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

Mechanistic model of zooplankton

Work package: **WP3**

Deliverable No. (D3.3) – Title**: Mechanistic model of zooplankton**

Lead Beneficiary (WP3): DTU Aqua

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

Editors (Name, Institution): Brian MacKenzie (DTU Aqua)

Contributors (Name, Institution):

Submission Date: 30/05/2022

Table of Contents

1 Executive Summary

This deliverable supplies a demonstration of the metazoan plankton component (essentially copepods) of a generalized trait-based plankton model (NUM 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, and how it is integrated with the unicellular component of the NUM model. The deliverable provides a "cook book" of how a user can access the model code and integrate it into a Matlab "front end" for their own use and experimentation. The deliverable also provides a set of examples that users can follow. All code is freely available.

2 Deliverable description from DoA

Code and documentation of trait-based model of the zooplankton (copepods) component of the marine ecosystem. Model will be coupled to unicellular model (D3.1) and to the transport matrix model (D1.2) to provide an estimate of functional diversity and key ecosystem metrics (export flux, trophic transfer efficiency).

This deliverable is a demonstration. The first part of this report is a description of the model methodology. Within this there is documentation for the rationale of the modelling approach, the model design building from energy budgets of individual organisms through to size structured populations of metazoans, and their interaction with their physical, biotic and biogeochemical environment. The second section documents how users can download and run their own model simulations. All the code is freely available on Github. The bulk of the code is written and compiled in Fortran, and for convenience (not just the readers' but also the developers), we have written some simple front end code in Matlab through which appropriate parameters can be passed, the various simulations can be run, and the results analysed and plotted. There is a similar front end written in R but is not expanded on here. We have developed 3 model systems of increasing complexity; (1) the very simplest a chemostat setting, (2) a water column setting driven by a climatological seasonal cycle (*e.g.* light, temperature, mixing) where users can choose any location and (3) a global setting over an annual cycle. The physical environment is supplied by a transport matrix – essentially the offline circulation results from a Geophysical Fluid Dynamic (GFD) ocean model. The NUM modelling framework presented here is relatively complete, but there is considerable scope for refinements.

3 Contribution to ECOTIP tasks

This deliverable is a follow up from deliverables D3.1 and D3.2 and describes the multicellular components of the NUM marine ecosystem model. In this context multicellular components refer to metazoan zooplankton typified by copepods. 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 to 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 interactions) when the full model is completed (MS18). The ultimate ambition of this task is to include a trait-based approach into biogeochemical models as a component to Earth System Models.

Table 1: Contribution of D3.3 (Mechanistic model of zooplankton) to addressing the knowledge gaps identified by ECOTIP.

Knowledge gaps to be investigated in ECOTIP	Contribution by deliverable D3.3
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	Assesses the coverage of coupled unicellular and metazoan plankton traits.
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	Is an integral part of the development of an integrated ecosystem model providing an essential link between primary production and

from an early ice melt. However, we do not know how that affects trophic linkages and food webs.	higher trophic levels (fish & fisheries). Provides a framework 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	

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

4 General Modelling Framework

The general modelling framework we pursue in ECOTIP is the NUM model. This is a generalized mechanistic and trait-based model of planktonic marine ecosystems that is founded, as much as possible on first principles – the fundamental axioms, assumptions and laws from which natural systems arise. This is what we mean by mechanistic. The focus on individual organisms follows naturally from this proposition as this is where the fundamental organizing principle of life – evolution by natural selection – applies. This view is very much facilitated by a trait-based approach which places the emphasis on what organisms do rather than the taxonomic group they belong to. Specifically, we characterize organisms by a handful of key functional traits (inherent characteristics) that govern how an organism performs (grows, survives, and reproduces) in a given ecological setting (light nutrients, prey, predators, competitors). The key insight here is that traits are invariably constrained by trade-offs. That is, no matter what trait values an organism has, it will never be able to always achieve superior performance in all places and at all times. Darwin's daemon cannot exist. First principles provide much of the mechanistic reasoning from which trade-offs can be formulated. This approach provides a modelling framework that is predictive and global is scope.

We term our approach the "NUM" modelling framework. This refers to Nutrient – Unicellular – Multicellular and describes one of the major trait axes of living organisms. The nutrient and unicellular components of this modelling framework were described in D3.1 and critically reviewed in D3.2. The unicellular component deals with phytoplankton, microzooplankton along an auto- mixo- heterotrophic axis. The topic of this deliverable is the multicellular components of the planktonic marine ecosystem. While organisms such as euphasids and chaetognaths certainly fall within this category, in this work we focus primarily on copepods as these are by far and away the major taxa occupying this role in most marine ecosystems, particulary in Arctic and subarctic reaches of the North Atlantic.

In terms of our modelling approach, the key difference between unicellular and multicellular organisms is that the former reproduce by cell division, while the latter grow often times many order of magnitude in body mass from a juvenile to a reproductive adult. It is this feature – the ontogenic progression of multicellular organisms – that the NUM modelling framework is able to resolve.

4.1 Background to NUM

Planktonic food webs are an intricate weave of trophic pathways channelling energy and matter between a wide variety of producers and consumers. Yet, the complexity of these food webs is often simulated as a simple NPZ model: a compartmentalized tropic chain linking the interactions between nutrients, phytoplankton and zooplankton (Evans & Parslow 1985, Franks 2002, Gentleman 2002). The obvious simplicity of this construction has led to a multitude of modifications too numerous to list here, each introducing an additional set of compartments to inject more realism into the simulations. Among the most successful of these have been the functional group models that recognize various nutrients and their pathways at the base of the pelagic food web (Follows & Dutkiewicz 2011), and have given rise to considerable advances in the understanding of the biogeochemistry of the oceans (Weitz et al. 2015). However, extending this approach to higher trophic levels, particularly towards fish and fisheries, remains elusive (Fulton 2010), in part because of an incomplete representation of zooplankton (Mitra et al. 2014). Throughout this development process, the basic NPZ model has remained remarkably resilient in the face of critique (Franks 2002), and in some sense has served as a foundation against which the effect of certain features of planktonic food webs can be quantified. Some recent examples have highlighted, for instance the importance of trophic interactions mediated by size structure (Baird & Suthers 2007, Fuchs & Franks 2010, Banas 2011, Ward et al. 2012), feeding behaviour (Castellani et al. 2013) and mixotrophy (Ward & Follows 2016, Chakraborty et al. 2017) in determining the structure and function of planktonic communities. It is with this approach in mind that we formulated the NUM modelling framework, specifically to explore how ontogeny – the growth in size of individuals from juveniles to reproductive adults – and associated life history affects descriptions of marine ecosystem dynamics.

The zooplankton compartment of NPZ models generally represents the biomass of two types of organism; protist (flagellates and ciliates) and meso-zooplankton (copepods). Apart from overall size, an important distinction between these is that protist undergo reproduction via simple cell division, whereas meso-zooplankton grow as individuals from an egg to an adult often spanning several orders of magnitude in mass throughout their lifetime. Most importantly, this change in size will alter not only the rate of physiological processes, but also the trophic position of the individual within the community – both in what it eats and what seeks to eat it. Precisely how this shifting diet influences the outcome of community dynamics remains unclear. It could be argued that such a population of ontogenic consumers could act as strong vertical integrators in the overall trophic transfer within a system, promoting stability of the system to sporadic perturbations (De Roos et al. 2008). Conversely, given that growth requires time, ontogeny can expose populations to trophic bottlenecks (Werner & Gilliam 1984), the emergence of cyclic cohort dynamics (McCauley & Murdoch 1987) or time delays (May 1973) for instance between the emergence of juveniles and peak consumption by adults. Indeed the latter is often cited as a key factor leading to the spring bloom in seasonal environments (Longhurst 1995, Behrenfeld & Boss 2014).

Copepods are the most widespread and abundant meso-zooplankton in the oceans, and are arguably the most abundant metazoans on earth (Turner 2004). These compose an important trophic link between primary producers (and their protist grazers) to higher trophic levels such as fish, seabirds and marine mammals. In this, they serve precisely the function that the Z compartment of NPZ models are designed to capture. Pelagic adult copepods range in mass from < 1 µgC (*e.g. Oithona similis, Microsetella norvegica*) to > 700 µgC (e.g. *Calanus hyperboreus, Eucheata norvegica*). In general they spawn eggs that are typically 0.01 to 0.001 of the mass of the adults (Kiørboe & Sabatini 1995, Andersen et al. 2016). From hatching, an individual moults successively into 5-6 naupli stages, which are followed by 5-6 copepodite stages before the individual finally moults into an adult. Depending on temperature and food conditions, this transition from egg to adult typically takes 20 to 40 days (Kiørboe & Sabatini 1995), although many species can arrest development at certain stages, producing resting eggs or entering diapause to survive adverse conditions.

Throughout this progression, individuals often change their trophic guild and feeding mode. Nauplii in particular tend to have quite distinct trophic arrangements that their older stages. Nauplii of several species for instance have been show to feed on bacterioplankton (Roff et al. 1995) connecting directly into the microbial loop, while others feed on a variety of autotrophic, mixotrophic and heterotrophic protist prey (Henriksen et al. 2007). The fraction of potential food sources available to nauplii and thus “selected for” depends on their feeding mode expressed in a variety of motility modes, e.g. ambush with occasional repositioning jumps, hop-sink, or continuous curvilinear swimming (Titelman & Kiørboe 2003). These feeding modes differ not only between species, but change over the ontogenic development of an individual (Titelman & Kiørboe 2003). Early naupliar stages often don’t feed at all, relying on reserves inherited from their mother. Different feeding modes translate not just into different diets but also different predation risk (Titelman 2001), not only in absolute terms, but also in the type of predator (e.g. ambush, cruise, visual, rheotactic) and hence their relative abundance (Visser & Fiksen 2013). What emerges is a trophic landscape dominated not so much by aggregated zooplankton species, but by the shifting demographics within the zooplankton community.

4.2 Unicellular component; phytoplankton and microzooplankton.

The unicellular component of the NUM model was described in detail in D3.1. Several of its components have appeared in the literature (Chakraborty et al. 2017, Hansen & Visser 2019, Serra-Pompei et al. 2020, Cadier et al. 2020). This model considers most of the major classes of unicellular organisms in the marine environment - from bacteria to microzooplankton. It uses size as a master trait. Trophic function is plastic and is determined by the cell’s investments in light harvesting, nutrient uptake(both mineral and dissolved organic), and phagotrophy. It inherently allows for mixed trophic strategies, *i.e.* mixotrophy to emerge as an optimal strategy. The organisms that occupy this trait axis we term generalists spanning from pure autotrophs to pure heterotrophs with mixotrophs inbetween. These include a broad class unicellular organisms that include representations corresponding to cyanobacteria (e.g. *Prochlorococcus*), algal cells (*Chlorarachniophyte* etc.), phytoflagellates, dinoflagellates, and ciliates.

Because of their importance in temperate and high latitude oceans, the model also incudes a trait axis representing diatoms – where the main trait axes are cell size and vacuole volume both of which determine their mass of the silicate shell together with investment in light harvesting. Finally, there is a representation of osmotrophic bacteria feeding on dissolved organic matter. The microbial loop has a relatively simple representation with a constant turn over time, and detrital material, both dissolved and particulate serving as a nutrition source for heterotrophic microorganisms.

For more information on this component of the NUM modelling framework, the reader is directed to two previous deliverables: D3.1 Mechanistic, trait-based model for unicellular plankton, and D3.2 Microbial traits.

4.3 Multicellular component

The model extends the unicellular component of the NUM framework by specifically including life history representations of a community of consumers that feed on unicellular organisms as well as each other, and grow from a juvenile to a reproducing adult. The pre-juvenile life stages (eggs and early naupliar stages) are not specifically considered as they are non-feeding and are fuelled by maternal investment of energy reserves. There is a mortality effect on these non-feeding stages that we set to a constant for simplicity. As with the unicellular component, size is considered a master trait, although in this context it is the asymptotic size (specifically the mass of reproductive adults m_a) with the size of a juvenile (mass m_0) taken to be a constant ratio with the adult size. There are other traits axes in the multicellular components (*e.g.* motility, reproductive strategy, vertical migration). In this work we focus on that which we posit as having has the most immediate effect on plankton community structure – namely feeding mode. Specifically zooplankton can adopt either a passive (*i.e.* ambush) or active (*i.e.* cruise of feeding current) feeding mode. These carry different cost-benefits in terms of maximum ingestion rate, energetic costs and predation risk.

The following subsections provide a more detailed description of this model component.

4.3.1 Individual level

The fundamental unit of consideration for modelling is that of an individual multicellular metazoan of mass *m*. The energy budget for such an individual (Figure 1) partitions income from food ingestion into metabolic costs, growth and excretion. These are primarily function of the instantaneous mass of the individual (Brown et al. 2004, Kiørboe & Hirst 2014, Andersen et al. 2016). Specifically, if *E*(*m*) [µgC L^{-1}] is the food available to the individual, then we define a feeding level as

$$f(m)=\frac{vm^{q}E(m)}{vm^{q}E(m)+hm^{n}} \qquad (1)$$

where vm^q [L day^{-1} µgC] is the clearance rate and hm^n [day^{-1}] is the maximum ingestion rate. The coefficients *q* and *n* represent appropriate allometric scaling. The feeding level *f*(*m*) thus lies between 0 and 1 and actual ingestion rate can be conveniently expressed as a fraction of the maximum: $hm^n\,f(m)$. Of this, a certain fraction ε is assimulated and the remaining fraction $(1-\varepsilon)$ is excreted as fecal pellets. The individual also pays a metabolic cost κm^p, so that the net potential growth rate is

$$\xi(m)=\varepsilon\cdot h\cdot m^{n}\cdot f(m)-\kappa\cdot m^{p} \qquad (2)$$

We can define a critical feeding level $f_c(m)=\kappa\cdot m^{p-n}/(\varepsilon\ h)$ below which metabolic costs exceed income; growth is no longer possible, and the individual starves and can be represented as a loss rate.

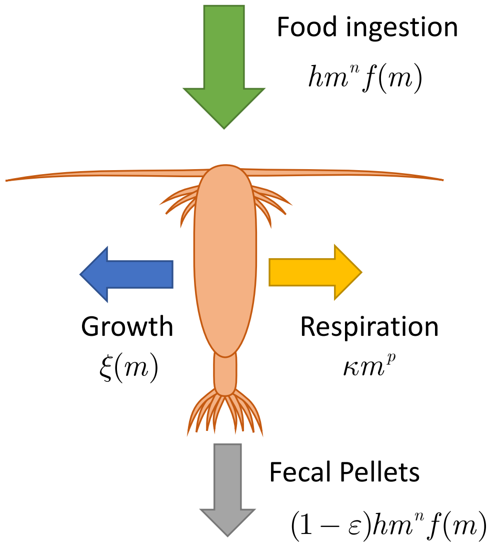


Figure 1: Energy budget for a multicellular organism. Ingested energy is partitioned between respiration, growth (somatic growth for individuals $m < m_a$, and reproduction for $m \geq m_a$) and fecal pellet production.

Parametrizing maximum ingestion rates hm^n [day^{-1}] follow from observations (Kiørboe & Hirst 2014) and provide an estimate for the coefficients n = -1/4 and h = 0.4 µgC$^{1/4}$ day^{-1}. The mass power coefficient n = -1/4 is consistent with the required general balance between ingestion and metabolic rates. While assimilation efficiencies vary widely, we assume a value of 2/3 which falls within observed ranges.

