019) Climate change has altered zooplankton-fuelled carbon export in the North Atlantic. Nat Ecol Evol 3:416–423.

Burd AB, Jackson GA (2009) Particle aggregation. Annu Rev Mar Sci 1:65–90.

Cadier M, Hansen AN, Andersen KH, Visser AW (2020) Competition between vacuolated and mixotrophic unicellular plankton. J Plankton Res 42:425–439.

Chakraborty S, Cadier M, Visser AW, Bruggeman J, Andersen KH (2020) Latitudinal variation in plankton traits and ecosystem function. Glob Biogeochem Cycles 34:e2020GB006564.

Chakraborty S, Nielsen LT, Andersen KH (2017) Trophic strategies of unicellular plankton. Am Nat 189:E77–E90.

De Robertis A (2002) Size-dependent visual predation risk and the timing of vertical migration: An optimization model. Limnol Oceanogr 47:925–933.

DeVries T, Holzer M (2019) Radiocarbon and helium isotope constraints on deep ocean ventilation and mantle-3He sources. J Geophys Res Oceans 124:3036–3057.

DeVries T, Weber T (2017) The export and fate of organic matter in the ocean: New constraints from combining satellite and oceanographic tracer observations. Glob Biogeochem Cycles 31:535–555.

Duyens LNM (1956) The flattening of the absorption spectrum of suspensions, as compared to that of solutions. Biochim Biophys Acta 19:1-12.

Edwards KF, Thomas MK, Klausmeier CA, Litchman E (2012) Allometric scaling and taxonomic variation in nutrient utilization traits and maximum growth rate of phytoplankton. Limnol Oceanogr 57:554–566.

Edwards KF, Thomas MK, Klausmeier CA, Litchman E (2015) Light and growth in marine phytoplankton: allometric, taxonomic, and environmental variation. Limnol Oceanogr 60:540–552.

Fasham MJ, Ducklow HW, McKelvie SM (1990) A nitrogen-based model of plankton dynamics in the oceanic mixed layer. J Mar Res 48:591–639.

Fiksen Ø, Follows MJ, Aksnes DL (2013) Trait-based models of nutrient uptake in microbes extend the Michaelis-Menten framework. Limnol Oceanogr 58:193–202.

Follows MJ, Dutkiewicz S (2011) Modeling Diverse Communities of Marine Microbes. Annu Rev Mar Sci 3:427–451.

Franks PJ (2002) NPZ models of plankton dynamics: their construction, coupling to physics, and application. J Oceanogr 58:379–387.

Hansen AN, Visser AW (2016) Carbon export by vertically migrating zooplankton: an optimal behavior model. Limnol Oceanogr 61:701–710.

Hansen AN, Visser AW (2019) The seasonal succession of optimal diatom traits. Limnol Oceanogr 64:1442–1457.

Hutchinson GE (1961) The paradox of the plankton. Am Nat 95:137–145.

Jackson GA (1998) Using fractal scaling and two-dimensional particle size spectra to calculate coagulation rates for heterogeneous systems. J Colloid Interface Sci 202:20–29.

Jónasdóttir SH, Visser AW, Richardson K, Heath MR (2015) Seasonal copepod lipid pump promotes carbon sequestration in the deep North Atlantic. Proc Natl Acad Sci 112:12122–12126.

Kenitz KM, Visser AW, Mariani P, Andersen KH (2017) Seasonal succession in zooplankton feeding traits reveals trophic trait coupling. Limnol Oceanogr 62:1184–1197.

Khatiwala S (2007) A computational framework for simulation of biogeochemical tracers in the ocean. Glob Biogeochem Cycles 21.
Kiørboe T (2011) How zooplankton feed: mechanisms, traits and trade-offs. Biol Rev 86:311–39.
Kiørboe T, Hirst AG (2014) Shifts in mass scaling of respiration, feeding, and growth rates across life-form transitions in marine pelagic organisms. Am Nat 183:E118–E130.
Kiørboe T, Visser AW, Andersen KH (2018) A trait-based approach to ocean ecology. ICES J Mar Sci 75:1849–1863.
Kirk JTO (1975) A theoretical analysis of the contribution of algal cells to the attenuation of light within natural waters II. spherical cells. New Phytol 75:21–36.
Kvale KF, Khatiwala S, Dietze H, Kriest I, Oschlies A (2017) Evaluation of the transport matrix method for simulation of ocean biogeochemical tracers. Geosci Model Dev 10.
Kwon EY, Sarmiento JL, Toggweiler J, DeVries T (2011) The control of atmospheric pCO2 by ocean ventilation change: The effect of the oceanic storage of biogenic carbon. Glob Biogeochem Cycles 25.
Litchman E, Klausmeier CA (2008) Trait-Based Community Ecology of Phytoplankton. Annu Rev Ecol Evol Syst 39:615–639.
Litchman E, Ohman MD, Kiørboe T (2013) Trait-based approaches to zooplankton communities. J Plankton Res 35:473–484.
Marañón E, Holligan PM, Barciela R, González N, Mouriño B, Pazó MJ, Varela M (2001) Patterns of phytoplankton size structure and productivity in contrasting open-ocean environments. Mar Ecol Prog Ser 216:43–56.
Marañón E, Holligan PM, Varela M, Mouriño B, Bale AJ (2000) Basin-scale variability of phytoplankton biomass, production and growth in the Atlantic Ocean. Deep Sea Res Part Oceanogr Res Pap 47:825–857.
Mariani P, Andersen KH, Visser AW, Barton AD, Kiørboe T (2013) Control of plankton seasonal succession by adaptive grazing. Limnol Oceanogr 58:173–184.
Martens EA, Wadhwa N, Jacobsen NS, Lindemann C, Andersen KH, Visser AW (2015) Size structures sensory hierarchy in ocean life. Proc R Soc B Biol Sci 282:20151346–20151346.
Miklasz KA, Denny MW (2010) Diatom sinkings speeds: Improved predictions and insight from a modified Stokes' law. Limnol Oceanogr 55:2513–2525.
Mojica KD, van de Poll WH, Kehoe M, Huisman J, Timmermans KR, Buma AG, van der Woerd HJ, Hahn-Woernle L, Dijkstra HA, Brussaard CP (2015) Phytoplankton community structure in relation to vertical stratification along a north-south gradient in the N ortheast A tlantic O cean. Limnol Oceanogr 60:1498–1521.
Neuheimer AB, Hartvig M, Heuschele J, Hylander S, Kiørboe T, Olsson KH, Sainmont J, Andersen KH (2015) Adult and offspring size in the ocean over 17 orders of magnitude follows two life history strategies. Ecology 96:3303–3311.
Papageorgiou G (1971) Absorption of light by non-refractive spherical shells. J Theor Biol 30:249–254.
Pasciak WJ, Gavis J (1974) Transport limitation of nutrient uptake in phytoplankton. Limnol Oceanogr 19:881–888.
Pinti J, Visser AW (2019) Predator-prey games in multiple habitats reveal mixed strategies in diel vertical migration. Am Nat 193:E000–E000.
Primeau F (2005) Characterizing transport between the surface mixed layer and the ocean interior with a forward and adjoint global ocean transport model. J Phys Oceanogr 35:545–564.
Raven JA (1984) A cost-benefit analysis of photon absorption by photosynthetic unicells. New Phytol 98:593–625.
Raven JA (1987) The role of vacuoles. New Phytol 106:357–422.
Raven JA (1997) The vacuole: a cost-benefit analysis. In: *Advances in Botanical Research*. Elsevier, p 59–86
Sarthou G, Timmermans KR, Blain S, Tréguer P (2005) Growth physiology and fate of diatoms in the ocean: a review. J Sea Res 53:25–42.
Serra-Pompei C, Hagstrom GI, Visser AW, Andersen KH (2019) Resource limitation determines temperature response of unicellular plankton communities. Limnol Oceanogr 64:1627–1640.
Serra-Pompei C, Soudijn F, Visser AW, Kiørboe T, Andersen KH (2020) A general size- and trait-based model of plankton communities. Prog Oceanogr.

Strathmann RR (1967) Estimating the organic carbon content of phytoplankton from cell volume or plasma volume. Limnol Oceanogr 12:411–418.

Visser AW, Grønning J, Jónasdóttir SH (2017) Calanus hyperboreus and the lipid pump. Limnol Oceanogr 62:1155–1165.

Weitz JS, Stock CA, Wilhelm SW, Bourouiba L, Coleman ML, Buchan A, Follows MJ, Fuhrman JA, Jover LF, Lennon JT, Middelboe M, Sonderegger DL, Suttle CA, Taylor BP, Frede Thingstad T, Wilson WH, Eric Wommack K (2015) A multitrophic model to quantify the effects of marine viruses on microbial food webs and ecosystem processes. ISME J 9:1352–1364.

8 Appendices (theory)

8.1 Light harvesting

The net absorption of light by a cell is based on the density and distribution of individual chromophores within the cells cytoplasm. From geometry, it can be argued that the maximum possible absorption rate is $L\pi r^2$ where L [mol photon $m^{-2}s^{-1}$] is the incident light flux, and r is the cell radius. This corresponds to all incident photons being absorbed.

Cells however are not perfect absorbers of light. Photons are absorbed by chromophore imbedded within the cell, and variations in the density of (i.e. investment in) chromophores change the photosynthetic potential of the cell. For a spherical, uniformly distributed distribution of chromophores with a mean density c [μm^{-3}], the absorption rate W [mol photon s^{-1}] is given by

$$W(r,\lambda)=\frac{L\pi}{2\lambda^2}\left[(1+2\lambda r)e^{-2\lambda r}+(2\lambda^2 r^2-1)\right] \quad (4.11)$$

(Duyens 1956, Papageorgiou 1971, Kirk 1975) where $\lambda = ac$ [μm^{-1}] is the light extinction coefficient within the cytoplasm, a [μm^2] is the optical cross sectional of each chromophore, and c the number density within the cell.

In a diatom, and potentially in other cells, the distribution of plastids is not uniform, but concentrated within a layer close to the cell surface. For our spherical shell model of a vacuolated cell, where no plastids, and hence no absorption occurs in the vacuole, the absorption is:

$$W(r,\lambda)=2\pi r^2 L\int_0^1 \xi\left(1-e^{-2\lambda r\ell(r,v)}\right)d\xi \quad (C.2)$$

where:

$$\ell(\xi,v)=\mathrm{Re}\left[\sqrt{1-\xi^2}-\sqrt{v^2-\xi^2}\right] \quad (C.3)$$

(Papageorgiou 1971). While numerically tractable, this give an unwieldy analytic solution. A more accessible formulation can be got by assuming a similarity in the net absorption by a vacuolated spherical cell and an un-vacuolated cylinder of equal radius r and equivalent protoplasm carbon mass, the geometry of which is illustrated in Figure (4).

