Deltakennis

Modelling Carrying Capacity

Deltakennis

Modelling Carrying Capacity

T. Troost

Rapport

december 2009

Client	Waterdienst
Title	Deltakennis

Abstract

In this study, the carrying capacity of the Oosterschelde is calculated using an ecosystem model that includes grazers. The grazer module is developed and improved in various steps. Starting point is a DEB model for individual cockle growth which was scaled up to a population growth model and incorporated in a fully integrated ecosystem model of the Oosterschelde. The model was applied to simulate a 'generalized' grazer, as well as grazers of various species or size-classes. The results of the various model variants are compared and validated by nutrient concentrations and carrying capacity-related variables such as primary production and turnover times.

When incorporating grazers in the ecosystem model, the fit of nutrient concentrations with measurements improves considerably, especially in the northern and eastern compartments of the Oosterschelde. However, the fits with carrying capacity-related variables in these compartments remain poor. This is probably due to imperfections in both the model and the measurements.

Taking into account interspecific variation or size structure in the grazer model does not have much impact on the predicted primary production and turnover times. This indicates that the grazer model is robust with regard to these variables, and that the simpler unstructured model can thus be well used to predict these variables at the scale of individual compartments. For questions regarding specific species or size-classes, however, a structured model-variant is required.

In order to show how the model may contribute to study the effects of management measures, it was applied in a preliminary case-study on the effects of opening the Krammer sluices.

References

Ver	Author	Date	Remarks	Review	Approved by
	dr. ir. T.A. Troost			dr. F.J. Los	Ir. T. Schilperoort

Project number	1200314.001
Keywords	grazers, carrying capacity, Oosterschelde
Number of pages	113
Classification	Geen
Status	Definitief

Inhoud

Dutch summary

De Zeeuwse delta kent verschillende functies. Zo dient deze bijvoorbeeld als natuurreservaat en voorziet hij in percelen voor de aquacultuur van schelpdieren. Het systeem staat niet stil, maar is aan verschillende natuurlijke en antropogene veranderingen onderhevig. Zo is bijvoorbeeld de morfologie aan het veranderen door de aanleg van de deltawerken (zandhonger). Ook zijn er een aantal invasieve soorten in opkomst, zoals de japanse oester en Amerikaanse zwaardschede (*Ensis*). Een andere ontwikkeling is de inzet van zogenaamde MZI's (mosselzaad invang installaties). Ten slotte worden er ook plannen gemaakt om een deel van de dammen te verwijderen, en zo bijvoorbeeld het Volkerak Zoommeer weer zout te maken.

Met het oog op deze veranderingen en ontwikkelingen onstaan er vragen over de impact die deze kunnen hebben op de draagkracht van de systemen voor o.a. wilde en gecultiveerde schelpdierpopulaties. Geïntegreerde ecosysteem modellen kunnen helpen bij het beantwoorden van deze vragen, mits de grazer-gerelateerde processen voldoende realistisch zijn meegenomen. Modellering van grazers is echter lastig omdat er veel feedback processen bij betrokken zijn. Het doel van dit project is dan ook om tot een betere grazer-modellering te komen.

Het ontwikkelen van een bruikbaar grazermodel is aangepakt in verschillende stappen. Als startpunt is een model gebruikt voor individuele kokkelgroei gebaseerd op de zogenaamde DEB theorie. Kokkels zijn eenvoudiger te modelleren dan bijvoorbeeld mosselen of oesters, die dichte bedden vormen. Het kokkelmodel is gekalibreerd voor de Oosterschelde door het opstellen en parameteriseren van de relatie tussen voedsel aanbod en opname (functionele respons).

Vervolgens is het individuele groeimodel voor kokkels opgeschaald naar populatieniveau. Dit populatie model is gekoppeld aan een generiek ecosysteem model (GEM). Door het model te runnen in een simpele schematisatie (een homogene bak water) kon het modelgedrag goed worden bestudeerd en naar aanleiding daarvan zijn een aantal processen en parameters worden aangepast om de resultaten consistent te maken met het individuele groei model.

Daarna is het simpele geïntegreerde ecosysteem model uitgebreid naar een meer realistisch model van de Oosterschelde. Hiervoor was het nodig om een nieuwe toepassing van het hydrodynamisch model (FLOW) en het generiek ecosysteem model (GEM) te maken. Als dit model wordt gedraaid zonder grazers, leidt dit met name in ondiepe (graasgedomineerde) gebieden tot onrealistische resultaten, wat de noodzaak van goede grazermodellering bevestigt.

Door de parameterwaardes die specifiek zijn voor kokkels te vervangen is het model te gebruiken voor het simuleren van een generieke grazer. Model-resultaten worden gevalideerd aan de hand van nutrienten concentraties en draagkracht-gerelateerde variabelen zoals totale primaire productie en verblijfstijden van de algen in het systeem. De resultaten van het GEM met grazers zijn beduidend beter dan die van het GEM zonder graas.

Vervolgens is gekeken wat het effect is als de generieke grazer wordt opgedeeld in een aantal grootte-klasses. Hierbij is naar twee model varianten met toenemende

complexiteit gekeken. Het blijkt dat grootte-klasses niet veel effect hebben op het modelgedrag; zowel de resulterende nutrienten concentraties en draagkracht blijven min of meer gelijk voor beide bestudeerde varianten. Het simpele (niet-gestructureerde) model is dus robuust en kan prima worden gebruikt om het gedrag van het systeem te bestuderen. De opdeling in grootte klasses kan echter wel weer relevant worden in het geval van studies naar specifieke grootte klasses, zoals bijvoorbeeld het geval is bij MZI's.

Daarna is ook een soort gevoeligheids-analyse uitgevoerd door te bestuderen wat het effect is van het parameteriseren van de graasmodule voor drie verschillende soorten: kokkels, mosselen, en oesters. Ook is er een simulatie uitgevoerd waarin deze drie soorten tegelijk in het systeem aanwezig waren. De resultaten laten opnieuw zien dat het simpele model voor een generieke grazer redelijk robuust is. Wel kunnen de modellen voor de verschillende soorten van belang worden op het moment dat er vragen zijn die specifiek met een van deze soorten te maken heeft, of met interspecifieke competitie.

Hoewel de gesimuleerde nutrienten concentraties nog niet in alle gevallen kloppen met de metingen, is het model toch toegepast in een case-studie. Dit om te laten zien hoe het model kan worden ingezet om de effecten van bepaalde beheersmaatregelen te bestuderen. De case-studie in dit rapport gaat over de effecten van het vergroten van de capaciteit van de Phillipsdam. Aangezien de case-studie 'quick & dirty' is uitgevoerd, moeten deze resultaten met voorzichtigheid worden geïnterpreteerd. Voorlopige resultaten suggereren dat in het bestudeerde scenario de draagkracht van de Oosterschelde mogelijk iets groter zal worden, maar de verschillen zijn erg klein en vallen ruimschoots binnen de onzekerheidsmarges van het model.

1 Introduction

The Rhine-Meuse-Scheldt delta in the Southwest of the Netherlands accommodates various functions, including nature reservation and aquaculture. For example, in the Oosterschelde, mussels (*Mytilus edulis*) are cultured on some 3900 ha of farm area, and oysters (mainly *Crassostrea gigas*) on approximately 1550 ha (Figure 1.1).

Between 1950 and 1997, most of the Delta area has been dammed off by a series of constructions to protect a large area of land around the delta from the sea. This resulted in various more or less separated systems each with its own distinct characteristics. The hydromorphology and ecology of these systems are not static or stable, but they are continuously changing and adapting in response to natural, anthropological, and climatological changes. For example, the morphology is changing and tidal flats are eroding due to the dams. Also, various invasive species have appeared whose densities are ever increasing (e.g. pacific oyster and American razor clam (*Ensis*)). Furthermore, plans exist to take out some of the dams and make the Volkerak Zoommeer saline again. Another recent development is the use of specific devices to catch mussel seed (MZI's).