When individuals are smaller than their asymptotic size ($m < m_a$), then they use net excess income for somatic growth. Once an individual reaches its adult size, this income is used for reproduction, producing juveniles of mass m_0. For all copepods, we set the ratio of m_a:m_0 as 100. Specifically we can define the birth rate of juveniles as $b = \sigma \max(0,\xi(m))$ where σ is the reproduction efficiency. The reproduction efficiency takes into account the ratio of males to females and the survival of eggs until hatching. Following (Kiørboe & Sabatini 1995), the survival of eggs up to hatching is about 0.5. This together with an assumed 1:1 sex ratio gives an overall reproductive efficiency of 0.25.

All rates vary with regards temperature. This is implemented as a Q_{10}, very much as is done with the unicellular component. For the multicellular organisms we impose $Q_{10} = 2$ on maximum ingestion rates, respiration and mortality rates (Serra-Pompei et al. 2019).

Potential food sources for multicellular organisms include (1) other multicellular organisms, (2) unicellular organisms and (3) fecal pellets. For any organism of mass m_i, available food is dependent on the range and shape of its feeding preference $\phi(m_i, m_j)$ where m_j represents that mass of a prey item. Following the same form as for unicellular organisms, this can be specifically written as

$$\phi(m_i, m_j) = \exp\left[-\frac{\left(\ln(\beta \cdot m_i / m_j)\right)^2}{2\varpi} \right] \tag{3}$$

where β is the preferred predator : prey mass ratio, and ϖ is the width of the preference kernel. The available food for this organism $E(m)$, is thus the sum of all the prey items it encounters modulated by its preference kernel.

4.3.2 Fecal pellet flux

The fraction of ingested carbon that is not as simulated or respired, *i.e.* $(1-\varepsilon)\cdot h \cdot m^n \cdot f(m)$, is assumed to form fecal pellets. These fecal pellets constitute a potentially significant source of the export production form the surface ocean. The flux itself is governed by both the mass of carbon involved, and how fast it sinks. For any particulate matter, sinking speed is set by particle size and particle excess density relative to water density. In the first instance, the size of a fecal pellet is related to the size of the individual that produced it. A useful measure from (Mauchline 1998) is $V = 3.5 \times 10^4\ [m\ (\mu gC)]^{0.938}\ \mu m^3$ (i.e. very nearly a linear function of mass). The sinking velocity is very much determined by the density of the fecal pellet which in turn depends on diet – a diet high in diatom content will produce dense fast sinking pellets through the ballasting effect of diatom frustules. For the moment we assume that fecal pellets sinking speed follows

$$w_s = 10^{-1.214}\ [V\ (\mu m^3)]^{0.513}\ \text{m day}^{-1}$$

(Small et al. 1979).

4.3.3 Population size structure, community and dynamics

While individual organisms grow from mass m_0 to m_a, the numerical representation requires a discretization into a convenient number of intervals. Choosing S intervals and a logarithmic spacing, then the mass ratio between successive bins is

$$\delta = \frac{m_{i+1}}{m_i} = \left(\frac{m_a}{m_0} \right)^{1/S} \tag{4}$$

The centre of the mass intervals are given by the vector **m**, where $m_i = m_0\ \delta_i$ ($i \in [0,S]$) while the edges of the intervals are at $[m_{i-}, m_{i+}] = m_0\ [\delta^{i-1/2}, \delta^{i+1/2}]$ ($i \in [0,S]$). At any given time, the population size structure is given by the vector **n** where the elements n_i indicate the number density of individuals in the mass range m_0 [$\delta^{i\,-1/2}$, $\delta^{i\,+1/2}$]. We can also define the vector **c** as the biomass of the population within the same intervals. These are related by **c** = **n** ○ **m** where ○ represents that piecewise matrix multiplication. This discretization of space introduces a few refinements. For example, the preference kernel becomes a matrix where

$$\mathbf{\Phi}=\Phi_{ij}=\int_{j-1/2}^{j+1/2}\int_{i-1/2}^{i+1/2}\exp\left[-\frac{\left((k-l)^2\ln(\beta\cdot\delta)\right)^2}{2\varpi}\right]dk\;dl \tag{5}$$

For any copepod defined by asymptotic mass m_a, the population size structure **c** = [c_1, c_2, …. c_S] is governed by the dynamics

$$\begin{aligned}\frac{dc_1}{dt}&=\sigma\max(0,\xi_S)\;c_S+\xi_1c_1-\gamma_1c_1-\mu_1c_1\\ \frac{dc_i}{dt}&=\gamma_{i-1}c_{i-1}+\xi_ic_i-\gamma_ic_i-\mu_ic_i\qquad\text{for}\quad 2\le i<S\\ \frac{dc_s}{dt}&=\gamma_{S-1}c_{S-1}+\left[\min(0,\xi_S)+\mu_S\right]c_S\end{aligned} \tag{6}$$

Note that ξ_i can be positive (indicating growth) or negative (indicating losses). The parameters γ [day^{-1}] and μ [day^{-1}] represent the development rate and mortality rate respectively.

The development rate γ can be derived from equilibrium conditions (De Roos et al. 2008) as:

$$\gamma=\frac{\xi}{1-\delta} \tag{7}$$

Importantly, development rates are directly linked to food availability and ingestion through ξ (Equation 2), and indirectly to temperature through temperature defence of metabolic rates and maximum clearance rates.

The metazoan community is composed of a number of different "species" each defined by a maximum (adult) body mass; each governed by dynamics described in Eq (6). Figure 2 illustrates an idealized number size spectrum for the plankton community predicted by a NUM model simulation. In this case there is a unicellular generalist component, and 4 copepod "species".

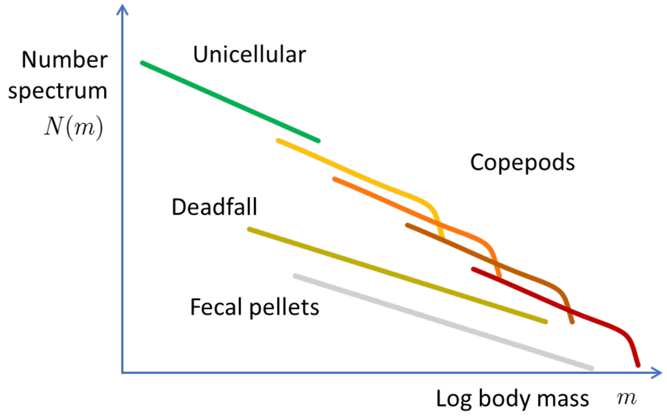


Figure 2: Conceptual community number spectrum emerging from a NUM model simulation.

4.3.4 Sheldon Spectrum

In this figure we introduce the community number spectrum. This together with the biomass spectrum useful concept in characterizing the plankton community. Firstly, we can choose $m_{\min}$ and $m_{\max}$ as the minimum and maximum body mass spanning the full range of unicellular and multicellular organisms expected in a simulation. For instance, $m_{\min}$ would be the minimum size for a generalist, and $m_{\max}$ would be the adult mass of the largest copepod. The interval $[m_{\min}, m_{\max}]$ can be divided into *Imax* intervals, each with representative mass m_i and lower and upper bounds $[m_i^-, m_i^+]$ to Specifically, we can define a community number spectrum N_i and the sum of all the individuals found in the interval m_i^+ to m_i^-. From this, the structure of the plankton community can be represented by the biomasses in the size groups B_i which is simply $B_i = m_i N_i$. This representation has the disadvantage that the level of the biomasses depend on the size-range of each group: broader (fewer) size-groups leads to higher average biomass level and *vice versa*. To avoid this dependency size distributions are often shown as "normalized size spectra" (Sprules & Barth 2016) by divided the biomass with the size range of the group: $b(m_i) = B_i / (m_i^+ - m_i^-)$. If we assume a scaling biomass spectrum, $b(m) = k\, m^{\lambda-1}$ then the relation between the normalized biomass spectrum and the binned size groups is:

$$B_i = \int_{m_i^-}^{m_i^+} b(m)\ dm = k\ \ln\left(\frac{m_i^+}{m_i^-}\right)$$

if $\lambda=0$. If size groups are evenly distributed on a log scale then the ratio (m_i^+/m_i^-) is constant (independent of mass) and the biomasses in each groups are roughly the same. To avoid that results depends on the binning of the size groups we here define the ``Sheldon" spectrum as:

$\mathcal{B}_i = B_i / \ln(m_i^+/m_i^-)$

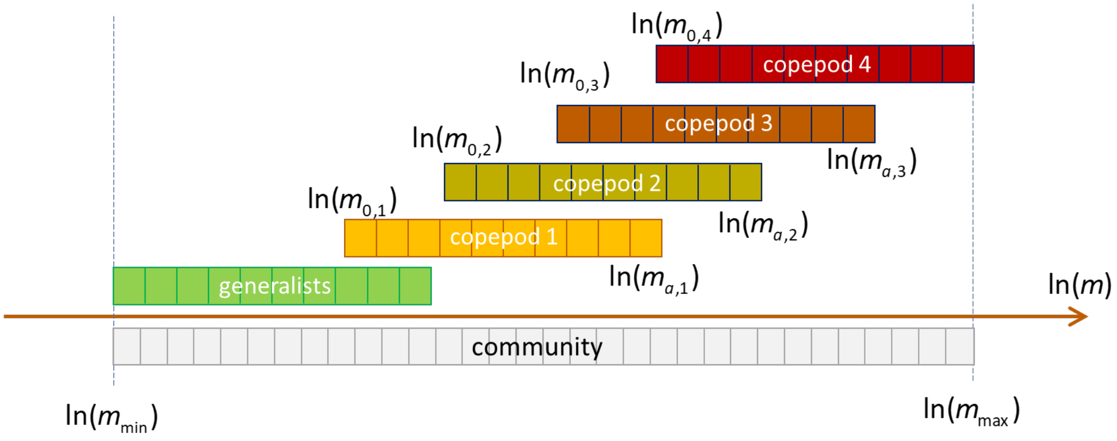


Figure 3: Relationship between the various logarithmically spaced mass bins.

In most cases, bins are equally spaced. For instance for the full range $[m_{\min}, m_{\max}]$ divided into Imax logarithmic equally spaced bins,

$$\delta' = \left(\frac{m_{\max}}{m_{\min}}\right)^{1/I_{\max}}$$

so that the boundaries of the i^{th} bin are $m_{\min}\,\delta'^{(i-1)}\,[1, \delta']$. That is, $\ln(m_i^+/m_i^-) = \ln(\delta')$ for all bins.

4.3.5 Mortality

Mortality μ can be subdivided into different contributions. That associate with starvation in already represented in the term ξ (when it becomes negative). In addition, mortality comes about from predation, both that from organisms inherently included in the simulation (other multicellular organisms), as well as other organisms (e.g. fish) that are not specifically represented. Mortality associated with predation by other simulated taxa can be seen as the flip side of feeding. Indeed, the same preference function $\phi(m_i,m_j)$ also controls mortality where m_j now represents that mass of the focal organism while m_j is the mass of its predator. Mortality rate associated with this predation is then a function of the preference kernel, the maximum ingestion rate of the predator as well as its feeding level.

Predation by unresolved higher trophic levels (e.g. fish) serve as a closure for the model. There are several options that the modelling framework leaves open for the moment. The simplest is that a selected range of the largest copepod size classes are subject to a constant mortality rate. More complex formulations take the general assumption that is prey are plentiful, so too are their predators. Following (Record et al. 2013), mortality due to higher trophic levels can be written as:

$$\mu_{HTL}(m) \approx \mu_{HTL,0}\, p_{HTL}(m)\; m^{-1/4}\; b(c(m)) \qquad (8)$$

where $\mu_{HTL,0}$ is a constant, $p_{HTL}(m)$ is a sigmoidal function that represents the transitions from internal to external predation, $m^{-1/4}$ reflects the general reduction of predation on larger organisms and $b(c(m))$ is an estimate of the higher trophic level predator biomass based on the abundance of their prey.

4.3.6 Export losses

The sinking of detrital material and associated mineral nutrients represents a small (typically 10 to 20%) but persistent loss term that limits production and trophic transfer in many parts of the world's oceans. This loss form the surface is also an important contribution to the biological pump, and the ocean's capacity to sequester atmospheric carbon. In this, detrital material comes in 3 basic forms;

- marine snow – conglomerates of dead unicellular organism and their exudates often ballasted with biogenic or atmospherically deposited minerals,
- fecal pellets – undigested prey components, often packaged into compact dense particles,
- dead-falls – the rear but rapidly sinking carcasses of multicellular organisms.

4.4 Transport Matrix

The Transport Matrix Method (TMM) (Primeau 2005, Khatiwala 2007) is a method that helps reducing computational time when running various types of global oceanic simulations. The method manages to keep the effects of advanced physical parameterizations while reducing the computational power used.
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

$$\frac{\partial \mathbf{c}}{\partial t} = \mathbf{A}(t)\ \mathbf{c} + \mathbf{q}(\mathbf{c}, t) \qquad (10)$$

Where **c** is the concentration of a particular tracer, **A**(*t*) is the transport matrix, and **q**(**c**,*t*) 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 (e.g. MIT global circulation model http://mitgcm.org/ or the University of Victoria Earth System Model http://terra.seos.uvic.ca/model/). Some are now being provided for climate change scenarios and will be made available to ECOTIP from our H2020 project cluster partners. For the purposes of this demonstration, we use a medium resolution product MITgcm_ECCO (http://kelvin.earth.ox.ac.uk/spk/Research/TMM/TransportMatrixConfigs).

MITgcm_ECCO provides a 360 (east-west) x 160 (north – south) horizontal grid and up to 23 vertical levels (water column depth permitting). The horizontal grid is 1^o x 1^o. Note that the north-south axis only spans from 80^oN to 80^oS – there is little reason for higher resolution at the North Pole while the South Pole is on dry land.

4.4.1 Transport Matrix Construction

A transport matrix (TM) essentially contains the tracer distribution after one time step from a GCM. The TM has size *N* x *N* , *N* being the total number of grid cells representing the simulation region (both horizontal and vertical), and each column of the TM describes the tracer distribution originating from a single cell such that column *j* in **A** describes the distribution of row *j* in the vector **c**. This means that it takes only one operation to simulate one time step in a global ocean model, as all small features and parametrizations of the GCM are in are included in the transport matrix.

The TM is of course very large, and demands considerable storage. MATLAB is designed specifically for efficient matrix storage and operations. As the TM only represents a single time step the transport is restricted to the neighbouring cells, which means that **A** will indeed be very sparse. In MITgcm_ECCO used here, **A** contains in excess of 68,000 x 68,000 grid cells of which only about 0.2% are nonzero.

The basic recipe for building a TM boils down to:

- Insert tracer of value 1 in a single grid cell of the GCM (set **c** to be a unit vector)
- Compute one time step
- Store the resulting tracer field, reinitialize the tracer field and compute another time step
- Repeat until the GCM is run for a specified period of time
- Insert the average of the resulting tracer fields as a column in the TM
- Reset the tracer field
- Repeat procedure for the next grid cell

In the current implementation, this routine is modified slightly to avoid numerical diffusion.

If the transport is time-independent, it is sufficient to only compute a single time step and store the distribution as a column in the TM. For the use of ocean circulation models this is not the case, as they tend to be time dependent and have seasonal changes in the transport. Therefore it is necessary to do the time averaging of the transport in order to get the best possible linear representation of the GCM transport.

As a practical matter and to speed up the TM construction process, the domain is divided into non-overlapping tiles, which consists of a grid cell containing a tracer in the center, and its surrounding neighbouring cells. The halo of dispersion is not always limited to the nearest neighbours, as transport beyond these is possible, especially in regions of deep convection. Therefore the halo typically contains more cells in the vertical. The procedure is the repeated for all cells in each tile. This approach gives a huge improvement in terms of computational time, as all tiles can be computed in parallel, as they do not overlap.