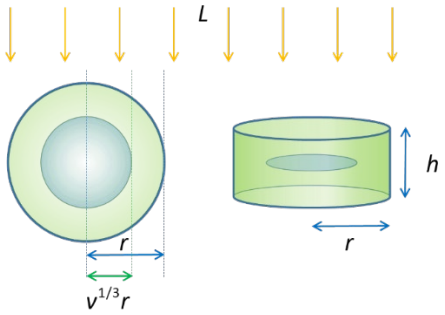


Figure 15: Approximating the light absorption characteristic of a vacuolated spherical cell, and a compressed cylinder of equivalent radius and volume. The optical path of a photon (and hence absorption) through the cylinder is much easier to estimate than for the vacuolated sphere.

Assuming the cell wall is negligible, the equivalent cylindrical volume is

$$\pi r^2 h = \frac{4\pi}{3} r^3 (1-v) \tag{4.12}$$

from which we get the cylinder height

$$h(r,v) = \frac{4}{3}(1-v)r \tag{4.13}$$

For a cylindrical arrangement of cytoplasm, with a uniform distribution of plastids, the total absorption of photons is given (almost by definition) as

$$W = \pi r^2 L\left[1-\exp(-\lambda h)\right] \tag{4.14}$$

where L [µE µm^{-2} s^{-1}] is the light intensity (PAR), and λ [µm^{-1}] is the light attenuation coefficient as defined above. From (Eq. A.6) we can deduce that h = 4 (1 – v) r / 3. This gives the following simple analytic expression for light affinity A_L [µmol C (µE µm^{-2})$^{-1}$] for a cell of radius r and a volume fraction v occupied by a central non-absorbing vacuole:

$$A_L = y\pi r^2\left[1-\exp\left(-\kappa\phi_L r\,(1-v)\right)\right] \tag{4.15}$$

where y is the photosynthetic quantum yield [µmol C µE^{-1}]. Investment in photosynthesis ϕ_L, can be lined linked to the light attenuation coefficient as:

$$\lambda = ac = (3/4)\ \kappa\phi_L \tag{4.16}$$

where an increase in investment, increases the concentration c of plastids; the return on investment is governed by the parameter κ [µm^{-1}]. Typical values (Raven 1984) suggest that λ = 0.1 µm^{-1} when about half of the cell's carbon is associated with photosynthesis. This suggests a value of κ = 0.2 µm^{-1}.

Finally, the fixed carbon flux to the cell J_L [µgC day^{-1}] from photosynthesis is then is given by:

$$J_L = yL\pi r^2\left[1-\exp\left(-\kappa\phi_L r\left(1-v\right)\right)\right] \tag{4.17}$$

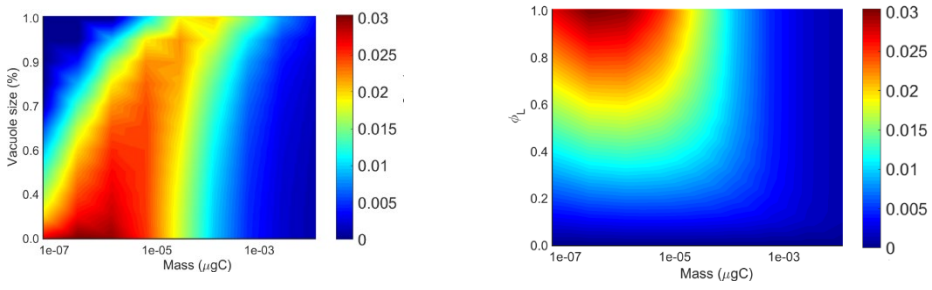


Figure 16: Specific light affinity (A_L/x; (µE m^{-2} s^{-1})$^{-1}$) as a function of carbon mass of the cytoplasm x (µg C) and (a) the vacuole size (v) for vacuolated cells (diatoms) or (b) investment in light (ϕ_L) for non-vacuolated cells (generalists).

9 Appendices (code)

9.1 Generalist-Diatom Model

The generalist-diatom model is the main component of the unicellular NUM model. It is coded with a MATLAB front end that provides a user interface for setting parameters, conducting post processing, and plotting results. The main subroutines however are written in FORTRAN90 so as to increase the efficiency of the program. The overall modelling strategy is to make the process modular, so as to enable future development.

9.1.1 Example MATLAB script

```
function sim = testGeneralistsDiatoms()
%%
% Set parameters:
%
       p = setupGeneralistsDiatoms;
       p = parametersChemostat(p);

%
% Calc derivatives:
%
       dudt = 0*ones(1,p.n);
       u = p.u0;
       [u, dudt] = calllib(loadNUMmodelLibrary(), 'f_calcderivatives', ...
              length(u), u, 60, 0.0, dudt);
       dudt
%
% Simulate
%
       tic
       p.tEnd = 200;
       sim = simulateChemostat(p, 60);
       toc
%
% Plot
%
     plotChemostat(sim)
```

This reads in parameters for the generalists and diatoms (e.g. size range, vacuolation, C:N ratios), and the physical environment. In this case a simple chemostat set up where light is constant, nitrogen and silicate are set by a deep water pool, and vertical exchange is set to a constant. The various routines needed for the calculation are loaded from the NUMmodel Library.

9.1.2 MATLAB code for the chemostat simulation is

```
%% Simulate the chemostat.
% In: p - parameter object obtained by calling a "setup" function followed
%       by a call to parametersChemostat; see e.g. the default value below.
%     L - Light
% Out:  sim - simulation object
%
       function sim = simulateChemostat(p, L)
       arguments
           p struct = parametersChemostat(setupGeneralistsOnly);
           L double = 100;
       end

%
% Find ix for nutrients and unicellulars:
%
       ix = 1:(p.idxB-1); % Nutrients
%
% Concentrations in the deep layer:
%
       uDeep = p.u0(ix);
%
% Simulate:
%
       [t,u] = ode23(@fDeriv, [0 p.tEnd], p.u0);
%
% Assemble result:
%
       sim.t = t;
       sim.N = u(:,p.idxN);
       sim.DOC = u(:,p.idxDOC);
       if isfield(p, 'idxSi')
           sim.Si = u(:,p.idxSi);
       end
       sim.B = u(:,p.idxB:end);
       sim.p = p;
       sim.rates = getRates(p, u(end,:), L);
       for iGroup = 1:p.nGroups
           sim.Bgroup(:,iGroup) = sum( u(:, p.ixStart(iGroup):p.ixEnd(iGroup)),2);
       end
 %
 % Function to assemble derivative for chemostat:
 %
    function dudt = fDeriv(t,u)
        dudt = 0*u';
        [u, dudt] = calllib(loadNUMmodelLibrary(), 'f_calcderivatives', ...
            length(u), u, L, 0.0, dudt);
        %
        % Chemostat dynamics for nutrients and unicellulars:
        %
        dudt(ix) = dudt(ix) + p.d*(uDeep-u(ix)');
        dudt = dudt';
    end
end
```