In view of these changes and of possible future developments in the region, questions arise with regard to the impact on natural and cultured shellfish populations. Integrated ecosystem models can help to answer these questions, provided that hydrodynamical, chemical and ecological processes are incorporated in a sufficiently realistic way. Whereas models for hydrodynamical, chemical and some ecological processes such as primary production have been well-developed over the past few years, improvements are still required on the modelling of secondary production and the effects of grazing on phytoplankton.

Therefore, the aim of this is study is to improve the modeling of these grazer dynamics. This is done by including a grazer module into an ecosystem model of the Oosterschelde. The model results are validated against observed chlorophyll and nutrient concentrations, as well as carrying-capacity related parameters such as primary production and turnover time.

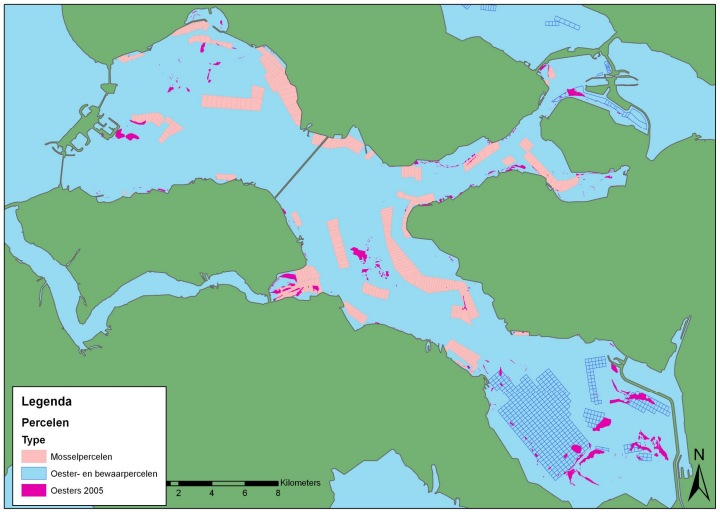


Figure 1.1. Musselfarms, oysterfarms, and wild oyster populations in the Oosterschelde in 2005.

1.1 Starting point

This report is an elaboration of the report written in 2008. Steps that were taken in the previous year were:

1 Modeling of individual cockle growth
Starting point was an individual growth model for cockles based on DEB theory. The model was calibrated for cockles in the Oosterschelde by parameterizing the functional response. This calibration was carried out by Wageningen IMARES (Wijsman et al, 2009).

2 Simulating population density time series
The individual growth model was scaled up to a population model, and was used to simulate a timeseries of cockle population densities from measured chlorophyll concentrations. The model was used to calibrate the population reference lengths and mortality rates.

3 Incorporating cockles in a basic integrated ecosystem model
By incorporating the cockle population model in a basic and homogeneous integrated ecosystem model, the effects of including food quality and feedback of grazers on algal concentrations and growth were studied.

4 Incorporating cockles in a Oosterschelde integrated ecosystem model
The cockle model was incorporated in a more realistic integrated ecosystem model representing the Oosterschelde system, for which a new application of the hydrodynamic (FLOW) model and ecosystem (GEM) model were set up.

1.2 This year's improvements

In June 2009 a workshop was organized in which a group of experts discussed which should be the main issues on which the research should focus. The main conclusion from the workshop was that the project should focus first on the questions related to carrying capacity, and not (yet) on those related to interspecific competition. Only after the model is capable of capturing the system behaviour well enough to calculate its carrying capacity, should it be extended to include interspecific differences and competition.

A second conclusion was that size structure (intraspecific differences) is considered to be of more importance with respect to carrying capacity than interspecific differences.

<u>The suggested strategy was to:</u>

1. Improve technical issues
    - Include threshold for filtration on intertidal mudflats
    - Check the FLOW model in the Northern compartment
2. Include one generalized shellfish species
    - Define an "average" parameter set for a generalized shellfish species
    - Initialize the generalized shellfish by the actual (known) shellfish distribution

3. Improve feedback processes
    - Incorporate the processes of filtering, selection, and pseudofaeces production
    - Check for data on size distribution of phytoplankton in the Oosterschelde.
    - Include additional algae (size-)groups/species in BLOOM
    - Include an assimilation preference for smaller sized algae in the grazer module
4. Include size structure in the shellfish population
5. Incorporate interspecific variation

Most of the grazer-related suggestions above have been dealt with. The phytoplankton-related suggestions have not been looked into yet.
The improvements that have been made are as follows:

1) A threshold for filtration on intertidal mudflats was included (section 3.2.1. and 3.2.2).
   The FLOW simulation was improved (section 5.3).
2) The model was parameterized for a generalized grazer species (section 3.4.1 and section 5.5).
   Initial densities were based on the actual (known) shellfish densities (section 5.6.1).
3) The feedback fluxes were improved by removing an error from the code (section 4.2).
   Psuedofaeces production was not yet included, but steps have been taken to facilitate its future implementation in the model (section 3.3).
4) A simulation was done with grazers consisting of 3 size classes (section 5.7).
   Two size-structured model variants were compared (section 5.7).
5) The model was parameterized for 3 different species, which were simulated both separately and simultaneously (section 5.8).

2 Approach

A range of shellfish and/or zooplankton-modules already exists. These include:

- Depletion module (MaBeNe)
- CONSBL (WL | Delft Hydraulics) **
- ERSEM-zooplankton module (NIOZ et al.) *
- ZOODYN (WL | Delft Hydraulics) *
- STORG (WL | Delft Hydraulics) *
- COCO&EMMY (IMARES)
- DEB models (VU)
- ShellSIM (Plymouth Marine Laboratory, PML) *
- Size structured model (UvA) *
- Individual based models (WUR)

(* incorporated in a research version of DELWAQ)
(** incorporated in the official DELWAQ version)

Each of these modules has its issues (see Workshop Report 2008) and so far, none has proven to work well for modelling a system's carrying capacity. A next step into grazer modeling could thus include the use and adaptation of an existing grazer module or the development of a new one. It was decided to discuss this choice and the further approach with a small group of experts. The discussions consisted of:

- Workshop Draagkrachtmodellering (See Workshop report 2008)
- Consult with Peter Herman (NIOZ) (Workshop report 2008, Appendix C)
- Questionnaire and conversation with Jeroen Wijsman (Wageningen IMARES) (Workshop report 2008, Appendix D)

2.1 Bottom-up approach

Based on the discussion in the workshop of 2008, it was decided that a stepwise and bottom-up approach was to be taken in the modelling of grazers. This means that the initial grazer module should be simple, and (only if necessary) it should be step by step increased in complexity. Also, the focus should lie on inter-specific variation, less on intra-specific variation.

As a first step, the module could be used for a habitat suitability analysis. Such an analysis may be more workable than a dynamic simulation, as it consists of a static description of the system. Also, changes in habitat suitability are of major interest, so predictions on this issue are very welcome. Moreover, to implement a grazer module into a integrated ecosystem model (GEM), a habitat suitability analysis can be used for determining the initial grazer distribution.

Furthermore, it was agreed that the Oosterschelde was the most appropriate system for testing the grazer module, as many shellfish data are available on the Oosterschelde, and shellfish play an important role in the system's dynamics. Also, previous studies have indicated that shellfish production in Oosterschelde is close to its carrying capacity. Any feedback effects from shellfish on primary production should therefore be easily detectable. In other words: if they are not detactable in the Oosterschelde, these effects are unlikely to be important in other systems. A case study may then analyse

the impact of the opening of the Phillipsdam on the secondary production in the Oosterschelde.