4.4.2 Explicit and Implicit Components

The final issue to note is that vertical and horizontal aspects of transport operate at quite different spatial and temporal scales. To accommodate this, the transport matrix **A** is divided into two: an explicit component $\mathbf{A}_E$ simulating horizontal advection-diffusion operating on a long time scale, and an implicit component $\mathbf{A}_I$ simulating vertical advection-diffusion operating on a short time scale. The mathematical formalism for this follow from the two ways Eq(10) can be solved numerically in time. The explicit scheme follows from the discretized from

$$\frac{\mathbf{c}^{n+1} - \mathbf{c}^{n}}{\Delta t} = \mathbf{A}\ \mathbf{c}^{n} + \mathbf{q}$$

where the superscript *n* indicates the time step. Evaluation of the righthand side is at time step *n*. On rearranging this leads to the following expression for $\mathbf{c}^{n+1}$

$$\mathbf{c}^{n+1} = \Delta t\ (\mathbf{A}\ \mathbf{c}^{n} + \mathbf{q}) + \mathbf{c}^{n} = (\Delta t\ \mathbf{A} + \mathbf{I}\ )\ \mathbf{c}^{n} + \Delta t\ \mathbf{q} = \mathbf{A}_E \mathbf{c}^{n} + \Delta t\ \mathbf{q}$$

Alternatively, the implicit scheme assumes the righthand side is evaluated at time step step *n*+1. That is

$$\frac{\mathbf{c}^{n+1} - \mathbf{c}^{n}}{\Delta t} = \mathbf{A}\ \mathbf{c}^{n+1} + \mathbf{q}$$

There is nothing in the mathematical formalism that can distinguish these expressions, both are equally true. Expanding this leads to a different expression for $\mathbf{c}^{n+1}$

$$\left[\frac{\mathbf{I}}{\Delta t} - \mathbf{A}\right]\mathbf{c}^{n+1} = \frac{\mathbf{c}^{n}}{\Delta t} + \mathbf{q} \Rightarrow \mathbf{B}\mathbf{c}^{n+1} = \frac{\mathbf{c}^{n}}{\Delta t} + \mathbf{q} \Rightarrow \mathbf{c}^{n+1} = \mathbf{B}^{-1}\left[\frac{\mathbf{c}^{n}}{\Delta t} + \mathbf{q}\right]$$

Write the explicit and implicit components as

$$\mathbf{A}_E = [\Delta t\ \mathbf{A} + \mathbf{I}]^{-1}$$

$$\mathbf{A}_I = \mathbf{B}^{-1} = \left[\frac{\mathbf{I}}{\Delta t} - \mathbf{A}\right]^{-1}$$

and the combined time stepping equation can be written

$$\mathbf{c}^{n+1} = \mathbf{A}_I\left[\mathbf{A}_E \mathbf{c}^{n} + \mathbf{q}\right]$$

The explicit and implicit components of the transport matrix can each ran at different time step intervals. This allows a convenient separation of time scales for vertical versus horizontal advection-diffusion processes and provides for an improved efficiency of the model code.

4.4.3 Water column

In many instances, we are primarily interested only what happens in the vertical dimension of the ocean. Seasonal forcing such as light, stratification and the entrainment of nutrients from below the pycnocline as well as the sinking of detrital material and vertical migrations are strong drivers of community structure and ecosystem functions. To this end, we have developed a routine that extracts transport matrix elements for a vertical water column. This is greatly facilitated by the implicit/explicit decomposition of the matrix, as only the implicit components are relevant for vertical exchange.

Table 2: depth intervals for MITgcm_ECCO

bin	lower depth (m)
1	5
2	15
3	27.5
4	45
5	65
6	87.5
7	117.5
8	160
9	222.5
10	310
11	435
12	610
13	847.5
14	1160
15	1542.5
16	1975
17	2450
18	2950
19	3450
20	3950
21	4450
22	4950
23	5450

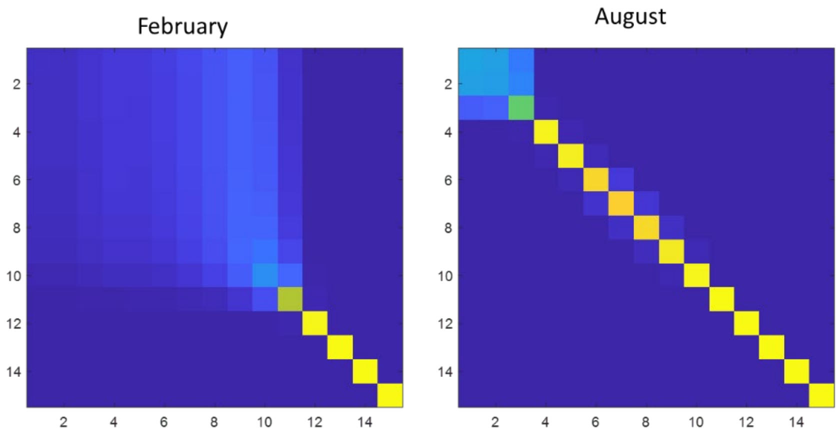


Figure 4: Implicit water column transport matrix elements extracted at latitude 60oN and longitude 10oW for February and august from the MITgcm_ECCO global simulation

Figure 4 shows an example of the transport matrix elements in the spring and summer in the North Atlantic (about 200 km west of the Faroe Islands). High values along the main diagonal (yellow boxes) indicates nearly all of the “released tracer” remains in the grid it was released from. That is there is very little mixing. In contrast, high mixing is characterized by smeared out blues (low values). In August, mixing only penetrates down to bin 3 whereas in the winter, mixing is much deeper; down to bin 11.

It should be noted here that the bin spacings in the vertical are not uniform. Table 2 provides a listing of the lower depth of the vertical bins in the MITgcm_ECCO transport matrix. For the example above, the summer mixed layer (down to bin 3) is in the range 30-40 m, while that during the winter is between 400-600 m deep.

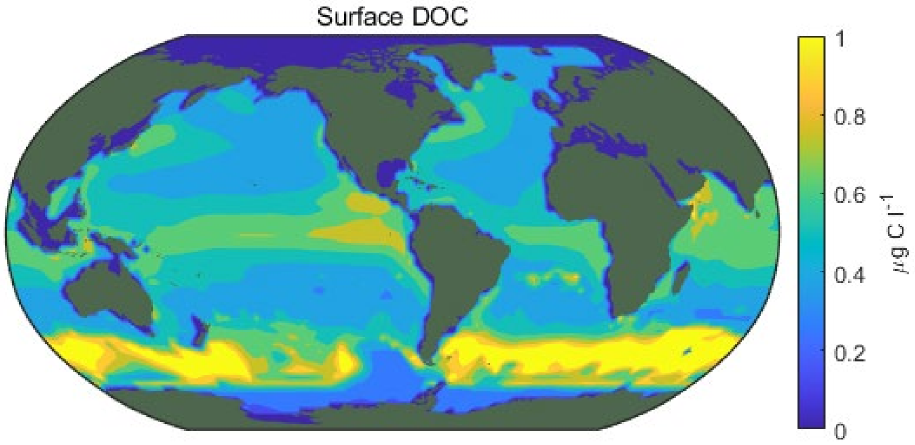
The implementation of various vertical transport processes, such as vertical migration and sinking of marine snow are much easier in the water column set up than in a full global simulation.

Every time a water column simulation is conducted at the same geographical location and using the same transport matrix model, the extracted water matrix is saved for future use. The algorithm for sorting these vertical exchange coefficients is as yet not very efficient, and using keeping saved copies speeds up the simulation considerably.

4.4.4 Global Model

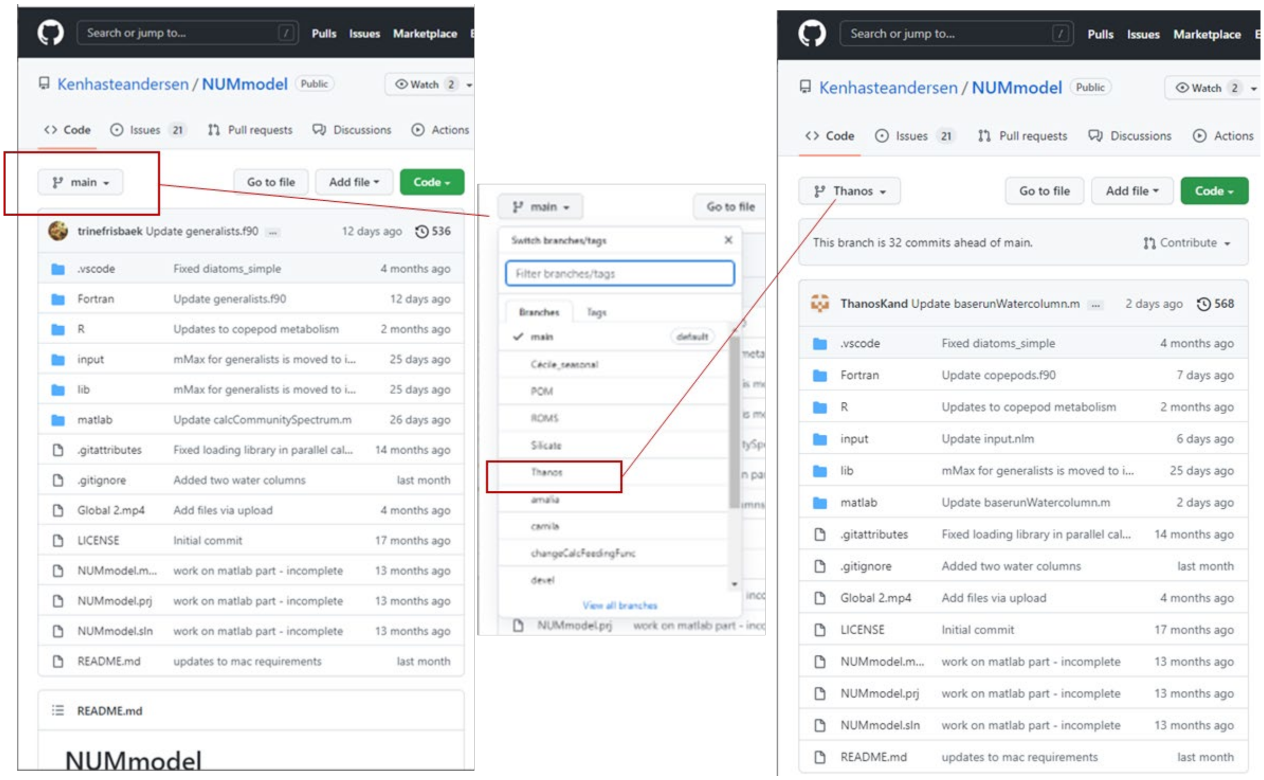
The global model utilizes the full power of the NUM model integrated with a Transport Matrix circulation simulation. The efficiency of simulation is greatly enhanced using parallel computing, and the code has been optimized with a parallelization in mind. To run the demonstration outlined below, you will need to install the Parallel Computing ToolBox in Matlab. With this in place, a global scale simulation of an annual cycle take a little over 5 minutes on an ordinary lap top computer.

The spin up time for the model is of some concern. To make this more efficient, we store the final state of all runs as potential initial conditions for future runs. This is done automatically, and matched simulation parameters and transport matrix specifications to any stored files that can be used as initial conditions.



5 Demonstration

The repository for the model code can be found at https://github.com/Kenhasteandersen/NUMmodel. This is an active site and currently runs about a dozen "branches" that are exploring different developments being made by various students, PhDs, postdocs and researchers. For the purposes of this demonstration, we will be using the "Thanos" branch.



trinefrisbaek Update generalists.f90 ... 12 days ago 536

.vscode	Fixed diatoms_simple	4 months ago
Fortran	Update generalists.f90	12 days ago
R	Updates to copepod metabolism	2 months ago
input	mMax for generalists is moved to i...	25 days ago
lib	mMax for generalists is moved to i...	25 days ago
matlab	Update calcCommunitySpectrum.m	26 days ago
.gitattributes	Fixed loading library in parallel cal...	14 months ago
.gitignore	Added two water columns	last month
Global 2.mp4	Add files via upload	4 months ago
LICENSE	Initial commit	17 months ago
NUMmodel.m...	work on matlab part - incomplete	13 months ago
NUMmodel.prj	work on matlab part - incomplete	13 months ago
NUMmodel.sln	work on matlab part - incomplete	13 months ago
README.md	updates to mac requirements	last month

This branch is 32 commits ahead of main.

ThanosKand Update baserunWatercolumn.m ... 2 days ago 568

.vscode	Fixed diatoms_simple	4 months ago
Fortran	Update copepods.f90	7 days ago
R	Updates to copepod metabolism	2 months ago
input	Update input.nlm	6 days ago
lib	mMax for generalists is moved to i...	25 days ago
matlab	Update baserunWatercolumn.m	2 days ago
.gitattributes	Fixed loading library in parallel cal...	14 months ago
.gitignore	Added two water columns	last month
Global 2.mp4	Add files via upload	4 months ago
LICENSE	Initial commit	17 months ago
NUMmodel.m...	work on matlab part - incomplete	13 months ago
NUMmodel.prj	work on matlab part - incomplete	13 months ago
NUMmodel.sln	work on matlab part - incomplete	13 months ago
README.md	updates to mac requirements	last month

5.1 Setting up

5.1.1 Installation

Accessing the library and running simulations requires a recent version of Matlab (2021 or later). On windows it requires the Matlab MEX module to be installed (Home -> Add-ons -> Get Add-ons -> MATLAB Support for MinGW-w64 C/C++ Compiler); on mac it requires Xcode to be installed. To run global simulation, it further requires that both the Mapping toolbox and the Parallel Computing toolbox are installed. Compiled versions of the library is available for windows (64 bit), linux and osx.

For those who wish to look deeper into the model, all of the source code for the Fortran library are also available. Re-compiling the library requires a Fortran compiler, e.g., gfortran. Use the makefile in the Fortran directory. Edit the compiler and flags in the makefile to suit your operating system and compile by writing: make. This is not a requirement for configuring, running or analyzing any of the simulations described below.

5.1.2 Connecting Matlab with GitHub folder

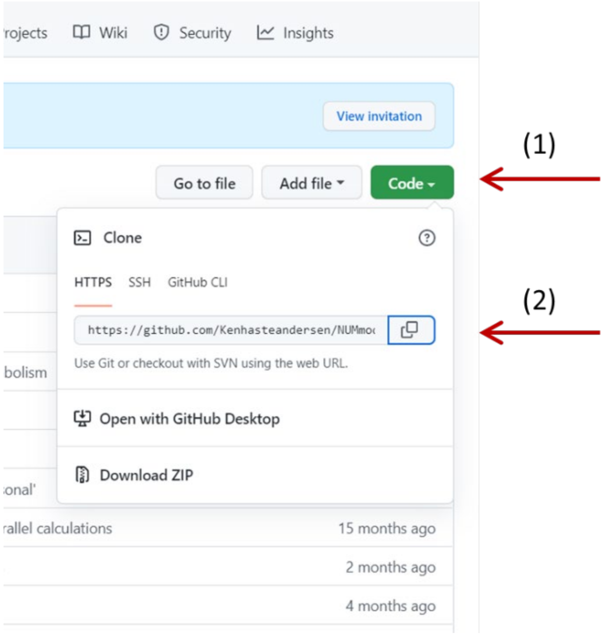
In Matlab, set up a folder where you would like to keep a clone of the NUM model.

Go to the GitHub site.