9.1.3 FORTRAN90 code NUMmodel

```
!
! Module to handle the NUM model framework
!
module NUMmodel
  use globals
  use spectrum
  use generalists
  use generalists_csp
  use diatoms
  use diatoms_simple
  use copepods
  use debug
  implicit none

  !
  ! Variables that contain the size spectrum groups
  !
  integer:: nGroups ! Number of groups
  integer:: iGroup ! Current number to be added (only used in addGroup)
  integer:: nNutrients ! Number of nutrient state variables
  integer:: idxB ! First index into non-nutrient groups (=nNutrients+1)
  type(typeSpectrum), dimension(:), allocatable:: group ! Structure for each group

  real(dp), dimension(:,:), allocatable:: theta ! Interaction matrix
  real(dp), dimension(:), allocatable:: palatability ! Palatability of each size group
  real(dp), dimension(:), allocatable:: upositive ! State variable constrained to be positive
  !
  ! Variables for HTL mortalities:
  !
  real(dp), dimension(:), allocatable:: pHTL ! Selectivity function for HTL mortality
  real(dp) :: mortHTL ! Level of HTL mortality (see below for override)
  real(dp):: gammaHTL ! Parameter for quadratic HTL mortality
  logical:: bQuadraticHTL ! Boolean flag to signify whether mortality is standard or "quadratic"
  ! Defaults to true; can be overridden but parameters must then be set with a
  ! call to parametersFinalize()

  type(typeRates):: rates

  real(dp), dimension(:), allocatable:: m, z, beta, sigma, AF, JFmax, epsilonF ! Feeding
parameters

contains

  ! ======================================
  !  Various model setups
  ! ======================================


  ! ----------------------------------------------
  ! A basic setup with only generalists
  ! ----------------------------------------------
  subroutine setupGeneralistsOnly(n)
    integer, intent(in):: n
    call parametersInit(1, n, 2) ! 1 group, n size classes (excl nutrients and DOC)
    call parametersAddGroup(typeGeneralist, n, 1.d0) ! generalists with n size classes
    bQuadraticHTL = .false. ! Use standard "linear" mortality
    call parametersFinalize()
  end subroutine setupGeneralistsOnly

  ! ----------------------------------------------
```

```
 ! A basic setup with only generalists -- Camila
 ! ------------------------------------------------
 subroutine setupGeneralistsOnly_csp()
   call parametersInit(1, 10, 2) ! 1 group, 10 size classes (excl nutrients and DOC)
   call parametersAddGroup(typeGeneralist_csp, 10, 0.1d0) ! generalists with 10 size classes
   call parametersFinalize()
 end subroutine setupGeneralistsOnly_csp

 ! ------------------------------------------------
 ! A basic setup with only diatoms:
 ! ------------------------------------------------
 subroutine setupDiatomsOnly(n)
  integer, intent(in):: n
  call parametersInit(1, n, 3) ! 1 group, n size classes (excl nutrients)
  call parametersAddGroup(typeDiatom, n, 1.d0) ! diatoms with n size classes
  bQuadraticHTL = .false. ! Use standard "linear" mortality
  call parametersFinalize()
end subroutine setupDiatomsOnly

! ------------------------------------------------
 ! A basic setup with only simple diatoms:
 ! ------------------------------------------------
subroutine setupDiatoms_simpleOnly(n)
  integer, intent(in):: n
  call parametersInit(1, n, 3) ! 1 group, n size classes (excl nutrients)
  call parametersAddGroup(typeDiatom_simple, n, 1.d0) ! diatoms with n size classes
  bQuadraticHTL = .false. ! Use standard "linear" mortality
  call parametersFinalize()
end subroutine setupDiatoms_simpleOnly

 ! ------------------------------------------------
 ! Generalists and diatoms:
 ! ------------------------------------------------
  subroutine setupGeneralistsDiatoms(n)
     integer, intent(in):: n
     call parametersInit(2, 2*n, 3)
     call parametersAddGroup(typeGeneralist, n, 1.d0) ! generalists with n size classes
     call parametersAddGroup(typeDiatom, n, 1.d0) ! diatoms with n size classes
     bQuadraticHTL = .false. ! Use standard "linear" mortality
     call parametersFinalize()
  end subroutine setupGeneralistsDiatoms

  subroutine setupGeneralistsDiatoms_simple(n)
     integer, intent(in):: n
     call parametersInit(2, 2*n, 3)
     call parametersAddGroup(typeGeneralist, n, 1.d0) ! generalists with n size classes
     call parametersAddGroup(typeDiatom_simple, n, 1.d0) ! diatoms with n size classes
     bQuadraticHTL = .false. ! Use standard "linear" mortality
     call parametersFinalize()
  end subroutine setupGeneralistsDiatoms_simple

 ! ------------------------------------------------
 ! A basic setup with generalists and 1 copepod
 ! ------------------------------------------------
 subroutine setupGeneralistsCopepod()
   call parametersInit(2, 20, 2)
   call parametersAddGroup(typeGeneralist, 10, 0.1d0)
   call parametersAddGroup(typeCopepod, 10, .1d0) ! add copepod with adult mass .1 mugC
   call parametersFinalize()
 end subroutine setupGeneralistsCopepod

 ! ------------------------------------------------
 ! A generic setup with generalists and a number of copepod species
```

```
! ------------------------------------------------
subroutine setupGeneric(mAdult)
  real(dp), intent(in):: mAdult(:)
  integer, parameter:: n = 10 ! number of size classes in each group
  integer:: iCopepod

  call parametersInit(size(mAdult)+1, n*(size(mAdult)+1), 2)
  call parametersAddGroup(typeGeneralist, n, 0.1d0)
  if ( size(mAdult) .eq. 0) then
     bQuadraticHTL = .false. ! Use standard "linear" mortality
  else
     do iCopepod = 1, size(mAdult)
        call parametersAddGroup(typeCopepod, n, mAdult(iCopepod)) ! add copepod
     end do
  end if
  call parametersFinalize()
end subroutine setupGeneric

! ------------------------------------------------
! A generic setup with generalists and a number of copepod species
! ------------------------------------------------
subroutine setupGeneric_csp(mAdult)
  real(dp), intent(in):: mAdult(:)
  integer, parameter:: n = 10 ! number of size classes in each group
  integer:: iCopepod

  call parametersInit(size(mAdult)+1, n*(size(mAdult)+1), 2)
  call parametersAddGroup(typeGeneralist_csp, n, 0.1d0)
  do iCopepod = 1, size(mAdult)
     call parametersAddGroup(typeCopepod, n, mAdult(iCopepod)) ! add copepod
  end do
  call parametersFinalize()
end subroutine setupGeneric_csp


! ====================================
!  Model initialization stuff:
! ====================================

! ------------------------------------------------
! Initialize parameters
! In:
!    nnGroups: number of size spectrum groups
!    nnGrid: total length of the grid (excl nnNutrients points for N, DOC, etc.)
! ------------------------------------------------
subroutine parametersInit(nnGroups, nnGrid, nnNutrients)
  integer, intent(in):: nnGrid, nnGroups, nnNutrients
  !
  ! Set groups:
  !
  nGroups = nnGroups
  nNutrients = nnNutrients
  nGrid = nnGrid+nnNutrients
  idxB = nNutrients + 1
  iGroup = 0
  !
  ! Allocate variables:
  !
  if (allocated(m)) then
      <<Deallocate variables>>
  end if

  <<Allocate variables>>
```

```
  bQuadraticHTL = .true. ! Default to quadratic mortality
end subroutine parametersInit
! ------------------------------------------------
!  Add a size spectrum group
!  In:
!    typeGroup: the group type (see definitions in Globals.f90
!    n: number of grid points
!    mMax: the maximum size (mid-point in grid cell)
! ------------------------------------------------
subroutine parametersAddGroup(typeGroup, n, mMax)
  integer, intent(in):: typeGroup, n
  real(dp), intent(in):: mMax
  !
  ! Find the group number and grid location:
  !
  iGroup = iGroup + 1
  if (iGroup.eq.1) then
     group(iGroup)%ixStart = idxB
  else
     group(iGroup)%ixStart = group(iGroup-1)%ixEnd+1
  end if
  group(iGroup)%ixEnd = group(iGroup)%ixStart+n-1

  if (group(iGroup)%ixEnd .gt. nGrid) then
     write(6,*) 'Attempting to add more grid points than allocated', &
         group(iGroup)%ixEnd, nGrid
     stop 1
  end if
  !
  ! Add the group
  !
  select case (typeGroup)
  case (typeGeneralist)
    group(iGroup) = initGeneralists(n, group(iGroup)%ixStart-1, mMax)
  case (typeGeneralist_csp)
    group(iGroup) = initGeneralists_csp(n, group(iGroup)%ixStart-1, mMax)
  case (typeDiatom)
    group(iGroup) = initDiatoms(n, group(iGroup)%ixStart-1, mMax)
    palatability(group(iGroup)%ixStart:group(iGroup)%ixEnd) = 0.5d0 ! palatability for diatoms
  case (typeDiatom_simple)
    group(iGroup) = initDiatoms_simple(n, group(iGroup)%ixStart-1, mMax)
    palatability(group(iGroup)%ixStart:group(iGroup)%ixEnd) = 0.5d0 ! palatability for diatoms
  case(typeCopepod)
     group(iGroup) = initCopepod(n, group(iGroup)%ixStart-1, mMax)
  end select
  group(iGroup)%type = typeGroup
  !
  ! Import grid to globals:
  !
  m(group(iGroup)%ixStart : group(iGroup)%ixEnd) = group(iGroup)%m
  z(group(iGroup)%ixStart : group(iGroup)%ixEnd) = group(iGroup)%z
  !
  ! Import feeding parameters:
  !
  beta(group(iGroup)%ixStart : group(iGroup)%ixEnd) = group(iGroup)%beta
  sigma(group(iGroup)%ixStart : group(iGroup)%ixEnd) = group(iGroup)%sigma
  AF(group(iGroup)%ixStart : group(iGroup)%ixEnd) = group(iGroup)%AF
  JFmax(group(iGroup)%ixStart : group(iGroup)%ixEnd) = group(iGroup)%JFmax
  epsilonF(group(iGroup)%ixStart : group(iGroup)%ixEnd) = group(iGroup)%epsilonF

end subroutine parametersAddGroup
! ------------------------------------------------
```

```
!  Finalize the setting of parameters. Must be called when
!  all groups have been added.
! ------------------------------------------------
subroutine parametersFinalize()
  integer:: i,j
  real(dp):: betaHTL, mHTL, mMax
  !
  ! Calc theta:
  !
  do i = idxB, nGrid
     do j = idxB, nGrid
        !theta(i,j) = exp( -(log(m(i)/m(j)/beta(i)))**2/(2*sigma(i)**2))
        theta(i,j) = palatability(j) * calcPhi(m(i)/m(j), beta(i), sigma(i), z(i))
     end do
  end do
  !
  ! Calc htl mortality
  !
  betaHTL = 500.
  if (.not. bQuadraticHTL) then
     !
     ! Standard HTL mortality. In this case "pHTL" represent the selectivity of HTL mortality
     !
     mHTL = m(nGrid)/betaHTL**1.5  ! Bins affected by HTL mortality
     mortHTL = 0.2
     pHTL(idxB:nGrid) = (1/(1+(m(idxB:nGrid)/mHTL)**(-2)))
  else
     !
     ! Linear HTL mortality (commonly referred to as "quadratic")
     ! In this case "pHTL" represents p_HTL*m^-1/4 from eq (16) in Serra-Pompei (2020)
     !
     mortHTL = 0.003 ! Serra-Pompei (2020)
     gammaHTL = 0.2 ! ibid
     mMax = maxval(m(idxB:nGrid))
     do i=idxB, nGrid
        if (m(i) .lt. (mMax/betaHTL)) then
           pHTL(i) = exp( -(log(m(i)*betaHTL/mMax))**2/4.)
        else
           pHTL(i) = 1.
        end if
     end do
     pHTL(idxB:nGrid) = pHTL(idxB:nGrid) * m(idxB:nGrid)**(-0.25)
  end if
  !write(6,*) m,pHTL

contains
  !
  ! Calculate the interaction coefficient between two size groups.
  ! In:
  !   z : The predator:prey body mass ratio between the two groups
  !   beta: preferred predator:prey body mass ratio
  !   sigma: width of selection
  !   Delta: ratio between upper and lower body mass in size groups
  !
  function calcPhi(z, beta,sigma, Delta) result(res)
    real(dp), intent(in):: z,beta,sigma,Delta
    real(dp):: res, s

    if (beta .eq. 0.d0) then
       res = 0.d0 ! If beta = 0 it is interpreted as if the group is not feeding
    else
       s = 2*sigma*sigma
       res = max(0.d0, &
```

```
        (Sqrt(Delta)*(((exp(-Log((beta*Delta)/z)**2/s) - 2/exp(Log(z/beta)**2/s) + &
        exp(-Log((Delta*z)/beta)**2/s))*s)/2. - &
        (Sqrt(Pi)*Sqrt(s)*(Erf((-Log(beta*Delta) + Log(z))/Sqrt(s))*Log((beta*Delta)/z) + &
        2*Erf(Log(z/beta)/Sqrt(s))*Log(z/beta) + &
        Erf((Log(beta) - Log(Delta*z))/Sqrt(s))*Log((Delta*z)/beta)))/2.))/ &
        ((-1 + Delta)*Log(Delta)) )
      end if
    end function calcPhi

  end subroutine parametersFinalize

  ! ====================================