2.2 DEB model

In the above mentioned workshop, it was agreed on that the so-called “STORG-module” was the most appropriate grazer module to start from (Wijsman, 2004). STORG has a general lay-out that can be used for suspension and deposit feeders, and is not size-structured nor individually based. Moreover, it has already been incorporated in DELWAQ. Initially, it was decided not to use a DEB model, because of the relatively complex DEB structure and because the creation of yet another grazer module was not desirable in general. Yet, after some discussion with Jeroen Wijsman, it was decided to convert the STORG module into a DEB model after all. This choice was based on the various advantages of DEB over other models:

- Generality

DEB models are based on physiological rules. This makes the structure relatively complex and less transparent than the more commonly used ‘scope of growth’ models such as ERSEM-organisms and COCO and EMMY models. However, owing to its physiological base, a DEB model can be used for several species, and parameter values are not system-specific. Furthermore, DEB models can accommodate a variety of complexities ranging from individual based models with a high level of physiological detail to simplified population models. These simplified population models may be as simple as any scope for growth model.

- Parameter value availability

Consistent sets of parameter values for DEB parameters already exist for various shellfish species (including *Mytilus edulis, Cerastoderma edule,* and *Crassostrea gigas*).

- Connection with other institutes

DEB models for shellfish are currently being developed and used by various institutes for applied science (NIOZ, IFREMER, Wageningen IMARES). Using a similar model structure facilitates cooperation and information exchange, now and in future projects.

2.3 Cooperation with IMARES

During the discussions with Jeroen Wijsman, it became apparent that a cooperation with IMARES in developing a grazer module might have several advantages. The contribution of IMARES could lie in the sharing of their model code and expert knowledge on shellfish physiology, and in providing data for validation and calibration, whereas the contribution of Deltares could consist of scaling-up of the individual based DEB model into a simpler population model, and the implementation into a GEM.

Cooperation with Wageningen IMARES consists of the following steps:

- conformity on DEB-code
  (The model code is compared, exchanged and discussed until conformity consists about its exact formulation. This step includes a mass-balance check.)

- calibration of the individual based model
  (The individual growth model is fitted to data on individual growth curves by calibrating the functional response function. This step is performed by Wageningen IMARES.)

- scaling-up to a population model
  (The model is scaled up to a population model and fitted to timeseries data on population densities. This step is performed by Deltares, but the density data is from Wageningen IMARES).

- jointly writing a scientific paper

- Over the course of 2009 a number of experiments were planned to determine some input parameters for the modelling more precisely. These experiments are also carried out in cooperation with IMARES. These parameters will be incorporated in the model at a later stage.

2.4 From cockles to a generalized grazer species

Cockles are more suitable modelling organisms than are mussels or oysters. This is because they do not form shellfish-beds as dense as mussels or oysters do. Shellfish-beds have complex consequences for food availability on a small scale due to their local effect on water flow, and on a large scale due to the formation of algal refuges (pers. comm. P. Herman, see Workshop report-Appendix C). Also, cockles fully complete their life-cycle in the Oosterschelde including the natural settlement of larvae, unlike mussels and oysters that are on a large scale sown and harvested. Last but not least, many data on cockle growth and distribution are available. For these reasons, the project was started by modeling cockles.

At a later stage, a ‘generalized’ grazer was introduced in the model. This generalized grazer is more representative for the various bivalves that are present in the Oosterschelde than cockles.

As a next step, the generalized grazer was split up into three size classes. Two model size-structured model variants were compared.

Finally, the generalized grazer was substituted by separate mussels, oysters and cockles. Also, these three species were simulated simultaneously.

3 The grazer module

3.1 DEB model for individual growth

3.1.1 DEB principles

The Dynamic Energy Budget theory is a modelling framework based on first principles and simple physiological-based rules that describe the uptake and use of energy and nutrients and the consequences for physiological organization throughout an organism's life cycle (Kooijman 2000).

DEB models are very general, and can thus be used for basically all species and lifestages. Furthermore, DEB models can accommodate a variety of complexities ranging from individual based models with a high level of physiological detail to simplified population models.

The aspect that makes DEB framework unique and separates it from so-called "net production" models, are its storage or reserve dynamics. The reserves play an important and central role in an organism's metabolism, and they are incorporated such that the organism is not directly dependent on its environment. As a result of the reserve dynamics, it can for example survive the periods in-between meals. The common DEB structure is given in Figure 3.1.

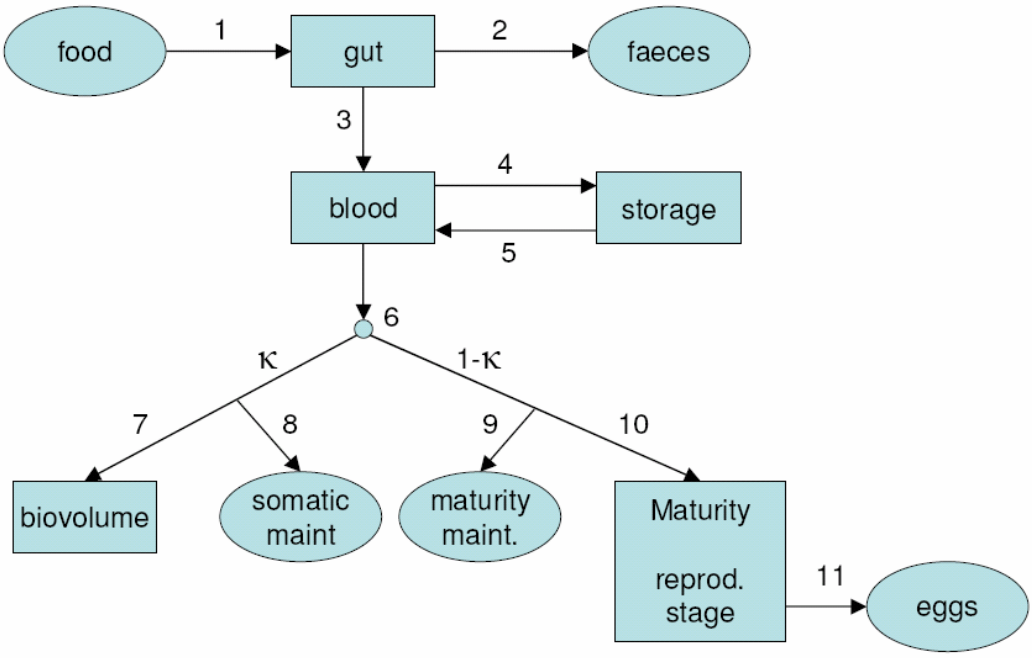


Figure 3.1. General structure of a DEB model with (1) ingestion, (2) defeacation, (3) assimilation, (4 and 5) storage or reserve dynamics, (6) utilization , (7) growth, (8) maintenance, (9) maturing, (10) reproduction, and (11) spawning.

In addition to its generality, the DEB framework is also flexible and the models can be extended to include species-specific characteristics that are necessary for a certain

application. In case of modeling shellfish, certain adjustments are made to include filterfeeding, spawning, and psuedofaeces production (section 3.2).

3.1.2 General DEB equations

The general DEB equations described below apply to the growth of an individual organism that does not change in shape during its life. The equations are based on first principles and physiological processes and their derivation is described in detail in (Kooijman 2000).

An individual organism state is represented by three state variables: structural volume (V, cm^3), reserves (E, Joule) and reproductive buffer (R, Joule). When the shape of a growing individual remains the same, its surface-to-volume ratio changes. This has an effect on the ratio between ingestion (surface-area dependent) and maintenance (volume-specific), so that growth will slow down when an organism becomes larger.