(1) Go to Code
(2) Then copy the https address for the code

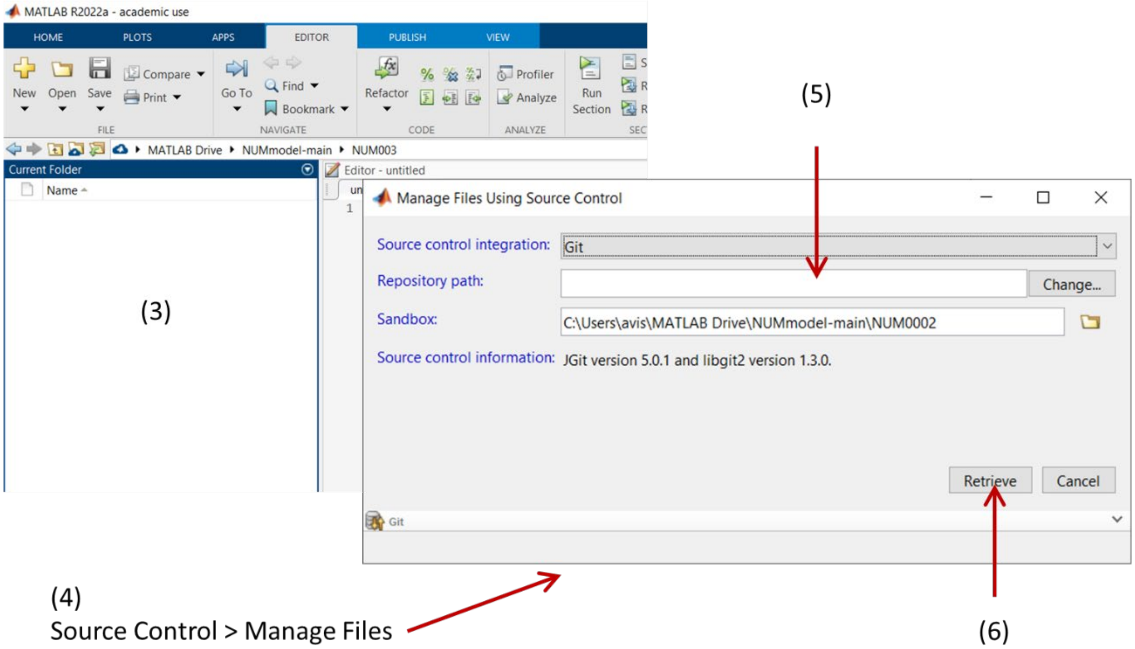
Go to your Matlab folder where you want the clone to appear.

(3) Right click in the file filed
(4) Boxes will appear. Choose Source Control and then Manage Files. The box `manage files` using source control will appear.
(5) Under source control integration choose Git. Then paste the https address you copied from GitHub into the Repository path.
(6) Press Retrieve



https://github.com/Kenhasteandersen/NUMmo

Use Git or checkout with SVN using the web URL.



Source control integration: Git

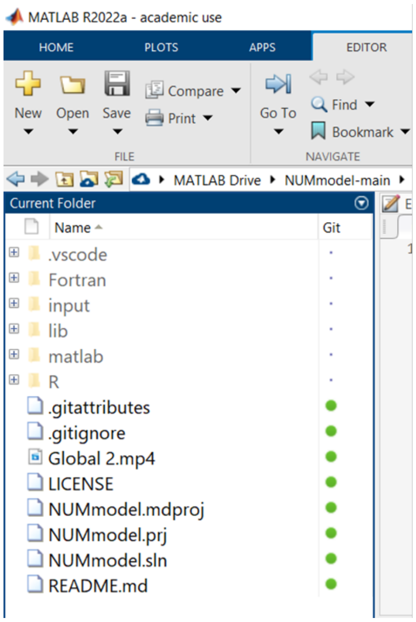
Repository path:

Sandbox: C:\Users\avis\MATLAB Drive\NUMmodel-main\NUM0002

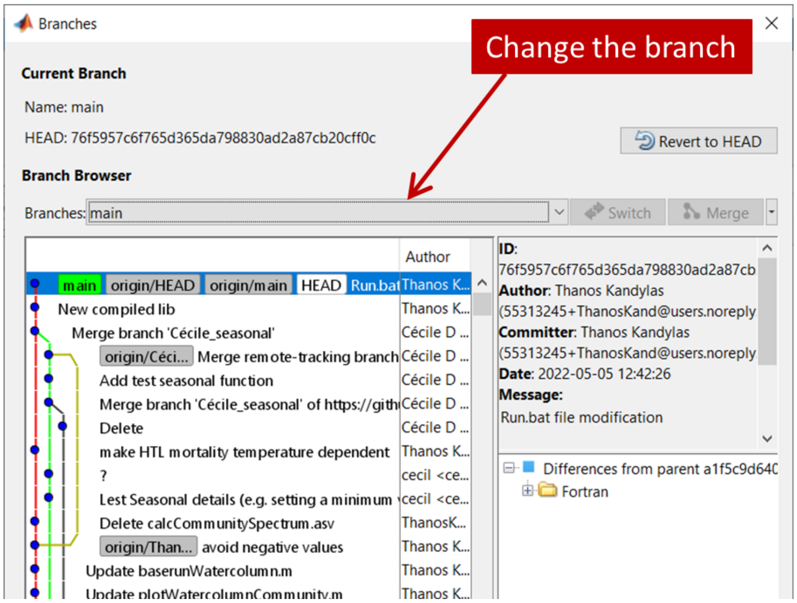
Source control information: JGit version 5.0.1 and libgit2 version 1.3.0.

(4)
Source Control > Manage Files

The retrieve process will take a bit of time, but when it’s done you are very nearly ready to go. The file folder should now be populated with files from the GitHub site, and look something like the left hand panel below. Mostly you will be working in the matlab folder for setting up and running simulations.



Name	Git
.vscode	·
Fortran	·
input	·
lib	·
matlab	·
R	·
.gitattributes	●
.gitignore	●
Global 2.mp4	●
LICENSE	●
NUMmodel.mdproj	●
NUMmodel.prj	●
NUMmodel.sln	●
README.md	●



Current Branch

Name: main

HEAD: 76f5957c6f765d365da798830ad2a87cb20cff0c

Branch Browser

Branches: main

	Author
main origin/HEAD origin/main HEAD Run.bat	Thanos K...
New compiled lib	Thanos K...
Merge branch 'Cécile_seasonal'	Cécile D ...
origin/Céci... Merge remote-tracking branch	Cécile D ...
Add test seasonal function	Cécile D ...
Merge branch 'Cécile_seasonal' of https://gith	Cécile D ...
Delete	Cécile D ...
make HTL mortality temperature dependent	Thanos K...
?	cecil <ce...
Lest Seasonal details (e.g. setting a minimum	cecil <ce...
Delete calcCommunitySpectrum.asv	ThanosK...
origin/Than... avoid negative values	Thanos K...
Update baserunWatercolumn.m	Thanos K...
Update plotWatercolumnCommunity.m	Thanos K...

ID: 76f5957c6f765d365da798830ad2a87cb
Author: Thanos Kandylas (55313245+ThanosKand@users.noreply
Committer: Thanos Kandylas (55313245+ThanosKand@users.noreply
Date: 2022-05-05 12:42:26
Message:
Run.bat file modification

One final option, you can change the branch of the code version that you are using. As mentioned before, this is an active site and there are several developments being made in parallel. These will become uploaded into the main branch when that are demonstrated to be functioning. For the purposes of this work, we use the Thanos branch, although it is in all likelihood that this will be merged with the main branch in the near future.

5.1.3 Basic structure

There are three levels of routines: top-level, medium-level and low-level. There are two model systems: an upper ocean represented as a chemostat and a global simulation with transport matrices.

5.1.3.1 Top-level matlab routines

These routines run a simulation and returns the results in a `sim` structure. All units are in µgC L^{-1} (or µgN L^{-1}). Units of light are µmol photons m^{-2} per s^{-1}:

- `baserunChemostat(mAdult)`. Runs a chemostat version of the model and plots the output. The argument is the adult body masses of copepods (in micro gram carbon) - send in an empty list to run only with unicellular plankton.
- `baserunChemostatEuler(mAdult)`. Uses the Fortran library and simple Euler time-stepping.
- `baserunGlobal()`. Runs a global simulation with only generalists.
- `baserunWatercolumn`. Run a water column extracted from a transport matrix.

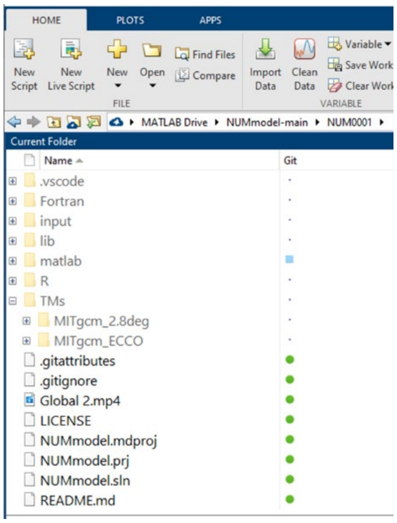
5.1.3.2 *Transport Matrix and file structure*

The simulation framework uses transport matrices which must be downloaded separately and placed in the directory `TMs`. Transport matrices must be downloaded from

http://kelvin.earth.ox.ac.uk/spk/Research/TMM/TransportMatrixConfigs

Choose MITgcm_ECCO. This will take some time to download. When downloaded, unzip and place in a folder named TMs in your main Matlab NUM directory. My folder looks as shown opposite.

Much of the simulation code is written in a Fortran library that is accessed through Matlab. There is also an option for using R as a front end, but will not be preseted here.



.vscode
Fortran
input
lib
matlab
R
TMs
MITgcm_2.8deg
MITgcm_ECCO
.gitattributes
.gitignore
Global 2.mp4
LICENSE
NUMmodel.mdproj
NUMmodel.prj
NUMmodel.sln
README.md

5.1.3.3 *Medium-level matlab routines*

The routines operate with two basic structures: a *parameter* structure and a *simulation* structure.

The parameter structure contains all parameters needed for a simulation. The simulation structure contains all the output, which can be used for analysis or for plotting.

Parameters are set with two calls: one to setup the size spectra to simulate and one to add the parameters for the simulation (chemostat or global). The size spectra are setup with a call to `setupXX` where XX represent the setup, e.g., `setupGeneralistsOnly` or `setupGeneric(mAdult)` (the latter includes copepods where `mAdult` is vector of copepod adult masses). Parameters for the simulation are subsequently set with a call to `parametersChemostat`or `parametersGlobal`. For example:

```
p = parametersChemostat( setupGeneralistsOnly() );.
```

Simulations are performed with calls to a simulation routine. For example:

```
        sim = simulationChemostat(p),
        sim = simulationWatercolumn(p, latitude, longitude),
        sim = simulationGlobal(p),
```

where `p` and `sim` are structures containing run parameters and simulation results respectively.

Plots are made with calls to the plot routines. A series of basic plots are made by a call to `plotSimulation(sim)`.

Some of these medium level routines are now packaged into accessible scripts. For instance

```
        mAdult = logspace(log10(0.2), log10(1000), 6)
        sim = baserunChemostat(mAdult)
```

runs a chemostat simulation for 6 classes of copepods whose asymptotic (adult) mass range logarithmically from 0.2 µgC to 1000 µgC.

5.2 Chemostat

The simplest representation of an ecosystem is that of a chemostat – where a well-mixed volume of water that has a constant through flow that both supplies nutrients and dilutes the plankton community. With a constant supply of sunlight, this allows a community of both unicellular and multicellular organism to become established.

The simplest simulation is a base run without metazoans. The code to run this is simply (cf appendix for code)

```
	sim = baserunChemostat
```

This produces the results depicted in Figure 5. All of the results as well as the configuration parameters can be found in the Matlab structure sim (Figure 6). The results include the full transients of the chemostat spin up process to indicate the transition to steady sate, and how stable this steady state is. This is also plotted in the out put but not included here for brevity.

The base physical parameters required for the chemostat simulation are temperature, light and dilution rate. For the simulation shown, temperature T = 10 °C, light L = 100 µmol photons $\mathrm{m}^{-2}\ \mathrm{sec}^{-1}$, and the dilution rate d = 0.01 day^{-1}.

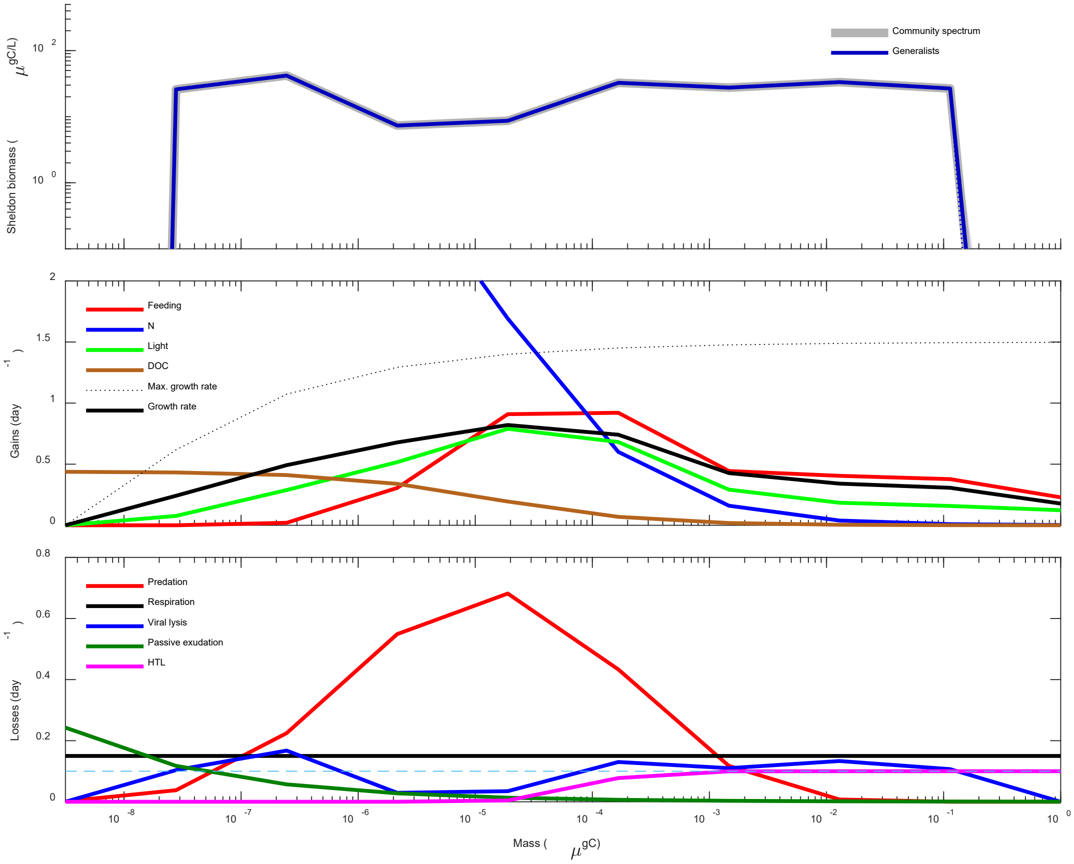
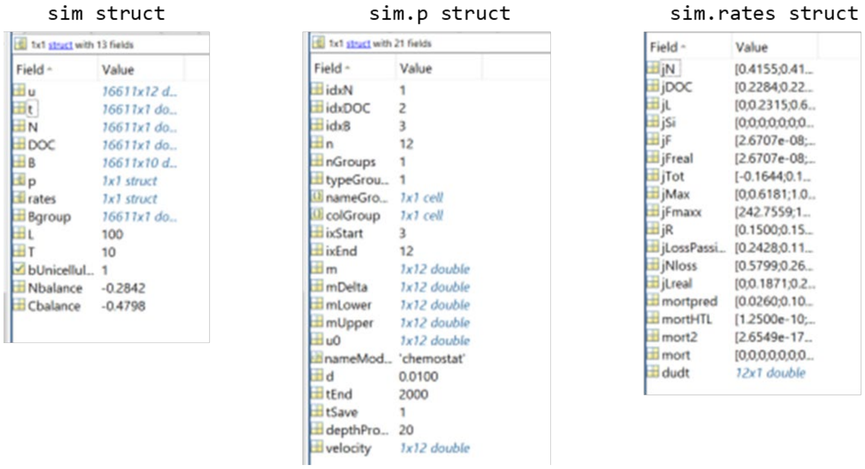


Figure 5: Results for baserunChemostat. This does not include multicellular organisms. The upper panel is the mass spectrum for the generalists. In this case it conforms very nearly to a classic Sheldon spectrum, i.e. very nearly uniform biomass per logarithmic interval of mass. The mid panel shows where these generalists derive their energy, (DOC, Light, Prey ingestion).



sim struct

1x1 struct with 13 fields

Field	Value
u	16611x12 d...
t	16611x1 do...
N	16611x1 do...
DOC	16611x1 do...
B	16611x10 d...
p	1x1 struct
rates	1x1 struct
Bgroup	16611x1 do...
L	100
T	10
bUnicellul...	1
Nbalance	-0.2842
Cbalance	-0.4798

sim.p struct

1x1 struct with 21 fields

Field	Value
idxN	1
idxDOC	2
idxB	3
n	12
nGroups	1
typeGrou...	1
nameGro...	1x1 cell
colGroup	1x1 cell
ixStart	3
ixEnd	12
m	1x12 double
mDelta	1x12 double
mLower	1x12 double
mUpper	1x12 double
u0	1x12 double
nameMod...	'chemostat'
d	0.0100
tEnd	2000
tSave	1
depthPro...	20
velocity	1x12 double

sim.rates struct

Field	Value
jN	[0.4155;0.41...
jDOC	[0.2284;0.22...
jL	[0;0.2315;0.6...
jSi	[0;0;0;0;0;0;0;0...
jF	[2.6707e-08;...
jFreal	[2.6707e-08;...
jTot	[-0.1644;0.1...
jMax	[0;0.6181;1.0...
jFmaxx	[242.7559;1...
jR	[0.1500;0.15...
jLossPassi...	[0.2428;0.11...
jNloss	[0.5799;0.26...
jLreal	[0;0.1871;0.2...
mortpred	[0.0260;0.10...
mortHTL	[1.2500e-10;...
mort2	[2.6549e-17...
mort	[0;0;0;0;0;0;0;0...
dudt	12x1 double

Figure 6: the output structures produced by sim = baserunChemostat. Model configuration parameters are contained in sim.p while specific rates are within sim.rates.