  !  Calculate rates and derivatives:
  ! ====================================

  !
  ! Calculate derivatives for unicellular groups
  ! In:
  !   gammaN and gammaDOC are reduction factors [0...1] of uptakes of N and DOC,
  !   used for correction of Euler integration. If no correction is used, just set to 1.0
  !   This correction procedure is needed for correct Euler integration.
  subroutine calcDerivativesUnicellulars(upositive, L, gammaN, gammaDOC, gammaSi)
    real(dp), intent(in):: upositive(:), L, gammaN, gammaDOC, gammaSi
    !type(typeRates), intent(inout):: rates
    integer:: i,j
    !
    ! Calc uptakes of all unicellular groups:
    !
    do iGroup = 1, nGroups
       select case (group(iGroup)%type)
       case (typeGeneralist)
          call calcRatesGeneralists(group(iGroup), &
               rates, L, upositive(idxN), upositive(idxDOC), gammaN, gammaDOC)
       case(typeGeneralist_csp)
          call calcRatesGeneralists_csp(group(iGroup), &
               rates, L, upositive(idxN),  gammaN)
         case(typeDiatom)
            call calcRatesDiatoms(group(iGroup), &
            rates, L, upositive(idxN), upositive(idxDOC) , upositive(idxSi), &
            gammaN, gammaDOC, gammaSi)
         case(typeDiatom_simple)
            call calcRatesDiatoms_simple(group(iGroup), &
            rates, L, upositive(idxN), upositive(idxDOC) , upositive(idxSi), &
            gammaN, gammaDOC, gammaSi)
       end select
    end do
    !
    ! Calc predation mortality
    !
    do i=idxB, nGrid
       rates%mortpred(i) = 0.d0
       do j=idxB, nGrid
          if (rates%F(j) .ne. 0.d0) then
             rates%mortpred(i) = rates%mortpred(i)  &
                  + theta(j,i) * rates%JF(j)*upositive(j)/(epsilonF(j)*m(j)*rates%F(j))
          end if
       end do
    end do
    !
    ! Assemble derivatives:
    !
    rates%dudt(1:(idxB-1)) = 0.d0 ! Set derivatives of nutrients to zero
```

```
    do iGroup = 1, nGroups
       select case (group(iGroup)%type)
       case (typeGeneralist)
          call calcDerivativesGeneralists(group(iGroup),&
               upositive(group(iGroup)%ixStart:group(iGroup)%ixEnd), &
               rates)
       case (typeGeneralist_csp)
          call calcDerivativesGeneralists_csp(group(iGroup),&
               upositive(group(iGroup)%ixStart:group(iGroup)%ixEnd), &
               rates)
          case (typeDiatom)
               call calcDerivativesDiatoms(group(iGroup),&
               upositive(group(iGroup)%ixStart:group(iGroup)%ixEnd), &
               rates)
        case (typeDiatom_simple)
               call calcDerivativesDiatoms_simple(group(iGroup),&
               upositive(group(iGroup)%ixStart:group(iGroup)%ixEnd), &
               rates)

       end select
    end do
  end subroutine calcDerivativesUnicellulars

  ! ------------------------------------------------
  !  Calculate the derivatives for all groups:
  !  In:
  !    L: light level
  ! ------------------------------------------------
  subroutine calcDerivatives(u, L, dt)
    real(dp), intent(in):: L, dt, u(:)
    integer:: i, j, iGroup
    real(dp):: gammaN, gammaDOC, gammaSi

    !
    ! Use only the positive part of biomasses for calculation of derivatives:
    !
    do i = 1, nGrid
       upositive(i) = max( 0.d0, u(i) )
    end do
    !
    ! Calc uptakes of food
    !
    do i = idxB, nGrid
       rates%F(i) = 0.d0
       do j = idxB, nGrid
          rates%F(i) = rates%F(i) + theta(i,j)*upositive(j)
       end do
    end do

    rates%flvl(idxB:nGrid) = AF(idxB:nGrid)*rates%F(idxB:nGrid) /
    (AF(idxB:nGrid)*rates%F(idxB:nGrid) + JFmax(idxB:nGrid))
    rates%JF(idxB:nGrid) = rates%flvl(idxB:nGrid) * JFmax(idxB:nGrid)
    !
    ! Calc HTL mortality:
    !
    if (bQuadraticHTL) then
       do i = idxB, nGrid
          rates%mortHTL(i) = calcHTL(upositive, i)*upositive(i)
       end do
    else
       rates%mortHTL(idxB:nGrid) = mortHTL*pHTL(idxB:nGrid)
    end if
    !
```

```
    ! Calc derivatives of unicellular groups
    !
    gammaN = 1.d0
    gammaDOC = 1.d0
    gammaSi = 1.d0

    call calcDerivativesUnicellulars(upositive, L, gammaN, gammaDOC, gammaSi)
    !
    ! Make a correction if nutrient fields will become less than zero:
    !
    if ((u(idxN) + rates%dudt(idxN)*dt) .lt. 0) then
       gammaN = max(0.d0, min(1.d0, -u(idxN)/(rates%dudt(idxN)*dt)))
    end if
    if ((u(idxDOC) + rates%dudt(idxDOC)*dt) .lt. 0) then
       gammaDOC = max(0.d0, min(1.d0, -u(idxDOC)/(rates%dudt(idxDOC)*dt)))
    end if
    if ((u(idxSi) + rates%dudt(idxSi)*dt) .lt. 0) then
      gammaSi = max(0.d0, min(1.d0, -u(idxSi)/(rates%dudt(idxSi)*dt)))
    end if
    if ((gammaN .lt. 1.d0) .or. (gammaDOC .lt. 1.d0) .or. (gammaSi .lt. 1.d0)) then
       call calcDerivativesUnicellulars(upositive, L, gammaN, gammaDOC, gammaSi)
    end if
    !
    ! Calc derivatives of multicellular groups:
    !
    do iGroup = 1, nGroups
      if (group(iGroup)%type .eq. typeCopepod) then
         call calcDerivativesCopepod(group(iGroup), &
            upositive(group(iGroup)%ixStart:group(iGroup)%ixEnd), &
            rates)
      end if
    end do
  end subroutine calcDerivatives
  !
  ! Returns the htl mortality divided by h for size bin i
  !
  function calcHTL(u, i) result(mHTL)
    real(dp) :: mHTL, B
    real(dp), intent(in):: u(:)
    integer, intent(in):: i
    integer:: j

    !
    ! First calc the biomass within a size range:
    !
    B = 0.d0
    do j = 1, nGrid
       if (m(j)>m(i)/3.16 .and. m(j)<m(i)*3.16) then
          B = B + u(j)
       end if
    end do

    mHTL = pHTL(i)*mortHTL/z(i)*u(i)**gammaHTL*B**(1.-gammaHTL)
  end function calcHTL

  ! ======================================
  !  Simulate models:
  ! ======================================

  ! ----------------------------------------------
  ! Simulate a chemostat with Euler integration
  ! ----------------------------------------------
  subroutine simulateChemostatEuler(u, L, Ndeep, diff, tEnd, dt)
```

```
  real(dp), intent(inout):: u(:) ! Initial conditions and result after integration
  real(dp), intent(in):: L      ! Light level
  real(dp), intent(in):: Ndeep ! Nutrient in the deep layer
  real(dp), intent(in):: SIdeep ! Silicate in the deep layer
  real(dp), intent(in):: diff      ! Diffusivity
  real(dp), intent(in):: tEnd ! Time to simulate
  real(dp), intent(in):: dt    ! time step
  integer:: i, iEnd

  iEnd = floor(tEnd/dt)

  do i=1, iEnd
     call calcDerivatives(u, L, dt)
     rates%dudt(idxN) = rates%dudt(idxN) + diff*(Ndeep-u(idxN))
     rates%dudt(idxDOC) = rates%dudt(idxDOC) + diff*(0.d0 - u(idxDOC))
     !if (idxB .gt. idxSi) then
     !  rates%dudt(idxSi) = rates%dudt(idxSi) + diff*(SIdeep - u(idxSi))
     !end if
     !
     ! Note: should not be done for copepods:
     !
     rates%dudt(idxB:nGrid) = rates%dudt(idxB:nGrid) + diff*(0.d0 - u(idxB:nGrid))
     u = u + rates%dudt*dt
     !write(6,*) calcN(u)
  end do
end subroutine simulateChemostatEuler

function calcN(u) result(N)
  real(dp), intent(in):: u(:)
  integer:: i
  real(dp):: N

  N = 0
  N = u(idxN)
  do i = 1, nGrid
     N = N + u(nNutrients+i)/5.68
  end do
end function calcN

! ------------------------------------------------
! Simulate with Euler integration
! ------------------------------------------------
subroutine simulateEuler(u, L, tEnd, dt)
  real(dp), intent(inout):: u(:) ! Initial conditions and result after integration
  real(dp), intent(in):: L      ! Light level
  real(dp), intent(in):: tEnd ! Time to simulate
  real(dp), intent(in):: dt    ! time step
  integer:: i, iEnd

  iEnd = floor(tEnd/dt)

  do i=1, iEnd
     call calcDerivatives(u, L, dt)
     u = u + rates%dudt*dt
  end do
end subroutine simulateEuler
! ------------------------------------------------
! Temperature Q10 function
! ------------------------------------------------
function fTemp(Q10, T) result(f)
  real(dp), intent(in), value:: Q10, T
  real(dp):: f
```

```
  f = Q10**(T/10.-1.)
end function fTemp

!=======================================
! Diagnostic functions
!=======================================

subroutine getMass(m_, mDelta)
 real(dp), intent(inout):: m_(nGrid), mDelta(nGrid)
 integer:: i

 m_ = m;
 do i = 1,nGroups
    mDelta(group(i)%ixStart:group(i)%ixEnd) = group(i)%mDelta
 end do
 end subroutine getMass

! ------------------------------------------------
! Get the ecosystem functions as calculated from the last call
! to calcDerivatives
! ------------------------------------------------
subroutine getFunctions(ProdGross, ProdNet,ProdHTL,eHTL,Bpico,Bnano,Bmicro)
  real(dp), intent(out):: ProdGross, ProdNet,ProdHTL,eHTL,Bpico,Bnano,Bmicro
  real(dp) :: conversion
  real(dp) :: ESD(nGrid)
  integer:: i

  ProdGross = 0.d0
  ProdNet = 0.d0
  ProdHTL = 0.d0
  Bpico = 0.d0
  Bnano = 0.d0
  Bmicro = 0.d0

  conversion = 365.*1d-6*1000. ! Convert to gC/yr/m3
  do i = 1, nGroups
     if (group(i)%type .eq. typeGeneralist) then
        ProdGross = ProdGross + conversion * &
        sum(  rates%JL(idxB:nGrid) * upositive(idxB:nGrid) / m(idxB:nGrid) )
        ProdNet = ProdNet + conversion * &
        getProdNetGeneralists(group(i),  upositive(group(i)%ixStart:group(i)%ixEnd), rates)
     end if
  end do

  ESD = 10000. * 1.5 * (m*1d-6)**onethird
  conversion = 1d-6*1000 ! Convert to gC/m3
  do i = idxB, nGrid
     if (ESD(i) .le. 2.) then
        Bpico = Bpico + conversion*upositive(i)
     endif

     if ((ESD(i).gt.2.) .and. (ESD(i) .le. 20.)) then
        Bnano = Bnano + conversion*upositive(i)
     endif

     if (ESD(i) .gt. 20.) then
        Bmicro = Bmicro + conversion*upositive(i)
     endif

     ProdHTL = ProdHTL + 365*conversion*rates%mortHTL(i)*upositive(i)
  end do

  eHTL = eHTL / ProdNet
```

```
    if (eHTL .gt. 1) then
       eHTL = -1.
    end if
  end subroutine getFunctions

  ! ------------------------------------------------
  ! Returns the rates calculated from last call to calcDerivatives
  ! ------------------------------------------------
  subroutine getRates(jN, jDOC, jL, jSi, jF, jFreal,&
    jTot, jMax, jFmaxx, jR, jLossPassive, &
    jNloss,jLreal, &
    mortpred, mortHTL, mort2, mort)
    use globals
    real(dp), intent(out):: jN(nGrid-nNutrients), jDOC(nGrid-nNutrients), jL(nGrid-nNutrients)
    real(dp), intent(out):: jSi(nGrid-nNutrients)
    real(dp), intent(out):: jF(nGrid-nNutrients), jFreal(nGrid-nNutrients)
    real(dp), intent(out):: jTot(nGrid-nNutrients), jMax(nGrid-nNutrients), jFmaxx(nGrid-
nNutrients)
    real(dp), intent(out):: jR(nGrid-nNutrients)
    real(dp), intent(out):: jLossPassive(nGrid-nNutrients), jNloss(nGrid-nNutrients),
jLreal(nGrid-nNutrients)
    real(dp), intent(out):: mortpred(nGrid-nNutrients), mortHTL(nGrid-nNutrients)
    real(dp), intent(out):: mort2(nGrid-nNutrients), mort(nGrid-nNutrients)

    jN = rates%JN(idxB:nGrid) / m(idxB:nGrid)
    jDOC = rates%JDOC(idxB:nGrid) / m(idxB:nGrid)
    jL = rates%JL(idxB:nGrid) / m(idxB:nGrid)
    jSi = rates%JSi(idxB:nGrid) / m(idxB:nGrid)
    jF = rates%flvl(idxB:nGrid) * JFmax(idxB:nGrid)/ m(idxB:nGrid)
    jFreal = rates%JF(idxB:nGrid) / m(idxB:nGrid)
    jTot = rates%Jtot(idxB:nGrid) / m(idxB:nGrid)
    jMax = 1.5 + 0*m(idxB:nGrid) ! NOTE: HARDCODED. Should be taken from generalists
    jFmaxx = JFmax(idxB:nGrid) / m(idxB:nGrid)
    jR = 1.5*0.1 + 0*m(idxB:nGrid)  ! NOTE: HARDCODED. Should be taken from generalists
    jLossPassive = 0* m(idxB:nGrid)  ! NOTE: HARDCODED. Should be taken from generalists
    jNloss = rates%JNloss(idxB:nGrid) / m(idxB:nGrid)
    jLreal = rates%JLreal(idxB:nGrid) / m(idxB:nGrid)
    mortpred = rates%mortpred(idxB:nGrid)
    mortHTL = rates%mortHTL(idxB:nGrid)
    mort2 = 0.0002*(nGrid-nNutrients)*upositive(idxB:nGrid)  ! NOTE: HARDCODED. Should be taken
from generalists
    mort = 0*m(idxB:nGrid)  ! NOTE: HARDCODED. Should be taken from generalists
  end subroutine getRates

end module NUMmodel
```