Assimilation

Organisms take up food from their environment. The energy ingestion rate (p_X, J d^{-1}) is proportional to the maximum surface-area-specific energy ingestion rate ({p_{Xm}}, J d^{-1} cm^{-2}), the scaled functional response (f), and the surface area of the organisms ($V^{2/3}$, cm^2). M_v is the so-called shape-correction function, which is 1 in case of isomorphic organisms.

$$p_X = \{p_{Xm}\} \cdot f \cdot V^{2/3} \cdot k_T \cdot M_v$$

Due to their limited capacity to assimilate ingested particles, only a fraction of the ingested food is assimilated, the rest is lost and released as faeces. The model assumes that the assimilation efficiency of food is independent of the feeding rate and the assimilation rate (p_A, J d^{-1}), which is calculated by

$$p_A = p_X . a_e$$

Assimilated energy is incorporated into a reserve pool from which it is used for maintenance, growth, development and reproduction following the so-called κ-allocation rule. A fixed proportion (κ) of energy from the reserves is allocated to somatic maintenance and growth and the remaining fraction (1-κ) is spent on maturity maintenance, development and reproduction.
The dynamics of the reserves are calculated as the balance between the assimilation and the mobilization rate (p_C, J d^{-1}), and the dynamics of the structural volume (growth) are based on the κ-fraction of the mobilization flux and somatic maintenance:

$$\frac{dE}{dt} = p_A - p_C$$

$$\frac{dV}{dt} = \frac{\kappa p_C - [p_M]V}{[E_G]}$$

$$p_C = \left(\frac{[E]}{\kappa[E]+[E_G]} \right) \cdot \left(\frac{a_e\{P_{Xm}\}[E_G]}{[E_m]} \cdot V^{2/3} + [p_M] \cdot V \right)$$

where [E] corresponds to the energy density of the organism (J cm^{-3}), $[E_G]$ is the volume specific costs for growth (J cm^{-3}) and $[E_m]$ is the maximum energy density of the reserve compartment. The parameter $[p_M]$ is the volumetric cost of maintenance (J cm^{-3} d^{-1}). The energy flow required for maintenance (p_M, J d^{-1}) is

$$p_M = [p_M] \cdot V$$

When the energy required for maintenance (p_M) is higher than the energy available for growth and maintenance ($\kappa\, p_C$) the energy for maintenance are paid by structural volume and the model organism shrinks.

It is assumed that both the structural body and the reserves have a constant chemical composition (assumption of strong homeostasis). At constant food density, the energy reserve density (E/V) becomes constant (weak homeostasis assumption) (Kooijman, 2000).

Maturity and reproduction

As mentioned above, a fixed proportion (1-κ) of the utilized energy (p_C) goes to maturation, maturity maintenance, and reproduction. Juveniles use the available energy for developing reproductive organs and regulation systems. Adults, which do not have to invest in development anymore, use the energy for reproduction and maintenance. Also, both adults and juveniles have to pay maturity maintenance costs. The transition of juvenile to adult is assumed to occur at a fixed size (Vp).

For juveniles, the maturation development costs are given by:

$$p_{dev} = \left(\frac{1-\kappa}{\kappa} \right) \cdot [E_G] \cdot \frac{dV}{dt}$$

The maturity maintenance costs for juveniles are given by:

$$p_J = \left(\frac{1-\kappa}{\kappa} \right) \cdot [P_M] \cdot V$$

For adults, the maturity maintenance costs are given by:

$$p_{mat} = \left(\frac{1-\kappa}{\kappa} \right) \cdot [P_M] \cdot V_P$$

And the costs for reproduction are given by:

$$\mathrm{P}_{rep} = (1-\kappa)\, p_C - P_{dev} - P_{mat}$$

The length of the organism (L, cm) can be calculated from the structural volume using the shape coefficient (δ_M) as follows:

$$L = \frac{V^{1/3}}{\delta_M}$$

Ash-free dry weight AFDW (g), excluding inorganic parts such as shells, can be obtained by summing-up the state variables V, E and R.

$$AFDW = \Psi_{AFDW\_WW}\rho V + \frac{E+R}{\Psi_{E\_AFDW}}$$

where Ψ_{AFDW_WW} is the conversion factor from wet weight to AFDW (g AFDW g Wet Weight^{-1}), ρ is the density of the flesh (g cm^{-3}), and Ψ_{E_AFDW} is the energy content of the reserves in ash-free dry mass (J g^{-1}).

Finally, it is assumed that all physiological rates are affected by temperature in the same way. This temperature effect is based on an Arrhenius type relation, which describes the rates at ambient temperature, $\dot{k}(T)$, as follows:

$$\dot{k}(T) = k_1 e^{\left(\frac{T_A}{T_1}-\frac{T_A}{T}\right)} \frac{1+e^{\left(\frac{T_{AL}}{T_1}-\frac{T_{AL}}{T_L}\right)}+e^{\left(\frac{T_{AH}}{T_H}-\frac{T_{AH}}{T_1}\right)}}{1+e^{\left(\frac{T_{AL}}{T}-\frac{T_{AL}}{T_L}\right)}+e^{\left(\frac{T_{AH}}{T_H}-\frac{T_{AH}}{T}\right)}}$$

where T is the absolute temperature (K), T_{AL} and T_{AH} are the Arrhenius temperatures (K) for the rate of decrease at respectively the lower (T_L) and upper (T_H) boundaries. T_1 is the reference temperature (293 K), T_A is the Arrhenius temperature, and k_1 is the rate at the reference temperature.

3.1.3 Shellfish-specific DEB equations

The DEB models for cockle and mussel are based on the standard DEB model which is described above. Some additions are made to the standard DEB model to incorporate shellfish specific aspects. These additions are not new but have been made before in other shellfish (DEB) modeling studies (Bacher & Gangnery 2006, Pouvreau *et al.* 2006, Rosland et al. 2009).
Specific for shellfish is that they filter food from the water column. Therefore, the relation between food uptake and food density is described by a scaled hyperbolic functional response f proposed by Kooijman (2006):

$$f = \frac{FOOD}{K'(Y)+FOOD} \quad \text{in which} \quad K'(Y) = X_K\left(1+\frac{Y}{Y_K}\right)$$

where FOOD is the available density of food, and Y is the concentration of inorganic matter which is calculated as the *total* particulate matter minus the particulate *organic*

matter (Y=TPM-POM). X_K and Y_K are the half saturation constants for food and for inorganic particles, respectively. The value of f varies from 0 (no food uptake) to 1 (*ad libitum* food conditions). When the available amount of food equals K'(Y), the food uptake rate is half the maximum uptake rate. The response curve corresponds to the Type II response curve of Holling (1959). When the amount of inorganic particles in the water column increases, the f value decreases, simulating the negative influence of these particles in the filtration capacity of the bivalve.
In the formulation of the functional response, the available density of food is defined as a single variable (FOOD). In reality, it is a function of the concentration of the various food items, the quality of the food and the food acquisition rate. The food acquisition rate, in turn, depends on factors such as filtration rate, selection efficiency, inundation, etc. In many shellfish DEB models the available amount of food is simplified by using the chlorophyll-a concentration as a proxy. In the present model, FOOD is described by a function of both the chlorophyll-a concentration (µg l^{-1}) and the detritus concentration (mg l^{-1}). Detritus (defined in this study as particulate organic matter, POM) consists of a range of components of which a fraction can be used by the bivalve as food. By incorporating detritus in the formulation, its importance as a food source for shellfish can be investigated. The unit of FOOD is in Chl-a equivalents (µg l^{-1}) and it is computed as:

$$FOOD = Chla + \alpha_{DET} DET \quad ,$$

where *Chla* and *DET* are the measured Chlorophyll *a* and detritus (POM) concentrations, and α_{Det} is the relative contribution of detritus to food in Chlorophyll-a equivalents (µg Chla mg DET^{-1}).
Also specific for shellfish is their spawning behaviour. Spawning events occur when enough energy is allocated into the gonads (Gonado-Somatic Index, GSI > $Thresh_{GSI}$) and when the water temperature is above a threshold value ($Thresh_{Temp}$). The gonads are released from the buffer at a rate of 2% per day until the temperature drops below the threshold value or the GSI<0.0001. A 2% per day gonad-release corresponds to a period of about one month during which half of the gonads are released.
Furthermore, the general DEB model does not differentiate between faeces and pseudofaeces production, as pseudofaeces production is a shellfish-specific process. Pseudofaeces production does not affect growth in a different way than faeces production, but the two products have different characteristics with respect to sedimentation and mineralization. Therefore, as long as interactions with the environment are not taken into account, the difference will not affect the model results. However, when environmental feedback is taken into account, it may play a more important role. At the moment, however, little is known about the exact differences between faeces and pseudofaeces with regard to their fate. Therefore, more research would be needed before the two substances can be incorporated separately in the model.