The same simulation, but now with 6 classes of metazoans can be runs with

```
        mAdult = logspace(log10(0.2), log10(1000), 6)
```

```
sim = baserunChemostat(mAdult)
```

where mAdult defines the maximum (adult) size of each metazoan class as being logarithmically spaced between 0.2 µgC and 1000 µgC. Results are depicted in Figure 7. These are for the steady state, i.e. after the populations have stabilized. Transients are depicted in Figure 8 and show that the model archives steady state rapidly.

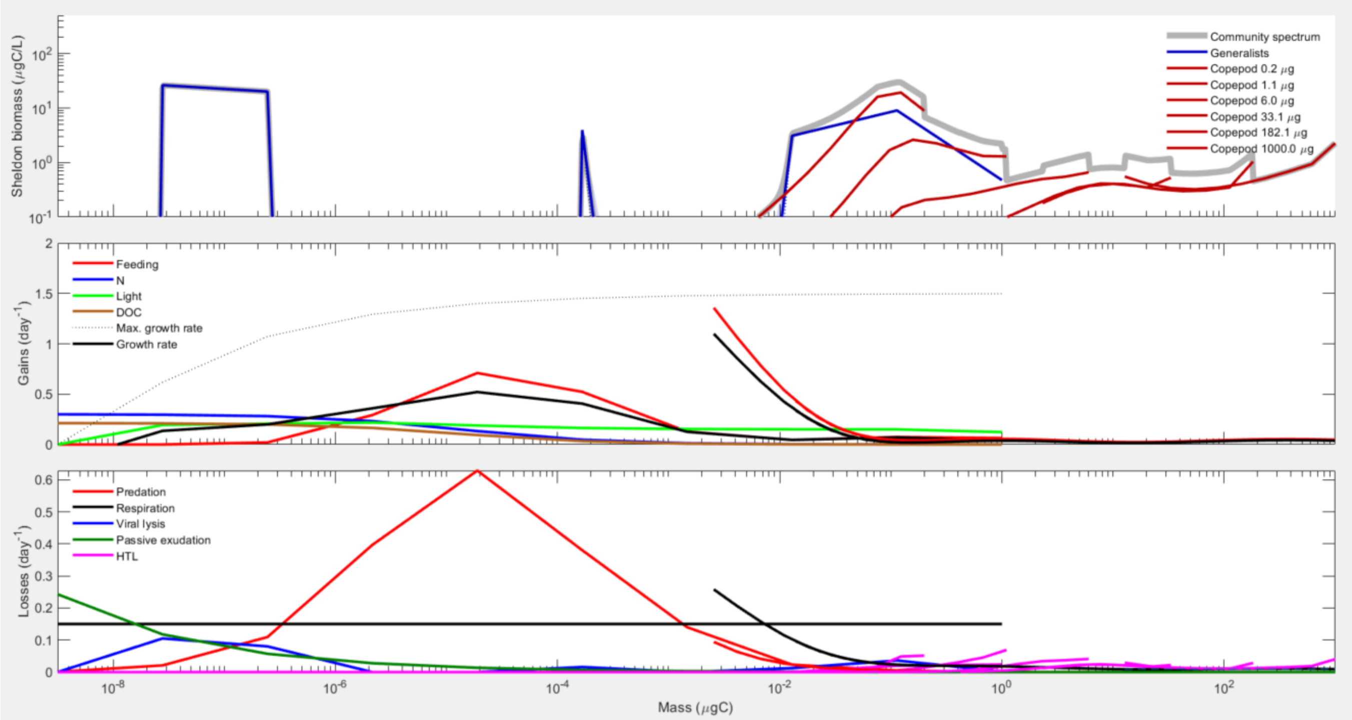


Figure 7: results for Chemostat simulation with 6 metazona classes. Upper panel shows biomass, mid panel the sources of energy (auto, mixo, herteotrophy) and the lower pan el the loss terms.

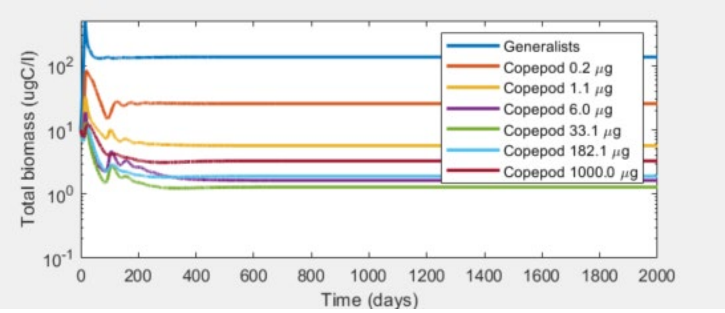
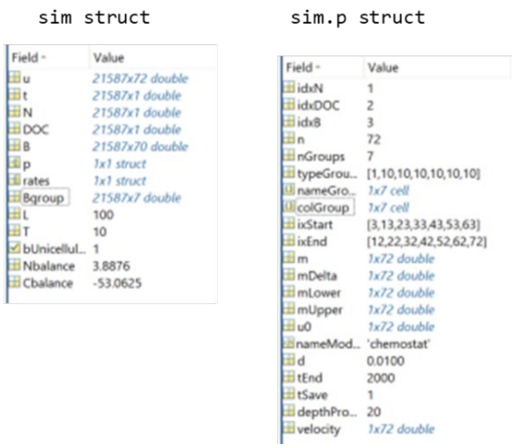


Figure 8: Transients to stable population configuration for chemostat simulation



sim struct

Field	Value
u	21587x72 double
t	21587x1 double
N	21587x1 double
DOC	21587x1 double
B	21587x70 double
p	1x1 struct
rates	1x1 struct
Bgroup	21587x7 double
L	100
T	10
bUnicellul...	1
Nbalance	3.8876
Cbalance	-53.0625

sim.p struct

Field	Value
idxN	1
idxDOC	2
idxB	3
n	72
nGroups	7
typeGrou...	[1,10,10,10,10,10,10]
nameGro...	1x7 cell
colGroup	1x7 cell
ixStart	[3,13,23,33,43,53,63]
ixEnd	[12,22,32,42,52,62,72]
m	1x72 double
mDelta	1x72 double
mLower	1x72 double
mUpper	1x72 double
u0	1x72 double
nameMod...	'chemostat'
d	0.0100
tEnd	2000
tSave	1
depthPro...	20
velocity	1x72 double

Figure 9: the output structures produced by sim = baserunChemostat (mAdult). Model configuration parameters are contained in sim.p

The results returned in the sim structure shown above. The size of the "biomass matrix" B (actually it includes other information as well) has increased and is now k x 72 where k is the number of time steps in the transients simulation. Importantly, the other dimension of this matrix has increased from the previous run to 72. The architecture of this matrix is
`B(:,1)` is nutrients
`B(:,2)` is dissolved organic carbon
`B(:,3:12)` is the 10 size classes of generalists
`B(:,13:22)` is the 10 size classes of the smallest metazoan (that with adult mass 0.2 µgC)
`B(:,23:22)` is the 10 size classes of the next larger metazoan (that with adult mass 1.1 µgC)
etc….
`B(:,63:72)` is the 10 size classes of the largest metazoan (that with adult mass 1000 µgC)

The architecture of this matrix will vary depending on other parameters. For instance, when including diatoms, the additional nutrient silicate is also included, as well as the number of classes of diatoms.

5.3 Water column

The base run for the water column setting (without metazoans) is initiated simply as

```
sim = baserunWatercolumn
```

The location of the chosen water column is at 60^{o}N and 10^{o}W (i.e. in the North Atlantic). If you want to change the location then enter

```
sim = baserunWatercolumn([], lat, lon)
```

where latitude `(lat)` is positive to the north and longitude `(lon)` is positive to the west with Greenwich as the prime meridian.

This produces a set of figures and a results structure sim. There are a number of figures, the most relevant being shown in Figs 10. 11. and 12.

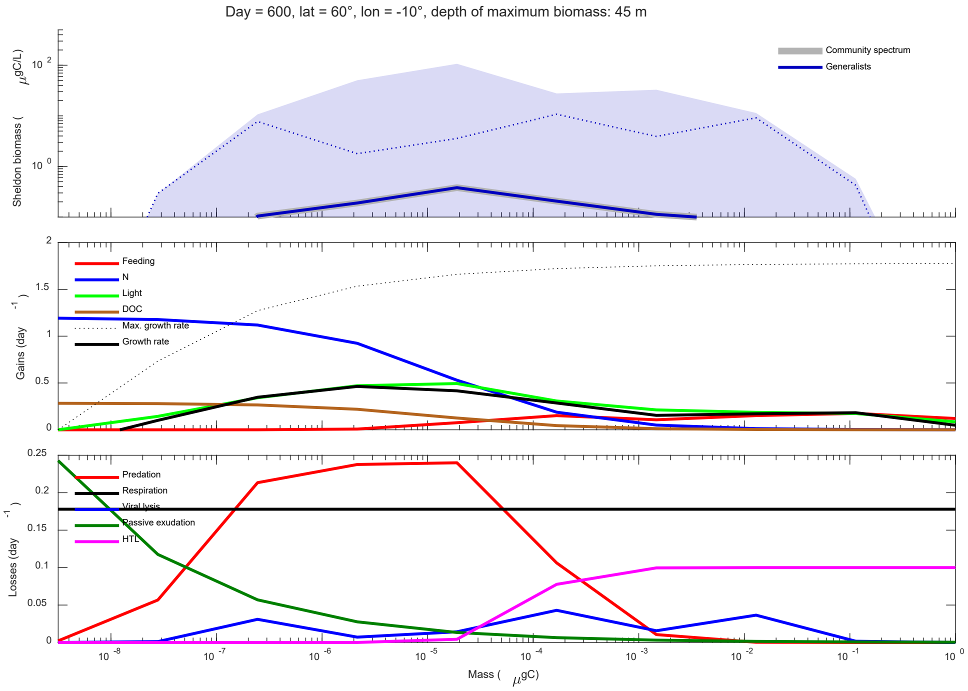


Figure 10: Results for baserunWatercolumn. This does not include multicellular organisms. The upper panel is the mass spectrum for the generalists at 45 m depth: the blue shaded area is an envelope for an annual cycle while the solid line is that seen on day 600 (i.e. around mid summer on the 2 year simulation). Other panels as explained above

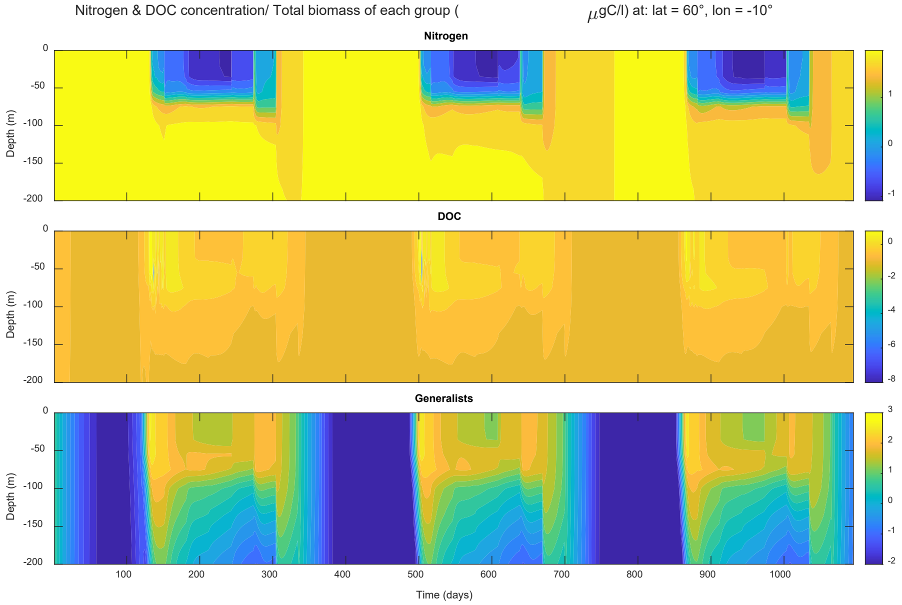


Figure 11:Depth time plots over the whole simulation period (3 years) for nitrogen, dissolved organic carbon and total generalist biomass. Clear seasonal cycles and the effect of stratification and vertical mixing appear in all 3 ecosystem components.

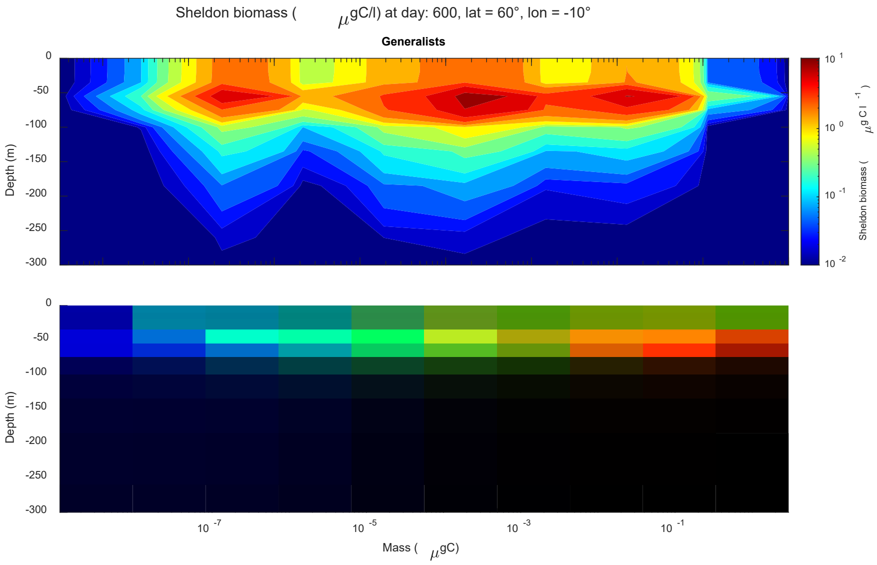


Figure 12: depth dependent size spectrum (upper panel) and trophic strategy (lower panel). The primary trophic strategies are coloured as primary colour blue = osmotriophy (uptake of DOC), green = photosynthesis, red = phagotrophy (ingestion of prey and particulate carbon). Intermediate colour (yellow, orange) indicate mixotrophy.