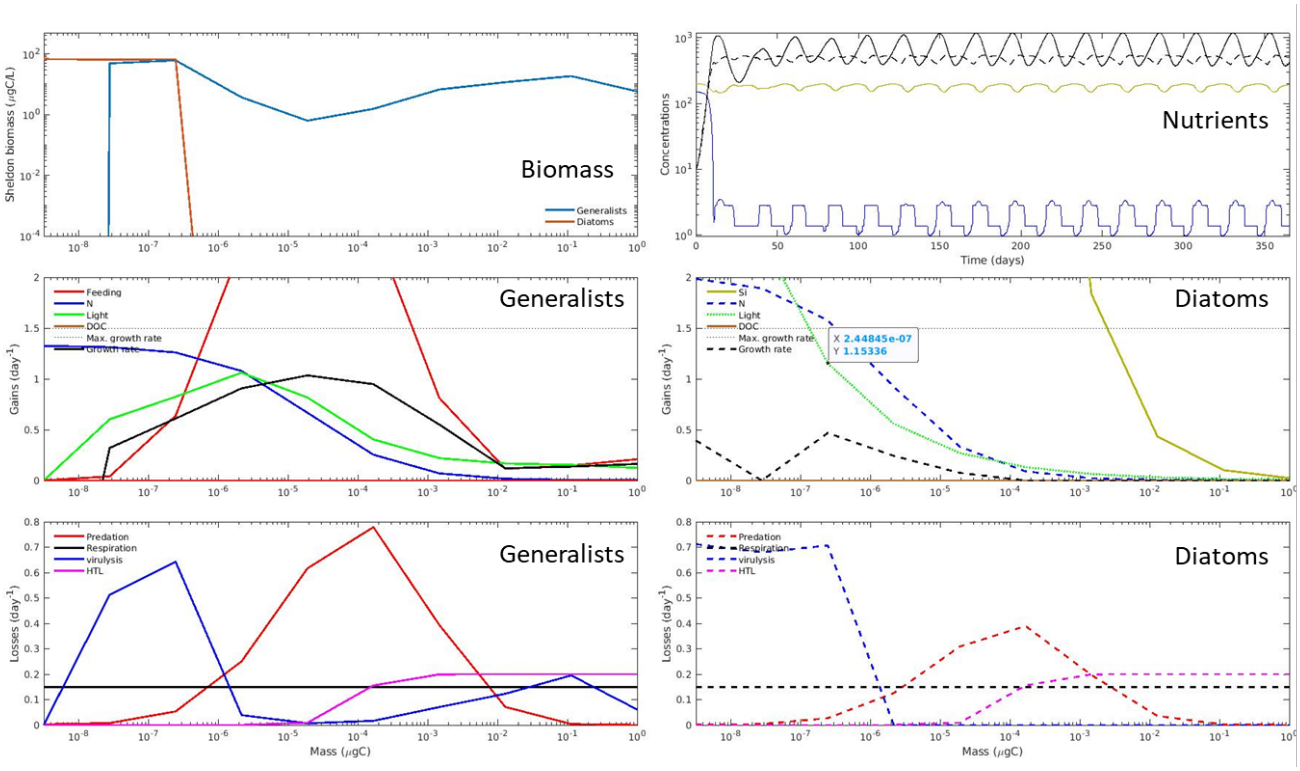


Figure 17: Typical output form the unicellular NUM model for a chemostat setting.