3.2 From individual based model to population model

To scale DEB models up to population level, several approaches can be adopted. In this report, two alternatives are described and their results are compared: modeling size-classes of isomorphs versus modeling a population consisting of V1-morphs.

3.2.1 Isomorphs

The DEB model described in the section above apply to the growth and reproduction of an individual organism that does not change in shape during its life. A growing organism that does not change in shape during its life is called an **iso-morph** (Kooijman 2000). When the shape of a growing individual remains the same, its surface-to-volume ratio changes. This has an effect on the ratio between ingestion (surface-area dependent) and maintenance (volume-specific), so that growth will slow down when an organism becomes larger.

To scale DEB models up to population level, several approaches can be adopted. An obvious approach is to model many individuals simultaneously. In case of iso-morphs, however, this approach leads to a complex model comprising many differently-sized individuals, which may give computational problems, may be difficult to initialize, and whose results may be difficult to analyze or understand.

An easier solution is to model various classes of similarly sized organisms that are equal in all aspects (length, structural volume and other state variables such as energy reserves and reproductive material). The organisms within each class thus follow the same growth trajectory. To go from individual iso-morphs to such size-classes of iso-morphs, some additional information is required on initial values, larval settlement and mortality, which issues are discussed below.

- Density

An additional state variable is needed to keep track of the total number of individuals in the population or size class. This variable is affected only by mortality, since recruitment is included as a new size class, and thus does not lead to an increase in the number of individuals in the existing size classes. The number of individuals can be used to calculate denisity and total biomass using the surface area or the biomass per individual, respectively. The other state variables remain similar to those in the individual model: structural volume (cm^3), the energy buffer (J), and reproduction buffer (J).

- Initialization

To describe each class of organisms, their initial size is required, as well as their initial density.

- larval settlement

Recruitment and settlement of larvae does not have to be included in the model formulations, but a new class of young/small individuals has to be included periodically in the simulation (e.g. once a year).

- mortality

Another important population process is mortality. The mortality rate constant was calculated on basis of the following formula:

$$n_1=n_0*exp(-m*t),$$

where n_1 is the number of individuals at time t_1; n_0 the number of individuals at time t_0; t the time period between time t_0 and t_1; and m the mortality rate constant.

Note that in addition to this 'background' mortality during summer, additional mortality takes place in winter due to starvation. This additional mortality is incorporated in the model, and occurs when maintenance is larger than the utilization rate ($p_c<p_m$) , or when maturity maintenance is larger than the flow to maturity ($p_j+p_r > (1-k)p_c$). In these cases, the growth rate and/or reproduction rate become negative. Though in some models these rates are constrained not to become negative, in our model they can. The corresponding interpretation is that in case the reserves are not sufficient, the organisms becomes smaller.

- feeding at intertidal mudflats

Obviously, the shellfish cannot feed during periods at which the mud flats fall dry. Therefore, the shellfish' functional response is set to zero when the depth of the grid cell in which it is located becomes smaller than 0.1m.

- equations and model code for iso-morphs

The model equations for individual iso-morphs are described in section 3.1. The full model code for populations of iso-morphs can be found in Appendix B.1.

3.2.2 V1-morphs

The model for iso-morphs described above requires four state variables per size-class of individuals (total number of individuals, structural volume, energy density, and reproduction density). The calculation may therefore be rather computational-intensive. Also, all these state variables have to be initialized, which can be difficult if detailed information on the various size-classes is not available.

When one of the above mentioned problems cannot be overcome, or if detailed output on the various size-classes of shellfish is not required (e.g. if the model is used to study overall ecosystem performance), an alternative solution is available to scale up from individual organisms to populations.

This alternative approach, leading to the simplest model as possible, is to approximate the population of differently-sized and growing individuals by a population of equally sized organisms that do not change in size. Organisms that have a constant surface-to-volume ratio are called **V1-morphs** (Kooijman 2000). This assumption can be made without problems when the organisms are small (such as unicellulars), because the change in size during their live is small too. The assumption may thus be more correct for small shellfish species (cockles) than for larger ones (oysters).

Modeling V1-morphs instead of isomorphs greatly simplifies the model structure, as one of the state variables (length, and thus the structural biomass of an individual) becomes a constant. As a result, a whole population can be simulated by three state variables (number of individuals or the total structural volume of the population, the energy density and reproductional density). The total biomass in this population can change due to mortality and growth, but the individuals of which it is constitutes do not change in size. Note that the population can still be split up into various size-classes.

<u>From iso-morphs to V1-morphs</u>
The iso-morph to V1-morph approximation can be included in the formulation by multiplying all surface-dependent rates by the so-called shape-correction function,

$M_v = V^{(1/3)} / (\delta_m * l_{ref})$, where δ_m is the shape-correction coefficient and l_{ref} is the reference length. M_v is already included in the equations for isomorphs in section 3.2.1, but does not affect the outcomes for isomporphs since for them M_v is equal to 1.

In a way, the reference length characterizes the population size composition. While it is easily incorporated in the model formulation, it is more difficult to determine its value, which will be done in this study through calibration (Section 4.1).

Other adjustments

Furthermore, like in the model for iso-morphic populations described above (section 3.2.1), additional information on reproduction and mortality are required, which are discussed below.

- Density

As a consequence of the assumption of a constant size, the individual organisms do not change in length, and their structural volume stays the same as well. Therefore, it is not needed to simulated the structural volume per individual. Instead, the state variables are expressed for the whole population (or size class), and are expressed per m^2: the total structural volume of the population (cm^3/m^2), the energy density (J/m^2), and reproduction buffer (J/m^2). Due to the change in units of the state variables, all related fluxes also change in their units (see Table 3.1).

- larval settlement

When simulating the population by one ‘super-organism’, the fate of the produced eggs and larvae has to be included explicitly. Obviously, when simulating a period of one year only, the fate of these reproductional products is not very relevant. However, when simulating a population over several years, larvae settling should be considered.

Since the recruitment rate seems to be related to environmental factors and post-settlement predation rather than to the standing stock or spawning biomass (Beukema & Dekker 2005), larvae settling was incorporated in the model to be independent from density and spawning biomass.

Larval settlement has two effects on population growth. First, it enables shellfish growth at all (suitable) locations, where shellfish are not yet present. This effect can be modeled by adding a constant small amount to the shellfish biomass. At locations where shellfish are already present, the small increase of biomass has very little effect, as the volume-to-surface relation of all shellfish is assumed to be constant. Therefore, the constant but small increase of biomass can be implemented more easily but equally effectively by setting a minimum to the shellfish density.

The second effect of larvae settling on population growth is that it leads to a decrease of the average body size in the population. This leads to a more favorable volume-to-surface ratio, thus facilitating population growth. In the population model, in which body sizes are assumed to be constant, this effect can be implemented by decreasing the reference length. In the calibration of the reference length (Section 4.1), the reduction of reference length due to recruitment is automatically taken into account.