Emerging trophic strategies are plotted in Figure 12 as a function of cell mass, and depth. This is for day 600 in the simulation (i.e. around mid summer) and provides a "snap shot" of the functional diversity of the plankton community. It reveals a dominant subsurface community (around 50 m) composed of a variety of auto- hetero- and mixotrophic organisms. The smallest organisms (< 10^{-7} μgC) are osmotrophs, feeding on dissolved organic carbon. The mid-range (10^{-5} to 10^{-3} μgC) are phototrophs blending into osmo-mixotrophs at the small end and phago-mixotrophs at the larger end. Large organisms (> 10^{-1} μgC) are phagotrophs feeding on particulate organic carbon, both living and dead.

Results for the same simulation including diatoms is plotted in Figure 13. This can be run by setting the line

```
p = setupGeneralistsDiatoms_simple;
p = parametersWatercolumn(p);
p.tEnd = 1095;
```

into the `baserunWatercolumn` script (see Appendix 8.2). This is a simple representation of diatoms and assumes a fixed vacuole size (60% of the total cell volume). These results are from a simulation at 60ºN and 10ºW (North Atlantic) and span 3 years.

The current water column configuration is best suited to annual cycles of productivity and community structure for user defined locations in the pelagic ocean. This is largely in compliance with the available physical forcing fields (i.e., Transport Matrices) that are currently available There general NUM model structure though is not restricted to these scales and has been demonstrated to be equally applicable to more focused temporal and spatial scales. These will be explored in ECOTIP with regional scale physical forcing and transport matrices for climate change scenarios in the sub-Artic north Atlantic.

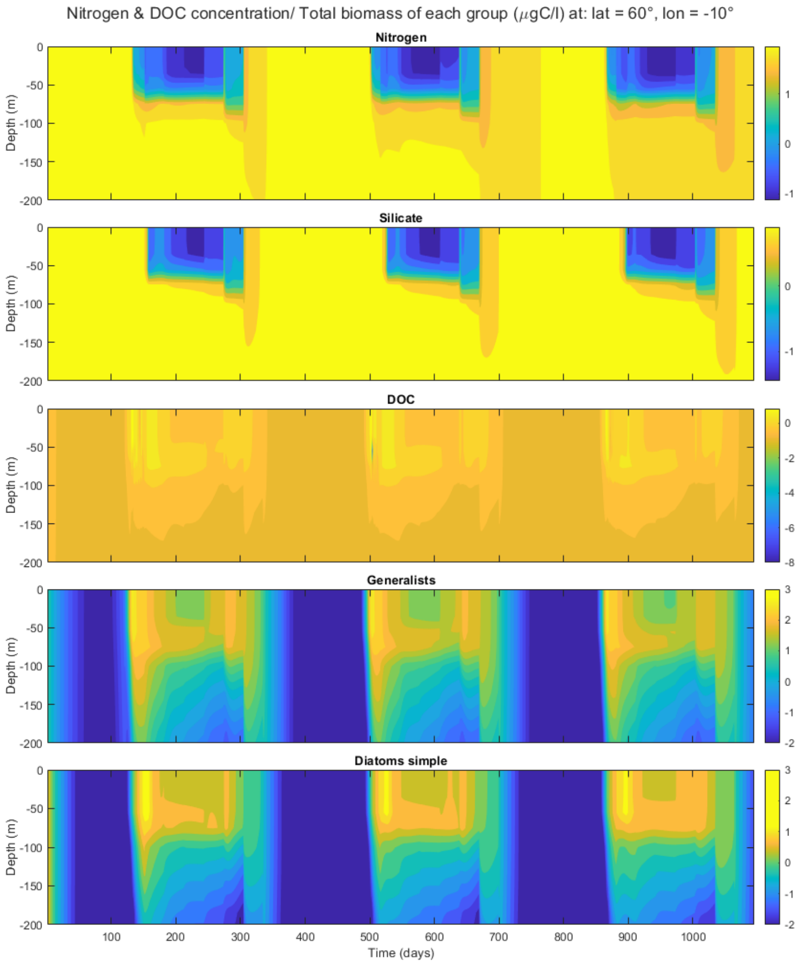


Figure 13: Results for baserunWatercolumn including diatoms with a vacuole volume fixed at 60% of the total cell volume. The upper panels depict seasonal cycles of nitrogen, silicate and dissolved organic carbon. The lower panes the biomass [µgC L^{-1}] of generalist and diatoms respectively.

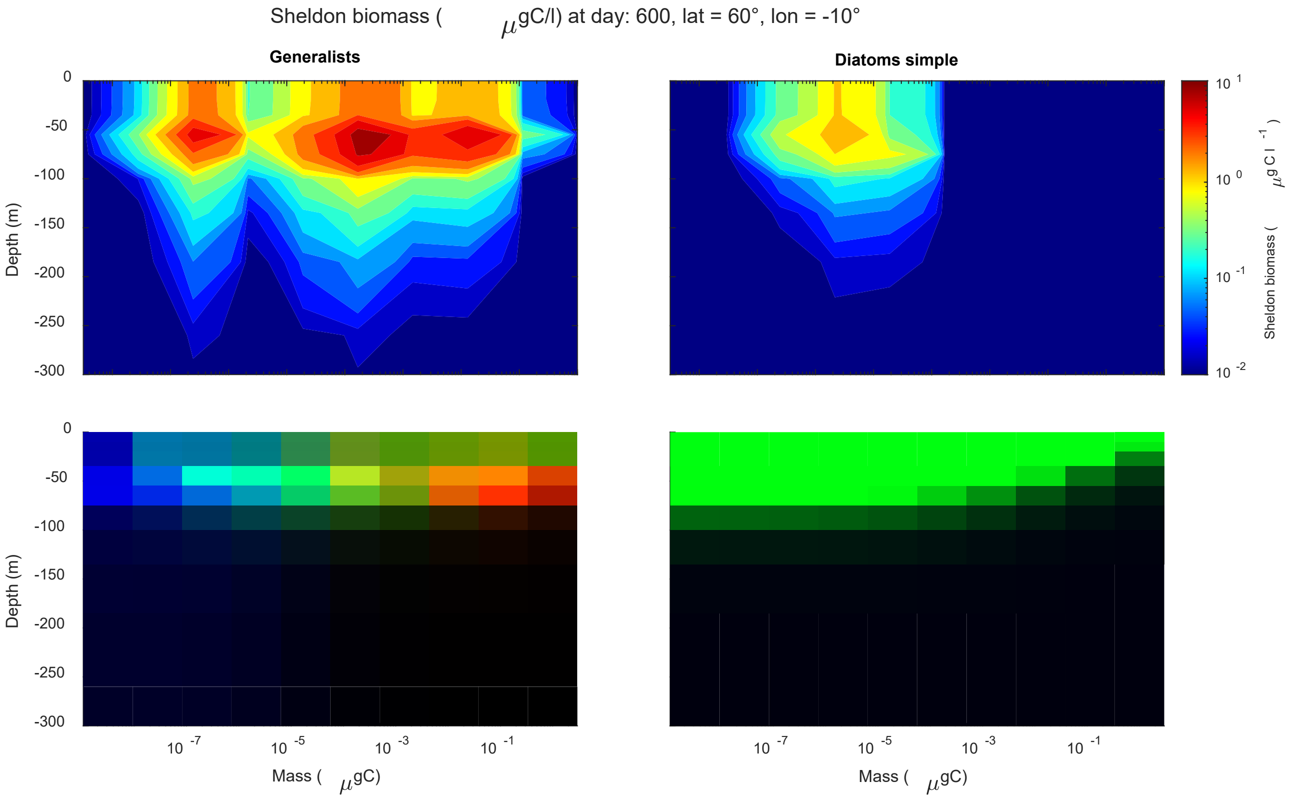


Figure 14: Results for baserunWatercolumn including diatoms with a vacuole volume fixed at 60% of the total cell volume. Upper row biomass and lower row trophic strategy. Trophic strategy coloured as primary colour blue = osmotriophy (uptake of DOC), green = photosynthesis, red = phagotrophy (ingestion of prey and particulate carbon). Intermediate colour (yellow, orange) indicate mixotrophy. As might be expected, diatoms are pure phototrophs.

As a final example for the water column simulation, we add 6 classes of metazoans as was also done with the chemostat. The code for this is:

```
    mAdult = logspace(log10(0.2), log10(1000), 6)
    sim = baserunWatercolumn(mAdult)
```

The location is the same as before; 60^{o}N and 10^{o}W in the North Atlantic. This produces a series of plots, the most important of which are shown in Figure 15, Figure 16 and Figure 17. An example of the community size spectrum, gains and losses are presented in Figure 15.

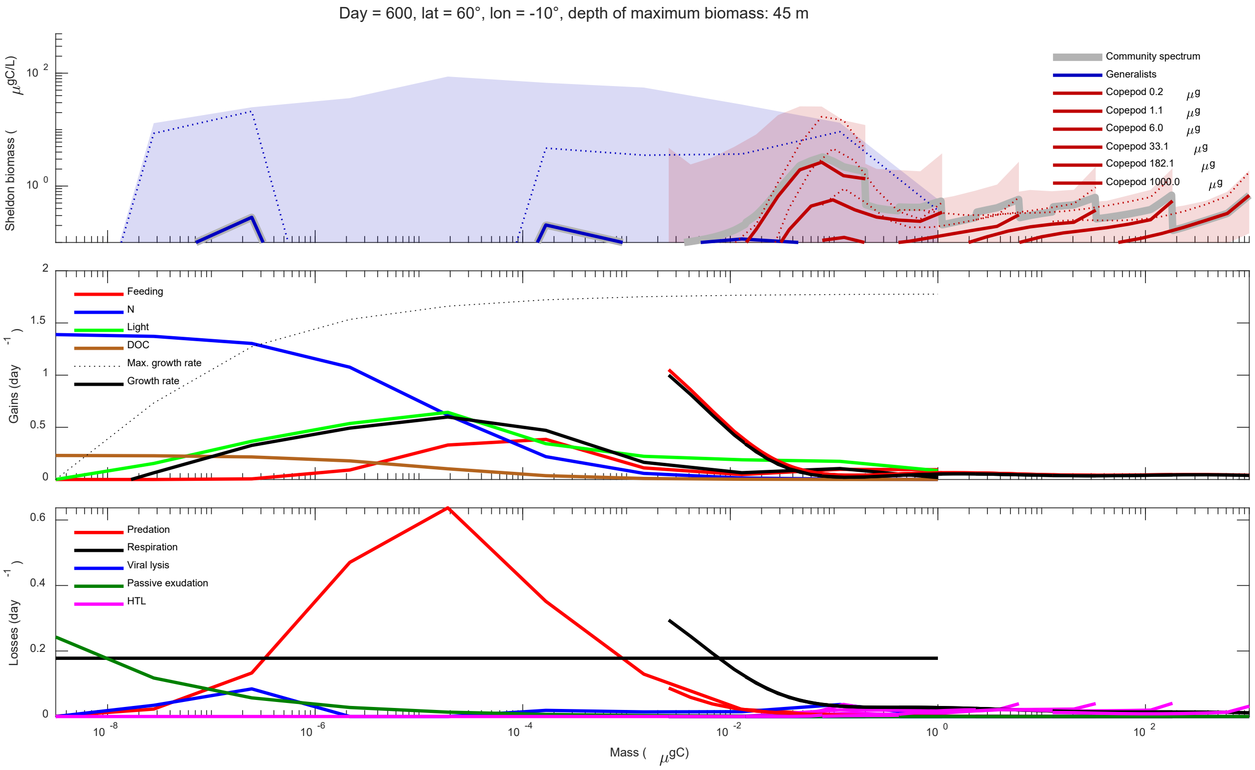


Figure 15: Water column simulation at 60^{o}N and 10^{o}W including 6 classes of metazoans. Sheldon biomass spectrum for generalists (blue) and copepods (red) (upper panel), gains (mid panel) and losses (lower panel) on day 600 of the simulation.

New features appear in the Sheldon spectrum (upper panel Figure 15) for the 6 copepod populations (the red lines) on day 600 (mid summer) of the simulation. Each line represents a single copepod population and its size (age) structure. These are summed to give the total copepod biomass spectrum (the grey line). The red shaded region represents the range of each population over an annual cycle.

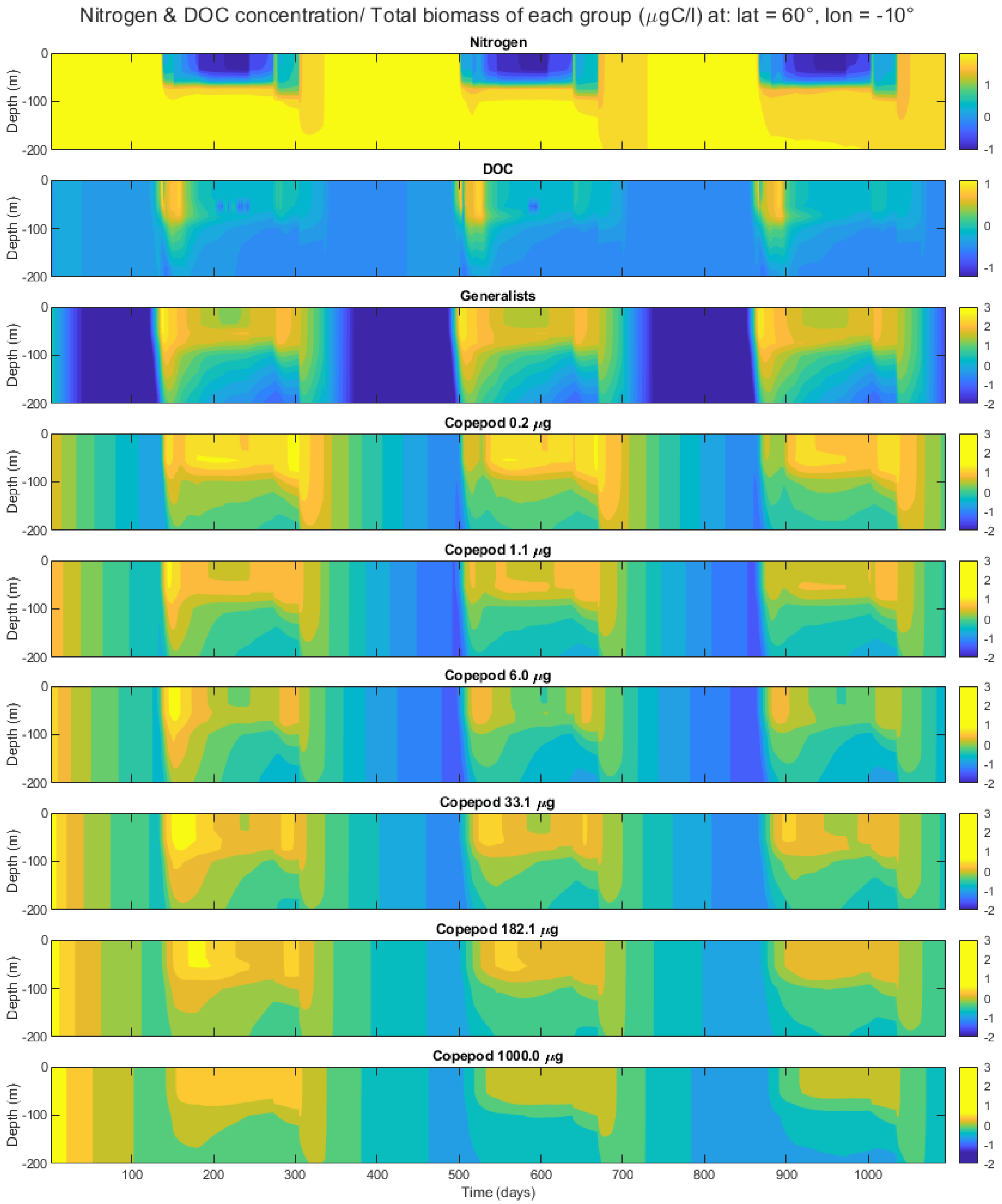


Figure 16: Water column simulation at 60ºN and 10ºW including 6 classes of metazoans. Upper panels Nitrogen, Silicate and generalist biomass. Lower panels, total biomass (summed over all sizes of individuals) for each copepod class defined by its adult mass.