9.2 Diatom Model

```
%% File that runs model Based on size non specific liebig full
close all
clear
res       = 100;                    % model resolution

%% Environmental timeseries
L  = 70; % W/m2 at 10 meters depth
N  = 40; % µg N/l
Si = 30; % µg Si/l [0.00494 0.00451 0.00367 0.00238 0.00083 0.00044 0.00040 0.00073 0.00167 0.00290 0.00403
0.00427]; % mol Si/m3
G  = 15; % µg C/l

% trait space
c         = logspace(-7,1,res); % carbon size range
c         = repmat(c,res,1);    % matrix of radius (v,c)
vfrac     = linspace(0,1,res);  % range of vacuole fraction
vfrac     = repmat(vfrac',[1 res]);

%
ccb_g     = 0.125*10^-6;        % µg C µm-3 (ONLY BIOMASS)
Vb        = c./ccb_g;           % biovolume
r0        = (Vb./(4/3*pi)).^(1/3); % equvalent spherical radius of biovolume (µm)
m         = 8*10^-3;            % membrane thickness (µm)
% equvalent spherical radius (including outer membrane, but no shell)
ESR       = (m^2./(vfrac - 1) + m^2./(vfrac - 1).^2)./((((3*m^3)./(2*(vfrac - 1).^2) + m^3./(vfrac - 1).^3 +
(m^3 + r0.^3)./(2*vfrac - 2)).^2 - (m^2./(vfrac - 1) + m^2./(vfrac - 1).^2).^3).^(1/2) - (3*m^3)./(2*(vfrac
- 1).^2) - m^3./(vfrac - 1).^3 - (m^3 + r0.^3)./(2*vfrac - 2)).^(1/3) - m./(vfrac - 1) +
((((3*m^3)./(2*(vfrac - 1).^2) + m^3./(vfrac - 1).^3 + (m^3 + r0.^3)./(2*(vfrac - 1))).^2 - (m^2./(vfrac -
1) + m^2./(vfrac - 1).^2).^3).^(1/2) - m^3./(vfrac - 1).^3 - (m^3 + r0.^3)./(2*(vfrac - 1)) -
(3*m^3)./(2*(vfrac - 1).^2)).^(1/3);
ESR(1,:)  = r0(1,:)+m;
V0        = 4/3*pi*ESR.^3;      % total volume (outer membrane, no shell)
%V0(1,:)   = 0;
Vmem      = 4/3*pi*(ESR).^3-4/3*pi*(ESR-m).^3; % outer (plasmalemma) membrane volume
Vv        = V0 - Vb - Vmem;     % Vacuole volume (inkluding inner totnoplast membrane)
Vv(1,:)   = 0;
rv        = (Vv./(4/3*pi)).^(1/3); % radius of vacuole
t         = (10.^(0.24*log10(V0)+0.21))*10^-3; % Shell thickness (µm)
Vs        = (4/3*pi*(ESR+t).^3)-4/3*pi*(ESR).^3;% shell volume (µm3)
RNC       = 1/7;                % (mol N (mol C)-1 )
RCN       = 7;                  % (mol C (mol N)-1 )
RCN_g     = RCN*12/14;          % (g C (g N)-1 )
RNC_g     = RNC*14/12;          % (g N (g C)-1 )
Ncyto     = c*RNC_g;            % µg N/cell cytoplasm (no tonoplast)
Vtono     = 4/3*pi*(rv).^3-4/3*pi*(rv-m).^3; % volume of (inner) tonoplast membrane

% densities
rhoW      = 1027;               % Density of seawater (kg/m3)
rhoB      = 1200;               % Density of biomass (kg/m3) SA
rhoS      = 2600;               % Density of shell (kg/m3) (max) SA
rhoV      = 1010;               % Density of vacuole (kg/m3)
% diatom overall density (r,v)
rhoD      = rhoS*Vs./(V0+Vs)+rhoV*Vv./(V0+Vs)+rhoB*(Vb+Vtono+Vmem)./(V0+Vs);

rhoCtono  = 600*10^9/10^18;     % carbon density of tonoplast µg C/µm3
rhoCmoltono = rhoCtono/12;      % µmol/µm3
ctono     = Vtono*rhoCtono;     % carbon in tonoplast µg C
cmem      = Vmem*rhoCtono;      % carbon in plasmalemma µg C
Cmoltono  = Vtono*rhoCmoltono;  % carbon in tonoplast µmol C
Cmolmem   = Vmem*rhoCmoltono;   % carbon in plasmalemma µmol C
CNtono    = 48;                 % CN ratio in membrane material (mol/mol)
Nmoltono  = Cmoltono./CNtono;   % Nitrogen in tonoplast µmol N
Nmolmem   = Cmolmem./CNtono;    % Nitrogen in plasmalemma µmol N
Ntono     = Nmoltono*14;        % Nitrogen in tonoplast µg N
Nmem      = Nmolmem*14;         % Nitrogen in plasmalemma µg N
Ntot      = Ncyto+Ntono+Nmem;   % Total nitrogen in cell (µg N cell-1)
Ctot      = c+ctono+cmem;       % Total carbon in cell (µg N cell-1)
```

```
membranecarb = (ctono+cmem)./Ctot; % fraction of carbon in membrane
CNtot     = Ctot./Ntot;          % effective CN ratio µg/µg
CNtot_mol = CNtot*14/12;         % mol/mol
ccb       = ccb_g/12*10^-6;      % (mol C µm-3) (ONLY BIOMASS)
ccbtot_mol= (Ctot)./(Vb+Vtono+Vmem)/12*10^-6; % mol C (µm3 biomass and membrane)-1
gammaSi   = rhoS*10^3/60.08*10^-18;% mole Si (µm3 shell)-1
K         = gammaSi./(ccbtot_mol); % mol Si per µm3 shell per mol C per µm3 biomass (GENOVEVEJ)
RSiC      = K.*Vs./(Vb+Vtono+Vmem);% mol Si (mol C)-1
RCSi      = 1./RSiC;             % mol C (mol Si)-1
RCSi_g    = RCSi*28/12;          % g C (g Si)-1
jmax      = 1;                   % maximum synthesis rate (day)-1
Jmax      = jmax*(c);            % maximum synthesis (µg C (day)-1)


% grazing
fG        = 0.09;
m0        = 0.05;

% affinities

r_meter   = (ESR)*10^-6;         % radius in meters
DN        = 1.88*10^-5*10^-4*60*60*24;% Nitrogen diffusion (m2/day) (1.62*10^-4)
DSi       = 1.88*10^-5*10^-4*60*60*24;% Silicon diffusion (m2/day) (1.62*10^-4)
cL        = 0.01*(ccb_g)^(2/3); % µg C /day /(W/m2)/µm3^(2/3))
AL        = cL.*(V0).^(2/3);     % µg C/day/(W/m2) (r,v)
ALspec    = AL./c;               % specific affiliny for light (1/day/(W/m2))
AN        = 4*pi*DN*(r_meter)*10^3;% Nitrogen affinity (l/day)
ASi       = 4*pi*DSi*(r_meter)*10^3;% Nitrogen affinity (l/day)

resp0     = 0.05;                % respiration rate (day-1)
Jr0       = resp0.*(Ctot);       % µg C/day

% costs
betaL     = 0.35;                % Cost of photosynthetic uptake and synthesis (µg C (µg C)-1)
betaN     = 3;                   % Cost of nitrogen uptake and synthesis (µg C (µg N)-1) SAT NED FRA 3.5
betaSi    = 1.05;                % Cost of silicification (µg C (µg Si)-1)


% sinking mortality
grav      = 9.80665;             % tyngdeaccelerationen (m s-2)
nu        = 1.07e-3;             % dynamic viscosity of water (Pa s)
H         = 20;                  % Depth of euphotic zone.
% sinking speed (m s-1)
w         = 2*grav*(10^-6*(V0./(4/3*pi)).^(1/3)).^2/(9*nu).*(rhoD-rhoW);
ms        = w./H.*60*60*24;      % sinkin mortality (day^-1)
ms(ms<0) = 0;

%% Find optimal growth in trait space
for n     = 1:length(L)

    % Division rate
    JN(:,:,n)       = AN*N(n);  % µg C/day
    JSi(:,:,n)      = ASi*Si(n);% µg Si/day
    JL(:,:,n)       = AL*L(n);  % µg N/day

    % pC = 1
    pN1(:,:,n)      = max(0,min(-(Jr0.*RCSi_g - JL(:,:,n).*RCSi_g +
JL(:,:,n).*RCSi_g*betaL)./(JN(:,:,n).*(CNtot*betaSi + RCSi_g*betaN + CNtot.*RCSi_g)),1));
    pSi1(:,:,n)     = max(0,min(-(CNtot.*Jr0 - CNtot.*JL(:,:,n) +
CNtot.*JL(:,:,n)*betaL)./(JSi(:,:,n).*(CNtot*betaSi + RCSi_g*betaN + CNtot.*RCSi_g)),1));
    % pN = 1
    pSi2(:,:,n)     = max(0,min((CNtot.*JN(:,:,n))./(JSi(:,:,n).*RCSi_g),1));
    pC2(:,:,n)      = max(0,(Jr0.*RCSi_g + CNtot.*JN(:,:,n).*RCSi_g + CNtot.*JN(:,:,n)*betaSi +
JN(:,:,n).*RCSi_g*betaN)./(RCSi_g.*(JL(:,:,n) - JL(:,:,n)*betaL)));
    % pSi = 1
    pN3(:,:,n)      = max(0,min((JSi(:,:,n).*RCSi_g)./(JN(:,:,n).*CNtot),1));
    pC3(:,:,n)      = max(0,(CNtot.*Jr0 + CNtot.*JSi(:,:,n).*RCSi_g + CNtot.*JSi(:,:,n)*betaSi +
JSi(:,:,n).*RCSi_g*betaN)./(CNtot.*(JL(:,:,n) - JL(:,:,n).*betaL)));

    % loop through all carbon (v) and vacuole (rad) sizes
    for rad=1:res
```

```
        for v=1:res
            if pN1(rad,v,n)<1 && pSi1(rad,v,n)<1
                pN(rad,v,n)      = pN1(rad,v,n);
                pSi(rad,v,n)     = pSi1(rad,v,n);
                pC(rad,v,n)      = 1;

                limmat(rad,v,n) =1;

            elseif pSi2(rad,v,n)<1 && pC2(rad,v,n)<1
                pSi(rad,v,n)     = pSi2(rad,v,n);
                pN(rad,v,n)      = 1;
                pC(rad,v,n)      = pC2(rad,v,n);

                limmat(rad,v,n) =2;

            elseif pN3(rad,v,n)<1 && pC3(rad,v,n)<1
                pSi(rad,v,n)     = 1;
                pN(rad,v,n)      = pN3(rad,v,n);
                pC(rad,v,n)      = pC3(rad,v,n);

                limmat(rad,v,n) =3;

            end
            pC(res,:,n)          = NaN;
            pSi(res,:,n)         = NaN;
            pN(res,:,n)          = NaN;
            Jlieb(rad,v,n)       = max(min([JL(rad,v,n)-Jr0(rad,v)-betaL*JL(rad,v,n)-
pN(rad,v,n)*betaN*JN(rad,v,n)-pSi(rad,v,n)*betaSi*JSi(rad,v,n) pN(rad,v,n)*JN(rad,v,n)*CNtot(rad,v)
pSi(rad,v,n)*JSi(rad,v,n).*RCSi_g(rad,v)]),0);

            if JL(rad,v,n)-Jr0(rad,v)-betaL*JL(rad,v,n)-pN(rad,v,n)*betaN*JN(rad,v,n)-
pSi(rad,v,n)*betaSi*JSi(rad,v,n)<0
                limmat(rad,v,n) =0;
            end
        end
    end
    Jtot(:,:,n)                  = Jmax.*Jlieb(:,:,n)./(Jlieb(:,:,n)+Jmax); % µg C/day
    dgrowth(:,:,n)               = Jtot(:,:,n)./(Ctot);

    % Predation mortality
    mu(:,:,n)                    = m0+fG*G(n)*(V0+Vs).^(-1/4);

    % Net growth in trait space
    growth(:,:,n)                = dgrowth(:,:,n) - mu(:,:,n) - ms;
    % calculate all optimal parameters
    growth1                 = growth(:);
    [g_opt(n),I(n)]         = max(growth1);          % save optimal growth with index
    [I_r(n), I_v(n)]        = ind2sub(size(growth(:,:,1)),I(n));
    v_opt(n)                = vfrac(I_r(n),I_v(n));  % optimal vacuole fraction
    r_opt(n)                = ESR(I_r(n),I_v(n));    % optimal size (radius)
    C_opt(:,n)              = c(I_r(n),:);           % cytoplasmic carbon content for optimal vacuole size
    C_opt_p(:,n)            = c(I_r(n),I_v(n));      % optimal cytoplasmic carbon content
    Ctotopt(n)              = Ctot(I_r(n),I_v(n));   % optimal total carbon content(µg C/µm3)
    dcost_opt(n)            = dgrowth(I_r(n),I_v(n),n);  % division rate at optimal growth
    ms_opt(n)               = ms(I_r(n),I_v(n));     % sinking rate at optimal growth
    mu_opt(n)               = mu(I_r(n),I_v(n),n);   % grazing loss at optimal growth