As a further refinement, a seasonal pattern of larvae settling could be included in the model. This could be done by making the settling rate (i.e. the small amount of biomass

that is constantly added to the shellfish biomass) dependent on time or season. Also, the reference length could be varied throughout time by means of a forcing function. However, such specific forcing functions complicate the model without providing any explanations, and will not be implemented unless this is absolutely required to fit the model results to the observations.

- mortality

For mortality of V1-morphs, the same rules apply as for the iso-morphs (see Section 3.2.1). This means that for the background mortality, a fixed fraction is subtracted from the population each time-step. In addition, starvation may lead to a decrease of structural volume. The corresponding interpretation (for isomorphs) is that in case the reserves are not sufficient, the organism becomes smaller. In the V1-model in which all organisms are of a constant size, the interpretation is that a fraction of the organisms dies, leaving relatively more food for the survivors.

- feeding at mudflats

Like in the population model for iso-morphs, the shellfish cannot feed during periods at which the mud flats fall dry. Therefore, the shellfish' functional response is set to zero when the depth of the grid cell in which it is located becomes smaller than 0.1m.

- model equations and code for V1-Morphs

The V1-model for individual growth is very similar to that of isomorphic individuals, see section 3.1. As explained above, it only differs in the shape-correction constant M_v, which is no longer equal to 1.

For populations, the model is adjusted as conceptually described above. The precise model equations that were adjusted for the population model are given below. The full model code of the V1-population model as it is incorporated in the GEM is provided in Appendix B.2.

The state of the V1-population is described by its state variables 'total structural body volume' (V) in units of $cm^3\ m^{-2}$ and 'energy reserves' (E) expressed in units of $J\ m^{-2}$. The reserves can also be quantified as energy density (E/V) in units of $J\ cm^{-3}$. Note that the units and interpretation of these state variables slightly differ from those in the isomorphic population model, where they still correspond to the individual state. The volume of an individual organism (V_d) in the V1-population, can be calculated from the reference length and the shape coefficient (δ_m) as follows:

$$V_d = (\delta_m L_{ref})^3$$

The density of individuals in the population (N, $\#\ m^{-2}$) can be calculated from the structural volume of the population per m^2 (V) and the volume of an individual organism (V_d):

$$N = \frac{V}{V_d}$$

Juveniles use the available energy for developing reproductive organs and regulation systems. Adults, which do not have to invest in development anymore, use the energy for reproduction and maintenance. In the isomorphic model, the transition of juvenile to

adult occurs at fixed size (V_p). In our population model, all individuals have an equal size (V_d), which is either larger or smaller than V_p. However, this size may be considered as a mean size, and to take into account some variation around this size, a fraction $V_p/(V_p+V_d)$ of the population is assumed to be smaller than V_p, while the rest is assumed to be larger than V_p. The maturation development and maintenance costs are thus assumed to be proportional to this fraction as well.

The maturation development costs are given by:

$$p_{dev} = \left(\frac{1-\kappa}{\kappa}\right)\cdot[E_G]\cdot\left(\frac{V_P}{V_P+V_d}\right)\cdot\frac{dV}{dt}$$

The maturity maintenance costs are given by:

$$p_{mat} = \left(\frac{1-\kappa}{\kappa}\right)\cdot[p_M]\cdot\left\{\left(\frac{V_p}{V_p+V_d}\right)+\left(1-\left(\frac{V_p}{V_p+V_d}\right)\right)\cdot\frac{V_p}{V_d}\right\}\cdot V$$

And the costs for reproduction are given by:

$$P_{rep} = (1-\kappa)\,p_C - P_{dev} - P_{mat}$$

Table 3.1 Description and units of state variables and fluxes, and the difference in their units between the isomorphic and V1-morphic models.

	description	**Unit (isomorph)**	**Unit (V1-morph)**
State variables		Variables apply to individual	Variables apply to population
V	Structural volume	cm^3	$cm^3\ m^{-2}$
E	Energy buffer	J	$J\ m^{-2}$
R	Reproductive buffer	J	$J\ m^{-2}$
Fluxes			
p_X	energy ingestion rate	$J\ d^{-1}$	$J\ m^{-2}\ d^{-1}$
p_A	assimilation rate	$J\ d^{-1}$	$J\ m^{-2}\ d^{-1}$
p_C	Catabolic rate	$J\ d^{-1}$	$J\ m^{-2}\ d^{-1}$
p_M	Somatic maintenance rate	$J\ d^{-1}$	$J\ m^{-2}\ d^{-1}$
p_J	Maturity maintenance rate juveniles	$J\ d^{-1}$	- (no difference between juvenile and adult)
p_{mat}	Maturity maintenance rate adults	$J\ d^{-1}$	$J\ m^{-2}\ d^{-1}$
p_{rep}	Reproduction rate	$J\ d^{-1}$	$J\ m^{-2}\ d^{-1}$
p_{dev}	Development rate	$J\ d^{-1}$	$J\ m^{-2}\ d^{-1}$

3.3 Pseudofaeces production

So far, pseudofaeces production was not yet included in the model, since too few data are available on which to base the required parameter values. Therefore, measurements are being done at NIOO to parameterize the pseudofaeces module.

Furthermore, a module for pseudofaeces production has already been developed (Wolfshaar, van de, K. 2007), which can be used in the grazer model. Implementation of this module requires some adjustments in the model code.

For the time being, measured silt concentrations are used to force the silt concentrations in the model. In these measured concentrations, the filtering effect of bivalves is already included.

3.4 Parameter values

3.4.1 Physiological parameter values

Values for physiological parameters are obtained from Van der Veer et al. (2006). Their set includes parameter values for various bivalve species at 293K (20 ° C), determined by a combination of direct estimates based on field and laboratory data and results of an estimation protocol for missing parameters (Table 3.2). Parameter values for mortality and reference length were discussed in section 3.2. Parameters for the functional response were based on the calibration performed specifically for this study (X_k = 2.0, α_{det} = 0, Wijsman et al, 2009), while inorganic matter was assumed not to play an important role ($Y_k = \infty$).

The advantage of DEB-models is that the same parameter values can be used for both the isomorphs and V1-morphs. For the generalized grazer, it was chosen to use the parameter values of mussels, as these lie nicely in-between the values of oysters and cockles.

Table 3.2. Measured and estimated parameter values for various bivalve species at 293K (20 ° C), after Van der Veer et al. (2006).

Symbol	Dimension	Interpretation	*M. balthica*	*M. arenaria*	*C. edule*	*M. edulis*	*C. gigas*
T_A	K	Arrhenius temperature	5800	5800	5800	5800	5800
$\{\dot{J}_{Xm}\}$	J cm^{-2} d^{-1}	Maximum surface area-specific ingestion rate	43.9	177.3	91.5	196.8	560.0
ρ	—	Losses due to digestion	0.25	0.25	0.25	0.25	0.25
$\{\dot{p}_{Am}\}$	J cm^{-2} d^{-1}	Maximum surface area-specific assimilation rate	32.9	133.0	68.6	147.6	420.0
$[\dot{p}_M]$	J cm^{-3} d^{-1}	Volume-specific maintenance costs	24	24	24	24	24
$[E_m]$	J cm^{-3}	Maximum storage density	2085	2180	2115	2190	2295
$[E_G]$	J cm^{-3}	Volume-specific costs of growth	1900	1900	1900	1900	1900
$[E_V]$	J cm^{-3}	Volume-specific structural energy content	1350	1350	1350	1350	1350
κ	—	Fraction of utilized energy spent on maintenance plus growth	0.80	0.75	0.80	0.70	0.45
δ_m	—	Shape coefficient	0.365	0.277	0.381	0.287	0.175
$\dot{v}$	cm d^{-1}	Energy conductance	0.016	0.061	0.032	0.067	0.183
$\dot{k}_M$	d^{-1}	Maintenance rate constant	0.013	0.013	0.013	0.013	0.013
g	—	Investment ratio	1.139	1.162	1.123	1.239	1.840
$\dot{r}_B$	y^{-1}	Von Bertalanffy growth rate	1.097	0.730	1.150	0.848	0.730

3.4.2 Natural mortality and harvesting rates

For cockles, Kamermans reported a natural mortality percentage of 28% (and thus a surviving percentage of 72%) of the cockles of one year and older between May 1st and September 1st, which period contains 123 days. Using the previously mentioned formulation (section 3.2.1), the resulting mortality rate for cockles thus comes down to 0.00267 gC/gC per day. In 1998, there was no fishery on cockles.