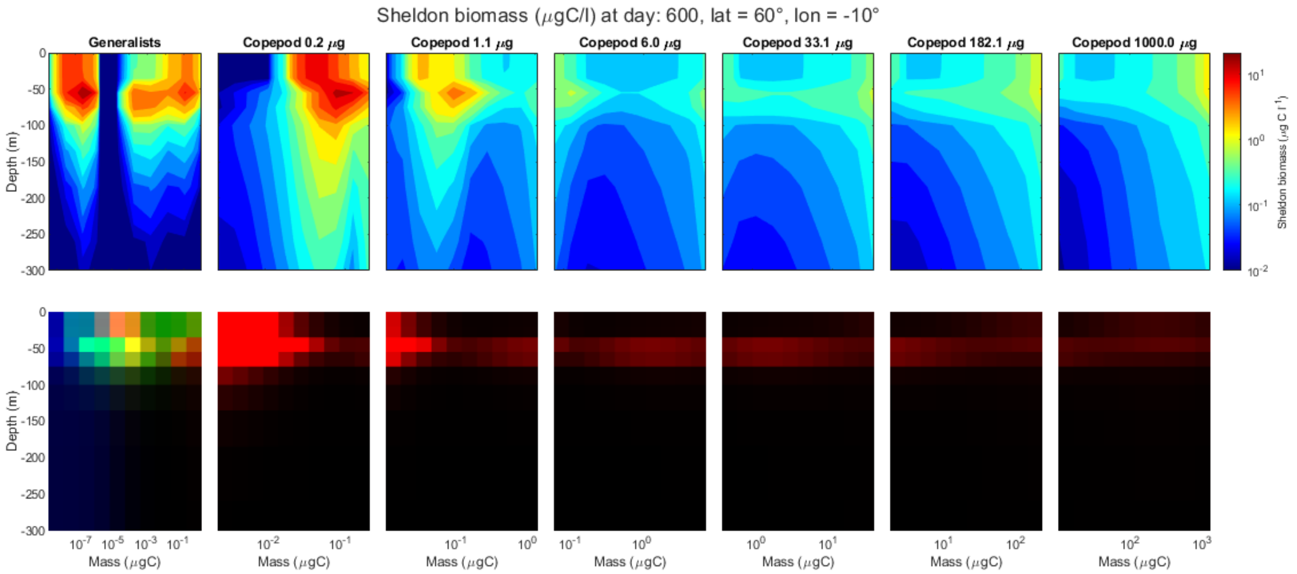


Figure 17: Depth mass distributions of biomass (upper row) and trophic strategy (lower rows) in a water column simulation including 6 classes of metazoans. Location is at 60ºN and 10ºW and results plotted for on day 600 of the simulation. Trophic strategy coloured as primary colour blue = osmotriophy (uptake of DOC), green = photosynthesis, red = phagotrophy (ingestion of prey and particulate carbon). Intermediate colour (yellow, orange) indicate mixotrophy. As might be expected, copepods are pure pahgotrophs.

The time evolution of nutrients and biomass is shown in Figure 16 over a 3 year period. It is clear at at this location, there is a strong influence of seasonality on annual cycles. Nutrients are significantly drawn down during spring and summer and are remixed to the surface in the autumn and winter. There is a clear surface spring bloom of generalists and a subsequent subsurface production at depth (around 50 m) during the summer. This is followed by a surface autumn bloom. These is a high production of DOC associated with the later phases of the spring bloom when nutrient concentrations in the surface are in decline.

The copepod community follows unicellular production. There is a succession of copepods according to their size, with the population of the smallest peaking earlier than larger. This is particularly clear in the last year of the simulation.

There are a few issues illustrated here that have not been fully resolved and are the subject of ongoing development. Firstly, it is clear from Figure 16 that the simulation has not reached steady state – there is for instance a relatively large difference in results between the last 2 years. A longer simulation could help here. More important with regards copepod populations, we have not as yet implemented seasonal vertical migration which is know to be a significant feature of copepod life cycles in temperate and polar seas. This leads to relatively high biomass, particularly of large copepods in the surface ocean during the winter. We have developed an overwintering algorithm that is based on the size of individuals and their capacity to store sufficient energy to survive diapause (Visser et al. 2020). This has not yet been fully implemented. Size based metabolic costs, lack of feeding and subsequent mortality is active during the winter in the current version of the model. That is, in general the population dynamics are sound, even though the plotted results are somewhat unsatisfying with regards what is observed.

We expect improved representation of observed dynamics as more process knowledge (e. g., vertical migration behaviour) is fully implemented.

Finally, Figure 17 shows a snap shot of the size depth distribution of the biomass and trophic strategy of the plankton community in the middle of the summer. There is a well developed unicellular community engaged in auto- mixo- and heterotrophy. The copepod community is dominated by smaller size classes. There is some evidence of cohorts passing through the copepod populations. The adult mass 0.2 µgC class for instance is almost entirely dominated by adults whereas the 1.1 µgC class is dominated by smaller life stages. Both of these are responsible for the heavy grazing pressure on the low to mid-range sized generalists leading to the conspicuous hole in their biomass spectrum. An animation of the copepod biomass spectrum shows clear sweeps of cohorts through the populations.

We are in the process of developing a suite of plotting routines to better analyse and visualize the community structure and succession of the metazoan community.

The tropic strategies for all copepods (Figure 17) is phagotrophy unsurprisingly, as they can neither photosynthesise nor take up DOC. We have implemented a version of the model where copepods have either an active (i.e. feeding current or cruise) or passive (i.e. ambush) feeding mode ((Serra-Pompei et al. 2020, Serra-Pompei et al. 2022)). There are clear trade-offs between these feeding modes as active feeding is energetically more costly, riskier but has a higher foraging efficiency than passive feeding (Mariani & Visser 2010, Mariani et al. 2013). This has been parameterized and validated but has not yet been included in the core library of the NUM model. In part this is because the treatment is somewhat cosmetic. True, there are well defined trade-offs, but more importantly, there is the possibility of direct impact on the unicellular community. Specifically, unicellular consumers? exhibit a range of motile abilities – very much with similar trade-offs that partition copepods along an active – passive feeding mode axis. The key trophic dynamic here is that passive feeding copepods tend to only encounter motile prey, while active feeding copepods only encounter immotile prey as motile prey can detect and escape their advances (Kenitz et al. 2017, Prowe et al. 2019). We are currently advancing this dimension of plankton trait-space into the NUM model.

5.4 Global

A relatively fast global simulation can be run simply with the instruction

```
	sim = baserunGlobal
```

This produces the plotted results shown in Figure 18 showing DOC (upper panel) nitrate (middle panel) and generalist biomass (lower panel). All of these are for the surface layer (the upper layer of the transport matrix) and in this case for January. At this time of year, the general picture is as expected with surface nutrients high in the northern hemisphere and low in the southern hemisphere. Unicellular biomass is relatively high throughout the Southern Ocean and along the equatorial upwelling zone. There are also some elevated peaks associated with the Western boundary currents in the Northern hemisphere.

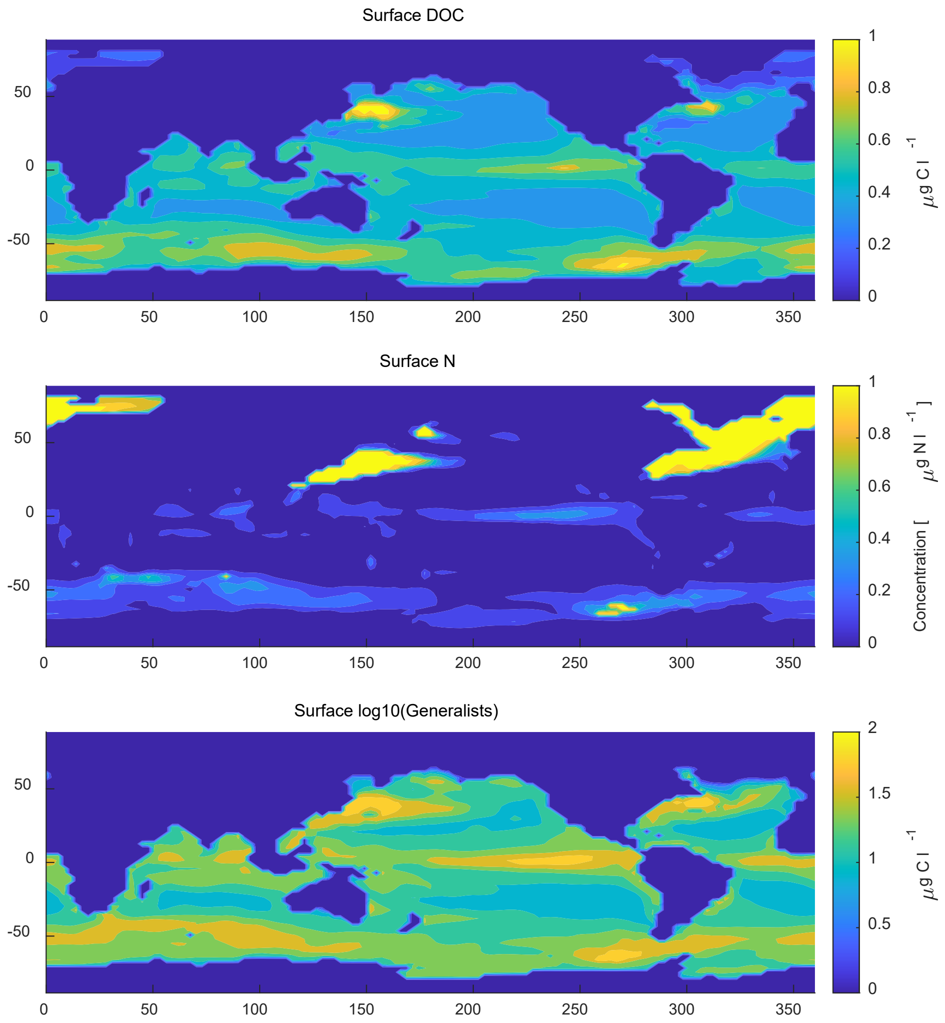


Figure 18: Global simulation. Simple and fast plotting using baserunGlobal. Plot is for January.

The results in the `sim` structure contain results for a full annual cycle. For instance, the matrix `sim.N` is a 4 dimensional matrix; in this case dimensioned as 128 x 64 x 15 x 12. The first ordinate is the east-west grid position (contained in `sim.x`), the second the north-south grid position (`sim.y`) the third the depth (`sim.z`) and the fourth the time evaluation. This latter is monthly covering the full year.

To plot different month, and/or a different map projection use

```
        plotGlobal(sim, iTime, sProjection)
```

where `iTime` is the month ordinate (e.g. 1 = January) while `sProjection` is a text string indicating the map projection. For instance, the plots in Figure 19 were produced using

```
        plotGlobal(sim, 1, ‘robinson’)
        plotGlobal(sim, 7, ‘robinson’)
```

There are several other map projections available. To access these, you will have to use the Mapping Toolbox installed on Matlab where you can find an exhaustive list of different map projections.

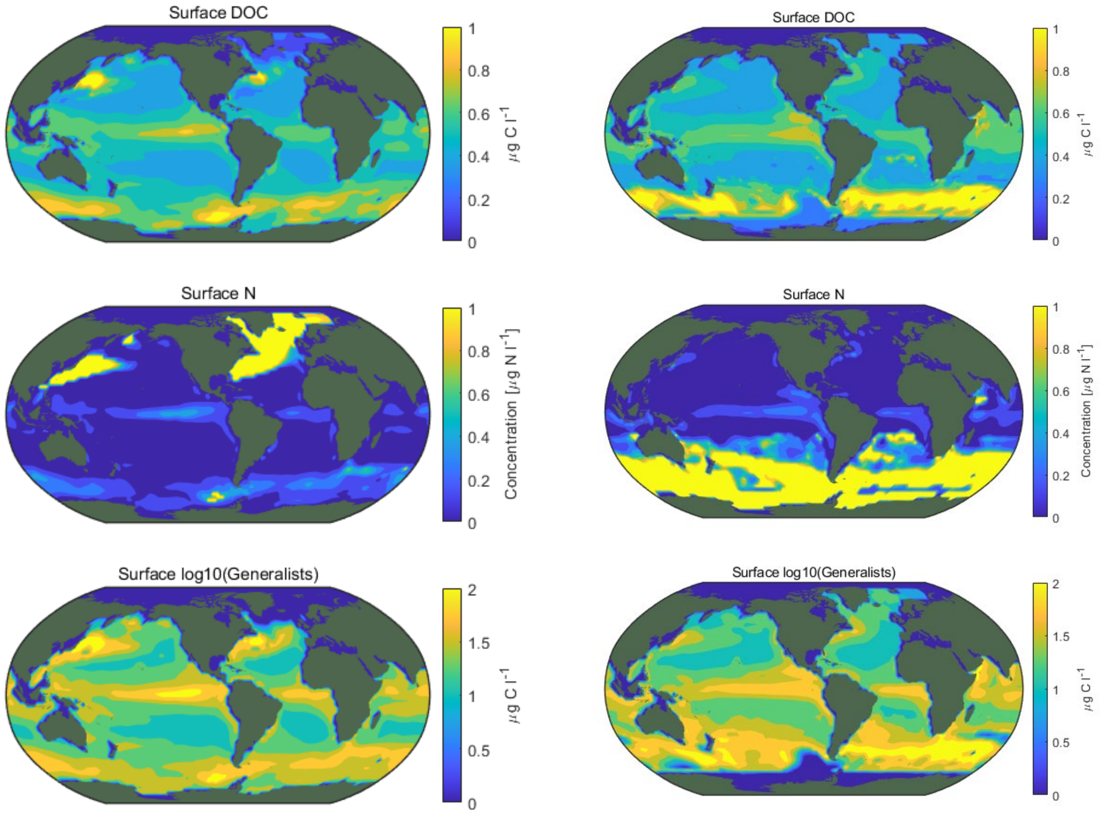


Figure 19: Simulation using baserunGlobal for January (left column) and July (right column) using the 'robinson' map projection.

The seasonal shifts in surface nitrate are clearly illustrated here, with high – low concentrations seen in the northern – southern hemisphere in January – July. The picture is not as clear for biomass and dissolved organic carbon, but it should be noted that these are for the surface ocean – to a depth of 10 m – and much of the primary production in mid summer (July in the north and January in the south) takes place sub surface (cf Figure 13 and Figure 16 for instance).