    % upper and lower 95% of optimal parameters
    growth2(:,:,n)               = growth/max(max(growth)); % scaled optimal growth
    [I_r95, I_v95]               = find(round(growth2(:,:,n),2)==0.95);
    if isempty(I_r95) || isempty(I_v95)
        Ir_max(n)                = res;
        Ir_min(n)                = 1;
        Iv_max(n)                = res;
        Iv_min(n)                = 1;
    else
        Ir_max(n)                = max(I_r95);
        Ir_min(n)                = min(I_r95);
        Iv_max(n)                = max(I_v95);
        Iv_min(n)                = min(I_v95);
    end
    Vpercent_opt95u(n)    = vfrac(Ir_max(n),Iv_max(n));     % vacuole percent at upper 95% of optimal growth
```

```
    r_opt95u(n)          = ESR(Ir_max(n),Iv_max(n));       % size (radius) at upper 95% of optimal growth
    C_opt95u(n)          = Ctot(Ir_max(n),Iv_max(n));      % carbon content at upper 95% of optimal growth
    Vpercent_opt95l(n)   = vfrac(Ir_min(n),Iv_min(n));     % vacuole percent at lower 95% of optimal growth
    r_opt95l(n)          = ESR(Ir_min(n),Iv_min(n));       % size (radius) at lower 95% of optimal growth
    C_opt95l(n)          = Ctot(Ir_min(n),Iv_min(n));      % carbon content at lower 95% of optimal growth
end

%% plots
set(0,'DefaultFigurePosition', [100 100 900 600]);
set(0,'DefaultAxesFontSize','default')
set(0,'DefaultLineLineWidth','default')
set(0,'DefaultFigureColor','w')
monthstr = {'Light limited', 'Non limited', 'Nut/grazing limited'};
color = [0 0 255;30 144 255];

fig=figure(1);
fig.Units = 'inches';
p = [7 3 6 5];
set(fig,'pos',p)
lpos2 = 0.13;
lpos22 = 0.06;
lpos1 = 0.08;
lofset = 0.1;
name1 = {'a','b','c','d','e','f'};
for n = 1:length(L)
    h = subplot(3,3,n);
    p = get(h,'pos');
    p(3:4) = p(3:4)*1.2;
    p(2)= p(2)*0.9;
    set(h,'pos',p)
    scatter(log10(C_opt_p(n)*10^6),v_opt(n)*100,g_opt(n)*100,
'filled','d','markerfacecolor','k','markeredgecolor','k'),
    box on
    hold on
    contour(log10(c(1,:)*10^6),vfrac(:,1)*100,growth2(:,:,n),[0.99 0.99],'linecolor',[0 0
255/255],'linewidth', 1.2);
    hold on
    contour(log10(c(1,:)*10^6),vfrac(:,1)*100,growth2(:,:,n), [0.90 0.90],'LineColor',[30 144
255]./255,'linewidth', 1.2);
    hold on
    contour(log10(c(1,:)*10^6),vfrac(:,1)*100,growth2(:,:,n), [0.8 0.8],'LineColor',[130 200
255]./255,'linewidth', 1.2);
    hold on
    contour(log10(c(1,:)*10^6),vfrac(:,1)*100,growth(:,:,n),[0 0],'k','linewidth', 1.2);
    hold on
    plot(log10(c(1,:)*10^6),ones(1,res)*v_opt(n)*100,'--k')% 'color',[0.1 0.1 0,1],'linestyle',)
    hold on
    contour(log10(c(1,:)*10^6),vfrac(:,1)*100,RSiC,[0.052 0.55],'linecolor',[0 1 0],'linewidth',0.7);
    hold on
    scatter(log10(C_opt_p(n)*10^6),v_opt(n)*100,g_opt(n)*100,
'filled','d','markerfacecolor','k','markeredgecolor','k'),
    box on
    if n == 1
        ylabel('Vacuole (%)')
    end
    title(monthstr(2))
    if n==1 || n==2 || n == 3
        set(gca, 'XTickLabel', []);
    end
    if n==2 || n == 3
        set(gca, 'YTickLabel', []);
    end
    if n== 3
        lh = legend('net growth','99%','95%','90%', 'growth = 0');
        set(lh,'orientation','horizontal')
        legend boxoff
        pos = get(lh,'pos');
        pos(1) = 0.12;
        pos(2) = lpos2;
        set(lh,'pos',pos)
    end
```

```
    %     h = subplot(3,3,n+3);
    %     p = get(h,'pos');
    %     p(3:4) = p(3:4)*1;
    %     set(h,'pos',p)
    text(-0.8,93,name1(n),'fontweight','bold')
    h = subplot(3,3,n+3);
    p = get(h,'pos');
    p(3:4) = p(3:4)*1.2;
    p(2)= p(2)*0.8;
    set(h,'pos',p)
    plot(log10(c(1,:)*10^6),growth(I_r(n),:,n),'k', 'linewidth',1.5), hold on
    plot(log10(c(1,:)*10^6),-ms(I_r(n),:),'m','linewidth',1.2), hold on
    plot(log10(c(1,:)*10^6),-mu(I_r(n),:,n),'r','linewidth',1.2), hold on
    plot(log10(c(1,:)*10^6),dgrowth(I_r(n),:,n),'b','linewidth',1.2), hold on
    gg = growth(I_r(n),:,n);
    indx = find(gg>0);
    gg = gg(indx);
    tt = log10(c(1,:)*10^6);
    tt = tt(indx);
    gg0 = zeros(size(gg));
    patch([tt(:); flipud(tt(:))],[gg(:); flipud(gg0(:))],[0.8 0.8 0.8], 'EdgeColor', [0.8 0.8
0.8],'Linewidth',1.5), hold on
    axis tight
    box on
    plot(log10(c(1,:)*10^6),growth(I_r(n),:,n),'k', 'linewidth',1.5), hold on
    plot(log10(c(1,:)*10^6),-ms(I_r(n),:),'m','linewidth',1), hold on
    plot(log10(c(1,:)*10^6),-mu(I_r(n),:,n),'r','linewidth',1), hold on
    plot(log10(c(1,:)*10^6),dgrowth(I_r(n),:,n),'b','linewidth',1), hold on
    ylim([-1.5 1.5])
    if n+3==5
        xlabel('Cell carbon (pg)')
    end
    if n+3==4
        ylabel('Rate (day^{-1})')
    end
    if n==2 || n == 3
        set(gca, 'YTickLabel', []);
    end
    if n== 1
        lh = legend('Net growth','Sinking loss','Grazing loss','Division rate');
        set(lh,'orientation','horizontal')
        legend boxoff
        pos = get(lh,'pos');
        pos(2) = lpos22;
        pos(1) = 0.12;
        set(lh,'pos',pos)
    end
    text(-0.8,1.35,name1(n+3),'fontweight','bold')
end

fig=figure(2);
fig.Units = 'inches';
p = [7 3 6 5];
name = {'a','b','c'};
set(fig,'pos',p)
for n = 1:length(L)
    h = subplot(3,3,n);
    p = get(h,'pos');
    p(3:4) = p(3:4)*1.2;
    p(2)= p(2)*0.9;
    set(h,'pos',p)
    imagesc(log10(c(1,:)*10^6),vfrac(:,1)*100,limmat(:,:,n),[0 4]), hold on
    scatter(log10(C_opt_p(n)*10^6),v_opt(n)*100,'d','markerfacecolor','k','markeredgecolor','k')
    if n==1
        ylabel('Vacuole (%)')
    end
    if n==2
        xlabel('Cell carbon (pg)')
    end
    % xlim([0 100])
    % ylim([0 rmax])
```

```
    if n==2 || n== 3
        set(gca, 'YTickLabel', []);
    end
    title(monthstr(2))
    text(-0.8,5,name(n),'fontweight','bold')
end

%% Find optimal growth in trait space

% non vacuolated cell
carb    = C_opt;                    % Carbon in cytopasm (µg C)
nitro   = carb*1/RCN_g;             % Nitrogen in cytoplasm (µg C)
size    = C_opt./ccb_g;             % Volume of cytoplasm
radi    = (size./(4/3*pi)).^(1/3);  % radius of cytoplasm (µm)
memvol  = 4/3*pi*(radi+m).^3-size;  % plasmalemma volume (µm3)
totsize = size + memvol;            % Volume including plasmalemma membrane
Cmem1   = rhoCtono*memvol;          % Carbon in plasmalemma µg C
Nmem1   = 1/CNtono*Cmem1*12/14;     % Nitrogen in plasmalemma
CNtot1  = (carb+Cmem1)./(nitro + Nmem1);% total new CN ratio
rNon    = ((totsize)./(4/3*pi)).^(1/3)*10^-6;% raduis of of cytoplasm and plasmalemma in meters (m)
AL1     = cL.*(totsize).^(2/3);     % µg C/day/(W/m2) (r,v)
AN1     = 4*pi*DN*(rNon)*10^3;      % l/day
Jr01    = 0.05*(carb+Cmem1);        % respiratory cost
Jmax1   = jmax*carb;                % maximum synthesis

for v = 1:length(carb(1,:))
    JL1(:,v)    = AL1(:,v)*L(v);
    JN1(:,v)    = AN1(:,v)*N(v);
    for rad     = 1:res
        pNN(rad,v)    = max(0,min((JL1(rad,v) - Jr01(rad,v) - JL1(rad,v)*betaL)/(JN1(rad,v)*betaN +
JN1(rad,v)*CNtot1(rad,v)),1));
        Jlieb1(rad,v) = max(min([JL1(rad,v)-Jr01(rad,v)-betaL*JL1(rad,v)-pNN(rad,v)*betaN*JN1(rad,v)
pNN(rad,v)*JN1(rad,v)*CNtot1(rad,v)]),0);
    end
    mg(:,v)     = m0 + fG*G(v)*(totsize(:,v)).^(-1/4);
end

Jtot1   = Jmax1.*Jlieb1./(Jlieb1+Jmax1);
div     = Jtot1./carb;
w1      = 2*grav*(10^-6*(size./(4/3*pi)).^(1/3)).^2/(9*nu).*(rhoB-rhoW);
ms1     = w1./H*60*60*24;   % sinkin mortality (day^-1)
grow    = div-mg-ms1;

% Plot comparison between diaton and non-diatom
fig=figure(3);
fig.Units = 'inches';
p = [7 3 6 5];
set(fig,'pos',p)
lpos2 = 0.13;
lpos22 = 0.06;
lpos1 = 0.15;
lofset = 0.19;
for n=1:length(L)
    h = subplot(3,3,n);
    p = get(h,'pos');
    p(3:4) = p(3:4)*1.2;
    p(2)= p(2)*0.9;
    set(h,'pos',p)
    plot(log10(carb(:,n)*10^6),growth(I_r(n),:,n),'k','LineWidth',1), hold on
    plot(log10(carb(:,n)*10^6),grow(:,n),'--k','LineWidth',1)
    lh = legend('Net growth');
    set(lh,'orientation','horizontal')
    legend boxoff
    pos = get(lh,'pos');
    pos(1) = lpos1;
    pos(2) = lpos2;
    set(lh,'pos',pos)
    if n ~=1
        set(gca, 'YTickLabel', []);
    elseif n == 1
        ylabel('Rate (day^{-1})')
```

```
        set(gca, 'XTickLabel', []);
    end
    if n == 2 || n== 3
        set(gca, 'XTickLabel', []);
    end
    ylim([0 1])
    xlim([-1 5])
    title(monthstr(2))
    text(-0.8,0.95,name1(n),'fontweight','bold')
    h = subplot(3,3,n+3);
    p = get(h,'pos');
    p(3:4) = p(3:4)*1.2;
    p(2)= p(2)*0.8;
    set(h,'pos',p)
    plot(log10(carb(:,n)*10^6),dgrowth(I_r(n),:,n),'b','LineWidth',1), hold on
    plot(log10(carb(:,n)*10^6),-ms(I_r(n),:),'m','LineWidth',1), hold on
    plot(log10(carb(:,n)*10^6),-mu(I_r(n),:,n),'r','LineWidth',1), hold on
    plot(log10(carb(:,n)*10^6),-mg(:,n),'--r','LineWidth',1), hold on
    plot(log10(carb(:,n)*10^6),div(:,n),'--b','LineWidth',1), hold on
    plot(log10(carb(:,n)*10^6),-ms1(:,n),'--m','LineWidth',1)
    if n+3==5
        xlabel('Cell carbon (pg)')
    end
    ylim([-1.5 1.5])
    xlim([-1 5])
    if n+3 ~=4
        set(gca, 'YTickLabel', []);
    elseif n+3 == 4
        ylabel('Rate (day^{-1})')
    end
    lh = legend('Division rate', 'Sinking loss', 'Grazing loss');
    set(lh,'orientation','horizontal')
    legend boxoff
    pos = get(lh,'pos');
    pos(1) = lpos1+lofset;
    pos(2) = lpos2;
    set(lh,'pos',pos)
    text(-0.8,1.35,name1(n+3),'fontweight','bold')
end
```