For oysters and mussels, and for the three size classes of the size-structured grazers, natural and fishery mortality rates were adopted from the Keyzones project (Blauw et al, 2007). Mussels are harvested rather intensively, while oysters are harvested much less. It was assumed that bivalves in the smallest size class (juveniles) are not being harvested at all, but that their natural mortality is relatively high (some 50% per year). Of the medium sized grazers 67% was assumed to be harvested during a whole year, while 90% of the large grazers were assumed to be harvested within a period of 4 months at the beginning of the year. Corresponding mortalities are shown in Table 3.3.

Table 3.3. Natural mortality and harvesting rates

	natural mortality	fishery mortality
cockles	0.00267	-
mussels	0.00029	0.00190
oysters	0.00029	8.34000e-005
small generalized grazer	0.00190	-

medium generalized grazer	0.00029	0.00304
large generalized grazer	0.00029	0.01900

3.4.3 Initial and reference lengths

The V1-morphs are described by a fixed reference length. This length was chosen such that the population remained more or less stable over the year. This is for example shown in Figure 5.4, where it can be seen that the state variables (specifically structural volume and the energy density) at the end of the year are more or less equal to their initial values. For the size-structured V1-models, the reference lengths were set such that the total population remained more or less stable over the year. Suitable values were found by an iterative process of trial and error, but could be found relatively easily, and the resulting values are more or less reliable.

Isomorphs grow in size, and therefore they require an initial length instead of a fixed reference length. Because size affects the growth non-linearly, and iso-morphs grow during their lifes, it was more difficult to find suitable initial values than for the V1-morphs, and it could well be that other values lead to better results. Used values are shown in Table 3.4.

3.4.4 Conversion factors

Throughout this report, the following conversion factors were used:
1 gWW = 1cm^3 = 0.12g AFDW and 1 gAFDW = 0.4 gC = 23kJ.

Table 3.4. Initial and reference lengths (cm)

	reference length	initial length
cockles V1-morphs	2.3	-
mussels V1-morphs	5.0	-
oysters V1-morphs	35.0	-
small V1 grazer	4.0	-
medium V1 grazer	5.0	-
large V1 grazer	6.0	-
small isomorphic grazer	-	3.0
medium isomorphic grazer	-	4.5
large isomorphic grazer	-	5.5

3.5 Calibration of the functional response

The main calibration parameter of the DEB model is the saturation constant 'X_k' or, more generally speaking, the main calibration function is the functional response connecting food availability and food uptake. This function also includes the effects of food quality, e.g. the effect of inorganic matter on the uptake rate.

The calibration of the functional response has been carried out by Wageningen IMARES (Wijsman et al, 2009), using the individual growth model for iso-morphs. The standard functional response function is used: $f= X/(X+X_k)$, where X is the food concentration, and Xk is the saturation constant. The food concentration, however, was calculated as follows:$X= (Chl+a_{pom}*POM)/(a_{tpm}*TPM)$, where Chl is the chlorophyll concentration in ug/l, POM is the particulate organic matter concentration in mg/l, and TPM is the total particulate matter in mg/l. The calibrated parameter values are shown in Table 3.5, and the underlying report is included as appendix C. Calibration has been carried out for each of the compartments separately, and for the Oosterschelde system in total.

Table 3.5. Calibrated parameter values and their goodness of fit, see Appendix C.

Compartiment	Calibrated parameter value			Sum of squared residuals		
	X_k	α_{POM}	α_{TPM}	L	AFDW	Total
West	2.36	0.02	0.11	5.4	14.0	19.4
Central	2.47	0.02	0.11	3.8	13.6	17.4
North	4.76	0.17	0.22	3.4	17.0	20.3
East	3.62	0.04	0.16	7.9	27.6	35.4
Total	3.21	0.27	0.15	4.6	16.5	21.0

4 Integrated ecosystem model - Basic

As a next step, the cockle model was incorporated in a generic ecosystem model (GEM). First, a very simple model grid was used consisting of a single grid cell with a surface area, depth, and nutrient loading roughly based on those of the Oosterschelde (flow rate= 20.4E+03 m/s, cross-sectional area= 81000 m^2, surface area= 304E+06m^2, volume=2750E+06m^3, length=27700m), which model was used before and described in more detail in (Van de Wolfshaar, 2007).

At the time this step was carried out, the calibrated parameter values for the functional response were not yet available, and a provisional function was used in which only algal and detrital concentrations were involved, with a the saturation constant of 0.5 mug/l. Since the integrated model calculates algal and detritus concentrations, they can be used directly to calculate food concentration (assuming an equal preference of the cockles for algae and detritus), and chlorophyll concentrations are no longer necessary as a proxy. The mean chlorophyll content of algal dry matter and detritus was calculated on basis of the results of a simulation period of one year and was found to lie around 0.004. The saturation constant X_k was adjusted accordingly (X_k=0.5 mg/l).

The integrated model in this chapter takes into account all feedbacks of cockles on their surroundings. These feedbacks include the direct feedback effect of grazing leading to a decrease in local algae concentrations. In addition, however, the cockles may also *in*directly affect algal concentrations through an increase of nutrient concentrations by respiration, and through a decrease in algal growth due to the diminished algal biomass. Furthermore, the shellfish also affect their surroundings through an increase of detritus in the sediment by mortality and faeces production.

Another difference with the models of the previous sections is that the GEM takes into account food *quality,* by considering the stoichiometry of algae and detritus. As algae have a much lower phosphorous content than the cockles, part of the food is turned into faeces, and can thus not be used for growth.

4.1 Small grazer densities

To test the results of the integrated ecosystem model for consistency with those of the calibrated cockle model from the previous chapters, the integrated model was first run with small initial cockle densities. At small densities, grazing hardly affects food concentrations, in which case the results should be very similar to those of the non-feedback model-variant and differences can only arise from the effects of food quantity and quality, which should then be corrected for.

Initial results showed unrealistically large food losses via (pseudo)faeces, which made clear that some adjustments were needed indeed. The first adjustment involved the assimilation efficiency, which was set to one (AE=1). Underlying idea is that the assimilation efficiency in the simple model is a proxy for the stoichiometrical losses, which was thus no longer needed.

Second adjustment considered the CPN-ratio of the grazers, which was adjusted such that the stoichiometrical losses added up to 75% of the total food uptake, though still with a fixed ratio between N and P (T_N=0.11 T_P=0.016). Underlying idea was that this

CPN-ratio does not represent the actual CPN-ratio of the grazer's biomass, but that it represents the minimum stoichiometric ratio of food required by the grazers. This required ratio is below the ratio of the grazer biomass because the food is not only used to generate new biomass but also to pay maintenance and overhead costs for which nitrogen or phosphorous are not essential.

After the above adjustments were made, results still showed less growth of the cockle population in the integrated model than in the simple model. This turned out to be due to the fact that the overall food concentration calculated in the integrated model is smaller (the bloom starts later) than the food concentrations forced into the simple model. Apparently, the modeled cockles of reference length 2.3 cm cannot handle these small food concentrations. The used reference length, however, was based on cockle samples taken in very suitable cockle areas, and may overestimate the average reference length. When decreasing the reference length to 1.5, results are very similar to the results of the non-feedback model-variation with a forcing function for chlorophyll as proxy for food as used in the previous chapters. Figures 5.1a shows the resulting structural density (cm^3/m^2), energy density (J/m^2) and gonadal density (J/m^2). The blue curves in Figure 4.2 show the corresponding nutrient and chlorophyll concentrations. In the remainder of this report, however, the reference length was kept at 2.3cm.