Finally, in Figure 20 we present the equivalent model run including diatoms. This is available with the same statement as above, i.e.

```
sim = baserunGlobal
```

but with the code (cf Appendix 8.3) slightly changed to read:

```
if (nargin==0)
    % p = setupGeneralistsOnly(10, true); % Use 10 size groups and parallel execution
    p = setupGeneralistsDiatoms_simple(10, true);
    p = parametersGlobal(p); % Use standard low-res model
    % p = parametersGlobal(10,2); % Use MITgcm_ECCO
    p.tEnd = 365;
end
```

changing the setup for generalists only to generalist and diatoms.

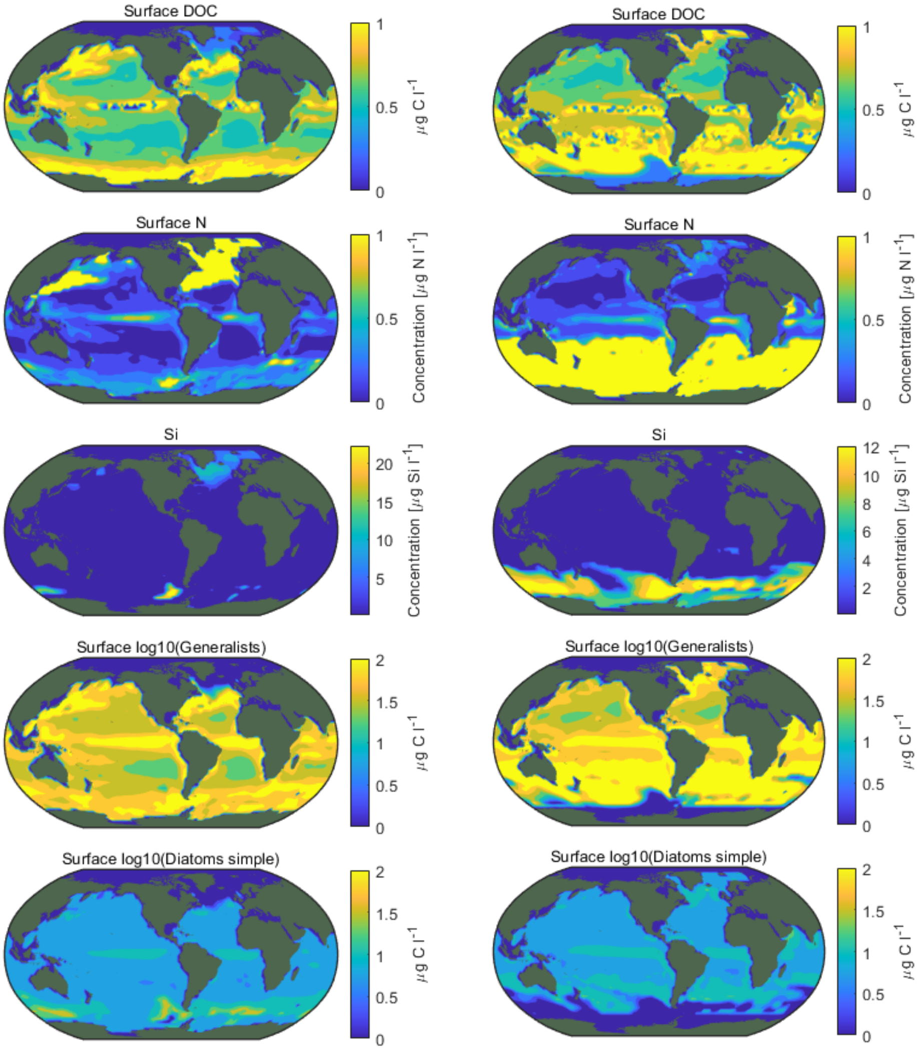

Surface DOC
Surface DOC
μg C l⁻¹
1
0.5
0
Surface N
Surface N
Concentration [μg N l⁻¹]
Si
Si
Concentration [μg Si l⁻¹]
20
15
10
5
12
10
8
6
4
2
Surface log10(Generalists)
Surface log10(Generalists)
2
1.5
1
0.5
0
Surface log10(Diatoms simple)
Surface log10(Diatoms simple)

Figure 20; Simulation using baserunGlobal for January (left column) and July (right column) using the 'robinson' map projection including diatoms and silicate.

As before, there is a clear seasonal variation in surface nutrients, both nitrates and silicate. The latter is particularly pronounced in the Southern Ocean and it is here where these is a large region where diatoms exhibit seasonal dynamics. These variations are not as conspicuous in the Northern hemisphere.

6 Conclusion

A demonstration of the NUM model with a full coverage of the unicellular and multicellular components of the plankton community is presented here. These are implemented in a suite of different environmental settings, from a highly idealized chemostat to a realistic water column and finally to a full global simulation. The model is integrated into a transport matrix representation of the physical environment (chiefly circulation and mixing, but also temperature and salinity fields). This ability will be exploited in the latter phase of ECOTIP to run climate change scenarios on a regional scale in the sub-polar North Atlantic. The modelling framework will also be integrated into a higher trophic level model of fish. The key difference in the model treatment of metazoans in NUM and fish is the dominant role that migrations play in the life history and population dynamics of fish. In this we will integrate the NUM model with FEISTY – a trait and size based model of fish – essentially divided into functional type (forage fish, large pelagic fish, and demersal) and their age structure.

The NUM modelling framework presented here, while relatively complete, is a project of ongoing development. This document presents a guide for its implementation that is freely accessible to researchers around the world. While examples given demonstrate the scope of the modelling framework, they only scratch the surface of the model's full capabilities. There is an active community at DTU Aqua, within ECOTIP and beyond which is engaged in developing the modelling framework (e.g. vertical fluxes of organic material, vertical migration of metazoans, expanded trait axes of the microbial community, integrating with fisheries production models), and applying it to describing various processes from geographic patterns of biodiversity, global biogeochemical cycles and global change.

7 References

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.

Baird ME, Suthers IM (2007) A size-resolved pelagic ecosystem model. Ecol Model 203:185–203.

Banas NS (2011) Adding complex trophic interactions to a size-spectral plankton model: Emergent diversity patterns and limits on predictability. Ecol Model 222:2663–2675.

Behrenfeld MJ, Boss ES (2014) Resurrecting the ecological underpinnings of ocean plankton blooms. Annu Rev Mar Sci 6:167–194.

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

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

Castellani M, Rosland R, Urtizberea A, Fiksen Ø (2013) A mass-balanced pelagic ecosystem model with size-structured behaviourally adaptive zooplankton and fish. Ecol Model 251:54–63.

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

De Roos AM, Schellekens T, Van Kooten T, Van De Wolfshaar K, Claessen D, Persson L (2008) Simplifying a physiologically structured population model to a stage-structured biomass model. Theor Popul Biol 73:47–62.

Evans GT, Parslow JS (1985) A model of annual plankton cycles. Biol Oceanogr 3:327–347.

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.

Fuchs HL, Franks PJ (2010) Plankton community properties determined by nutrients and size-selective feeding. Mar Ecol Prog Ser 413:1–15.

Fulton EA (2010) Approaches to end-to-end ecosystem models. J Mar Syst 81:171–183.

Gentleman W (2002) A chronology of plankton dynamics in silico: how computer models have been used to study marine ecosystems. Hydrobiologia 480:69–85.

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

Henriksen CI, Saiz E, Calbet A, Hansen BW (2007) Feeding activity and swimming patterns of Acartia grani and Oithona davisae nauplii in the presence of motile and non-motile prey. Mar Ecol Prog Ser 331:119–129.

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

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, Sabatini M (1995) Scaling of fecundity, growth and development in marine planktonic copepods. Mar Ecol Prog Ser:285–298.

Longhurst AR (1995) Seasonal cycles of pelagic production and consumption. Prog Oceanogr 36:77–167.

Mariani P, Andersen KH, Visser AW, Barton AD, Kiørboe T (2013) Control of plankton seasonal succession by adaptive grazing. Limnol Oceanogr 58:173–184.

Mariani P, Visser AW (2010) Optimization and emergence in marine ecosystem models. Prog Oceanogr 84:89–92.

Mauchline J (1998) Adv. Mar. Biol. 33: The biology of calanoid copepods.

May RM (1973) Time-delay versus stability in population models with two and three trophic levels. Ecology 54:315–325.

McCauley E, Murdoch WW (1987) Cyclic and stable populations: plankton as paradigm. Am Nat 129:97–121.

Mitra A, Castellani C, Gentleman WC, Jónasdóttir SH, Flynn KJ, Bode A, Halsband C, Kuhn P, Licandro P, Agersted MD, Lindeque PK, Koppelmann R, Møller EF, Gislason A, Nielsen TG, St John M (2014) Bridging the gap between marine biogeochemical and fisheries sciences; configuring the zooplankton link. Prog Oceanogr 129:176–199.

Prowe AF, Visser AW, Andersen KH, Chiba S, Kiørboe T (2019) Biogeography of zooplankton feeding strategy. Limnol Oceanogr 64:661–678.

Record N, Pershing A, Maps F (2013) Emergent copepod communities in an adaptive trait-structured model. Ecol Model 260:11–24.

Roff JC, Turner JT, Webber MK, Hopcroft RR (1995) Bacterivory by tropical copepod nauplii: extent and possible significance. Aquat Microb Ecol 9:165–175.

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 189:102473.

Serra-Pompei C, Ward BA, Pinti J, Visser AW, Kiørboe T, Andersen KH (2022) Linking plankton size spectra and community composition to carbon export and its efficiency. Glob Biogeochem Cycles:e2021GB007275.

Small L, Fowler S, Ünlü M (1979) Sinking rates of natural copepod fecal pellets. Mar Biol 51:233–241.

Sprules WG, Barth LE (2016) Surfing the biomass size spectrum: some remarks on history, theory, and application. Can J Fish Aquat Sci 73:477–495.

Titelman J (2001) Swimming and escape behavior of copepod nauplii: implications for predator-prey interactions among copepods. Mar Ecol Prog Ser 213:203–213.

Titelman J, Kiørboe T (2003) Motility of copepod nauplii and implications for food encounter. Mar Ecol Prog Ser 247:123–135.

Turner JT (2004) The importance of small planktonic copepods and their roles in pelagic marine food webs. Zool Stud 43:255–266.

Visser AW, Brun P, Chakraborty S, Spaanheden Dencker T, Daniël van Denderen P, van Gemert R, van Someren Gréve H, Heilmann I, Wejlemann Holm M, Huld Jónasdóttir S, Kenitz KM, Kiørboe T, Lindegren M, Mariani P, Tor Nielsen L, Pancic M, Payne M, Pécuchet L, Azaña Schnedler-Meyer N, Høgsbro Thygesen U, Törnroos A, Andersen KH (2020) Seasonal strategies in the world’s oceans. Prog Oceanogr:102466.

Visser AW, Fiksen Ø (2013) Optimal foraging in marine ecosystem models: Selectivity, profitability and switching. Mar Ecol Prog Ser 473:91–101.

Ward BA, Dutkiewicz S, Jahn O, Follows MJ (2012) A size-structured food-web model for the global ocean. Limnol Oceanogr 57:1877–1891.

Ward BA, Follows MJ (2016) Marine mixotrophy increases trophic transfer efficiency, mean organism size, and vertical carbon flux. Proc Natl Acad Sci 113:2958–2963.

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.

Werner EE, Gilliam JF (1984) The ontogenetic niche and species interactions in size-structured populations. Annu Rev Ecol Syst 15:393–425.

8 Appendices

The following includes some of the high level source code for the base runs discussed above.

8.1 Code: baserunChemostat

```
% θ
% Make a basic run of the chemostat model
% In:
%  mAdult is the adult sizes of copepods (can be left empty to simulate only
%         unicellular organisms) (default = [], ie only generalists).
%
% Out:
%  sim: Structure holding the results of the simulation
%
function sim = baserunChemostat(mAdult)

arguments
    mAdult double = []
end
%
% Set parameters:
%
p = setupGeneric(mAdult);
p = parametersChemostat(p);
p.tEnd = 2000;
p.d = 0.01;     % dilution rate [1/day]
%
% Set to "normal" HTL mortality if there are no copepods:
%
if isempty(mAdult)
    setHTL(0.1, 1/500^1.5, false, false);
else
    setHTL(0.1, 1, true, true);
end
%
% Simulate
%
tic
sim = simulateChemostat(p, 100);
toc
%
% Plot
%
plotSimulation(sim);
checkConservation(sim);
```

8.2 code: baserunWatercolumn

```matlab
%
% Run a watercolumn with only generalists
%
% In:
%  lat, lon - latitude and longitude
%
% Out:
%  As simulation structure
%
function sim = baserunWatercolumn(mAdult, lat, lon)

arguments
    mAdult double = []
    lat double = 60;
    lon double = -10;
end

% p = setupGeneralistsOnly(25);
p = setupGeneric(mAdult);
p = parametersWatercolumn(p);
p.tEnd = 1095;

%
% Set to "normal" HTL mortality if there are no copepods:
%
if isempty(mAdult)
    setHTL(0.1, 1/500^1.5, false, false);
else
    setHTL(0.1, 1, true, true);
end

sim = simulateWatercolumn(p, lat,lon);

plotSimulation(sim)

checkConservation(sim);
```

8.3 code: baserunGlobal

```
% Make a basic run of the global transport-matrix model.
%
% Tranport matrices must be downloaded from
http://kelvin.earth.ox.ac.uk/spk/Research/TMM/TransportMatrixConfigs/
% (choose MITgcm_2.8deg), and put into the location '../TMs'
%
% In:
%  With no arguments it runs the simple generalist model
%  With a parameter argument it runs the setup specified in the parameters.
% Out:
%  A simulation structure
%
function sim = baserunGlobal(p)
%
% Setup a basic run of the global model with only generalists
%
if (nargin==0)
    p = setupGeneralistsOnly(10, true); % Use 10 size groups and parallel execution
    p = parametersGlobal(p); % Use standard low-res model
    %p = parametersGlobal(10,2); % Use MITgcm_ECCO
    p.tEnd = 365;
end
%
% Simulate
%
if exist(strcat(p.pathInit,'.mat'), 'file')
    % Load decent initial conditions
    disp('Loading initial conditions from file');

    sim = loadGlobal(p);
    sim = simulateGlobal(p,sim);
else
    sim = simulateGlobal(p);%,sim); % Simulate
end
sim.B(sim.B<0)=0; % Get rid of negative biomasses
%disp('Calculating functions')
%sim = calcGlobalFunction(sim); % Calculate functions
%
% Plots:
%
disp('Plotting')
plotSimulation(sim)

checkConservation(sim);
%
% CPU-heavy plots:
%
% animateGlobal(sim);
```

8.4 code: higher trophic level mortality

```
%
% Sets the higher trophic level mortality. Already done by default in the
% setup function, so this function is only needed to override the defaults.
%
% HTL mortality is either a standard mortality that is fixed by mortHTL, or
% it is a "quadradratic" mortality that is proportional to the biomass. In
% the latter case morthHTL is the constant in front of the quadratic
% mortality, and not the mortality itself.
%
% In:
%  mortHTL - the HTL mortality (or the constant if bQuadratic=true)
%  mHTL - the size where the HTL starts (usually set to the maximum size
%         divided by 500^1.5)
%  bQuadraticHTL - boolean which determines whether to use a fixed or
%                  "quadratic" mortality
%  bDecliningHTL - boolean which determines whether the HTL mortality
%                  declines with size as mass^-0.25.
%
% Out:
%  Nothing; the function only affects the fortran library.
%
function setHTL(mortHTL, mHTL, bQuadraticHTL, bDecliningHTL)

arguments
    mortHTL double {mustBeNonnegative} = 0.2;
    mHTL double = 1/500^1.5; % Suits simulations with only generalists
    bQuadraticHTL logical = false;
    bDecliningHTL logical = false;
end

calllib(loadNUMmodelLibrary(), 'f_sethtl', ...
    double(mHTL), double(mortHTL), logical(bQuadraticHTL), logical(bDecliningHTL) );
```