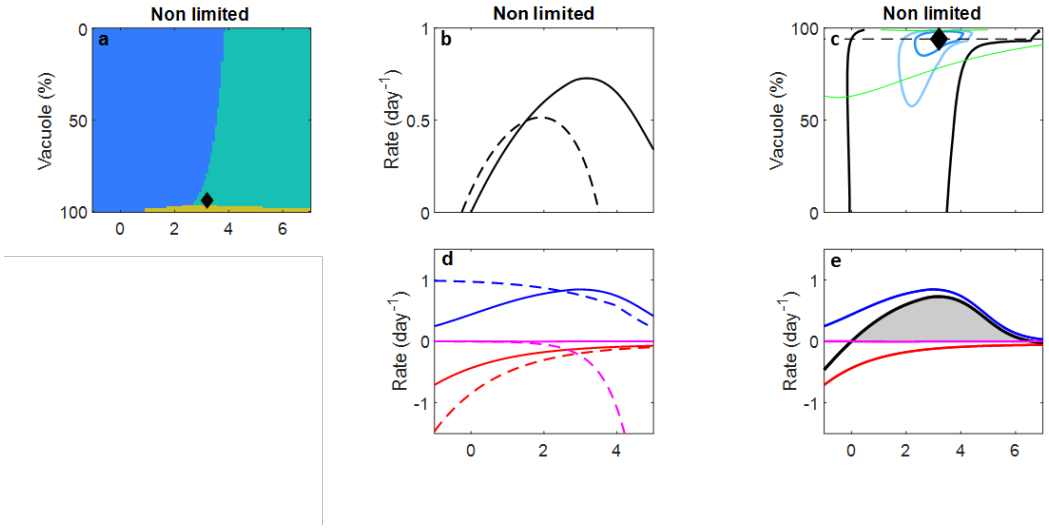


Figure 18: typical output from Diatom model. Specific parameter setting are: L = 70 Wm-2 N = 40 mg N m^{-3}, Si = 30 mg $^{-3}$, Z = 15 mg C m^{-3}.

9.3 Transport Matrix

The following code will reproduce Figure 14. Run `inputs_for_box` first (see below).

```
% % ComputeSquestration.m
% % Code developed by Jerome Pinti
% % For questions and remarks, please email pintijerome@gmail.com
% % This code is based on the transport matrix method developed by Tim
% % DeVries from UCSB. The transport matrix can be downloaded here:
        https://tdevries.eri.ucsb.edu/models-and-data-products/
% % References to the transport matrix are DeVries and Primeau 2011 and DeVries 2014.
%% Load the transport matrix
% load('91x180x48_omega3_Kvvariable_geoLuc_AH2e5.mat') %high resolution - ~10 minutes to inverse

load('CTL.mat') % lower resolution - less than a minute

%% Preparation of the transport matrix
grid = output.grid;
TR = output.TR; % yr^-1
msk = output.msk;
M3d = output.M3d; % land = 0. ocean = 1


VOL = grid.DXT3d.*grid.DYT3d.*grid.DZT3d;
V = VOL(msk.pkeep);

m = size(TR,1);
sink = zeros(m,1);
sink(1:length(msk.hkeep)) = 1e10; % instantaneous Surface SINK
SSINK = spdiags(sink,0,m,m);
A = TR-SSINK; % transport + sink in surface

% for load run-compiled-236
[lonq,latq,zq] = meshgrid(x,y,z);
Q = permute(Q, [2 1 3]);
q_ocim = interp3(lonq,latq,zq,Q,grid.XT3d,grid.YT3d,grid.ZT3d);
q_ocim = q_ocim(msk.pkeep);
q_ocim(isnan(q_ocim)) = 0;
export = V'*q_ocim / 1e15; % [PgC / yr]
tic
cseq = -A \q_ocim;
toc
TotCseq = V'*cseq / 1e15
seqime = TotCseq / export



% Re interpolate to Q
qocim = 0*M3d+NaN; % make a 3-d array of NaN
qocim(msk.pkeep) = q_ocim;
qocim(isnan(qocim)) = 0;

C_eq = 0*M3d+NaN;
C_eq(msk.pkeep) = cseq;
C_eq(isnan(C_eq)) = 0;

%% A simple plot
LAT = repmat(y,nx+1,1);
LON = repmat([x,360]',1,ny);
cc = sum(C_eq.*grid.DZT3d,3); cc = [cc, cc(:,end)]; % gC/m2/day
%North Pacific
xred = [140 180 180 140 140];
yred = [37 37 50 50 37];

% %North Atlantic
% xred = [320 350 350 320 320];
% yred = [53 53 66 66 53];
```

```
figure
cc(cc<0) = 0;
ax = axesm('mollweid','Frame','on','MapLatLimit',[-90 90],'Origin', [0 -160 0],'FLineWidth',0.5);
ax.XTick = [-120 -60 0 60 120 180];
ax.YTick = [-40 -20 0 20 40];
% objects = [handlem('grid'); handlem('mlabel'); handlem('plabel')];
% set(objects,'Clipping','on');
box off
axis off
load coast
geoshow(lat, long,'Color','k')

surfm(LAT', LON', cc,'AlphaData',~isnan(cc));%,'EdgeColor','none')
geoshow('landareas.shp', 'FaceColor', [0.5 0.5 0.5]);
plotm(yred,xred,'r')
colorbar

%% Figure from the poles
figure
subplot(121)
ax = axesm('eqdazim','Frame','on','MapLatLimit',[10 90],'Origin', [90 0 0],'FLineWidth',0.5);
% eqdazim
box off
axis off
load coast
% geoshow(lat, long,'Color','k')
surfm(LAT', LON', cc,'AlphaData',~isnan(cc));
geoshow('landareas.shp', 'FaceColor', [0.5 0.5 0.5]);
colorbar
plotm(yred,xred,'r')


subplot(122)
ax = axesm('eqdazim','Frame','on','MapLatLimit',[-90 -10],'Origin', [-90 -180 0],'FLineWidth',0.5);
% eqdazim
box off
axis off
load coast
% geoshow(lat, long,'Color','k')
surfm(LAT', LON', cc,'AlphaData',~isnan(cc));
geoshow('landareas.shp', 'FaceColor', [0.5 0.5 0.5]);
colorbar



% % inputs_for_box.m
% % Code developed by Jerome Pinti
% % For questions and remarks, please email pintijerome@gmail.com


%% Make base grid
x = 0:1:359; nx = size(x,2); % [degrees] longitude
y = -90:1:90; ny = size(y,2); % [degrees] latitude
z = 0:100:8000; nz = size(z,2); % [m] depth
dz = 100; % [m] vertical resolution


%% Make input for rough finmarchicus calculation ATLANTIC
a = 2.3; % [gC / m2 / day] Export

Q = zeros(nx,ny,nz);
s = gaussmf(z,[50 1000]); %first is sd and then mean
s = s/sum(s); %integral of the gaussian is 1
s = a * s / dz; % [gC / m3 / yr] Respiration rate at each depth
s = reshape(s,[1 1 size(s,2)]);

longres = (x>(360-40)) & (x<(360-10));
latres = (y>53) & (y<66);
Q(longres,latres,:) = repmat(s, sum(longres), sum(latres), 1); % [gC / m3/ yr]
```

```
%% Make input for rough Neocalanus calculation PACIFIC
a = 4.3; % [gC / m2 / day] Export

Q = zeros(nx,ny,nz);
s = gaussmf(z,[50 1000]); %first is sd and then mean
s = s/sum(s); %integral of the gaussian is 1
s = a * s / dz; % [gC / m3 / yr] Respiration rate at each depth
s = reshape(s,[1 1 size(s,2)]);

longres = (x>(140)) & (x<(180));
latres = (y>37) & (y<50);
Q(longres,latres,:) = repmat(s, sum(longres), sum(latres), 1); % [gC / m3/ yr]
```

This code produces Figure 14