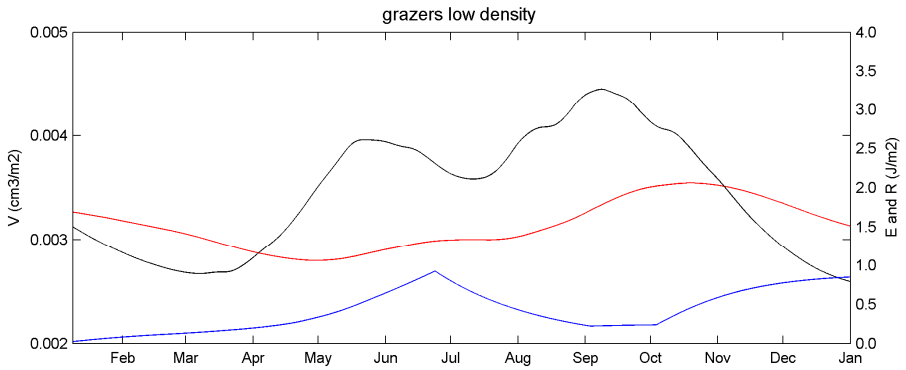
4.2 High grazer densities

Next, the model was run for higher initial cockle densities, at which the grazers do have a feedback effect on their food. The initial densities were set such that the grazers were more or less in steady state with their surroundings (see Figure 4.1B). Results showed that at a larger initial biomass the net growth of the grazers becomes negative, and the grazers are no longer at steady state.

Results show that, in this simple model, a maximum biomass of around some 1000 cm^3/m^2 (~1kgWW/m2) can be sustained. This predicted maximum density is more than the real densities that are observed in the Oosterschelde, especially when they are averaged over the whole area: around the year 2000, a total of some 5.9 million kg AFDW of shellfish was observed in the Oosterschelde (Geurts van Kessel et al., 2003). This can be converted into cm3/m2 by multiplying it by 1000 g/kg and 8.3 gWW/gAFDW, and dividing it by 304E6 m^2, while assuming that 1 cm^3 of biomass weighs approximately 1g. This results in an observed average density of some 160 cm^3/m^2 (~160 gWW/m^2).

Clearly, this simple ecosystem model overestimates cockle densities. This may be well due to a lack of heterogeneity. Some locations may be less productive than others, some substrates or other environmental conditions may be unsuitable for shellfish growth, and at some intertidal locations feeding may not be always possible. To investigate this further

A



B

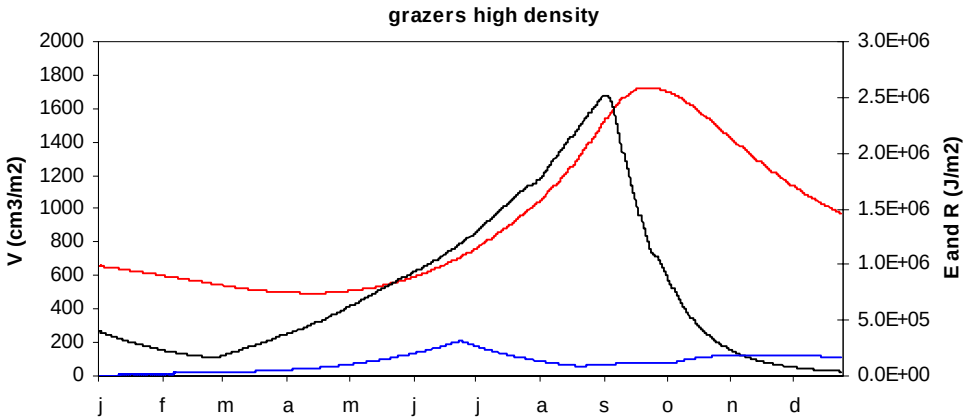


Figure 4.1. Structural density (red curve, left axis, cm^3/m^2), energy density (black curve, right axis, J/m^2) and gonadal density (blue curve, right axis, J/m^2) simulated over a year in a basic integrated ecosystem model, when starting from small (A) or large (B) initial densities.

5 Integrated ecosystem model - Oosterschelde

In this chapter, the simple ecosystem model was extended into a more realistic and spatially explicit ecosystem model. First, an inventory was made of the available ecosystem models for Delta systems in Delft3D. An overview of the current state of these models is given in Table 5.1. Although from this inventory it became clear that no useful ecosystem models were available for the Oosterschelde, it was decided during the workshop that the Oosterschelde system was the most suitable Delta system to use in this project.

5.1 Advantages and disadvantages of modeling the Oosterschelde

The reason that the Oosterschelde was chosen as the most suitable system is because it is a grazer dominated system, with many different cultured and natural populations, including cockles, mussels, oysters and *Ensis*. Moreover, more data is available on the shellfish in the Oosterschelde than of the shellfish in other systems.

The grazer-dominance in the Oosterschelde is considered an advantage for this project, because grazers affect the system's nutrient cycling, and the model predictions on nutrients will only fit measured values when the grazers are included properly in the model. A good fit will thus suggest a good grazer modeling, and provide a test of the grazer module.

A disadvantage of the grazer-dominance in the Oosterschelde, however, is that the model is not very useful for simulating specific shellfish species (such as the cockles) in absence of other grazers. To be able to study specific shellfish populations in grazer-dominated systems, some form of background grazing in the GEM is essential. Including background grazing may be done in various ways that range from simple solutions (increasing the algae sedimentation rates), via intermediate ones (including other grazing modules), to more complex ones (simultaneously modeling various shellfish species and populations). The first option will be a great oversimplification of the actual grazing in the Oosterschelde, the second option is not desirable, and the third is not yet feasible (though this project forms a building block). Yet, as a finger exercise, it was decided to incorporate our cockle model in the GEM without any other grazing. As such, the results will not represent specifically the cockle populations in the Oosterschelde, but they should at best be considered as general grazers.

Another disadvantage of modeling shellfish in the Oosterschelde is that a useful ecosystem model of the Oosterschelde did not exist yet in Delft3D, which thus implied that a new Oosterschelde integrated model had to be set up. Such a model should consist of an Oosterschelde grid, a hydrodynamic model (FLOW) and a generic ecosystem model (GEM). These models are discussed in the next few sections. To make perfectly calibrated and validated FLOW and GEMs takes a lot of work. In this project, however, the focus lies on shellfish grazing, and it was decided to put a balanced amount of effort into setting up the Oosterschelde models. As a result, their fit may not be perfect, and they may not be suitable as such for direct use in other projects. However, they can still act as a useful starting point in case more accurate models are needed.

Table 5.1. Inventory and status of integrated ecosystem models for Delta water systems available in Deltares.

	Volkerak-Zoommeer	Volkerak-Zoommeer	Grevelingen	Veerse Meer	Wester-schelde	Ooster-schelde	Ooster-schelde
2D/3D	3D (z lagen)	3D (sigma lagen)	3D (z-lagen)	3D (z-lagen)	2D	2D ScalOost	2D Zuno
starting point	GEM-model	GEM-model	mdf-file	bloom-model van tape	mdf-file	SDS-file	GEM-model
obtained from	R.Hulsberg en	E. Meijers	F.Zijl	A.Nolte	L.Arentz	A.Nolte	A. Blauw
flow-files	nvt	nvt	x	nvt	x	nvt	nvt
coupling	nvt	nvt	G. de Boer	nvt	x	x	nvt
boundaries	x	x	x	x	x	x	nvt
loads	x	x	x	x	x	x	x
inorganic matter			constant	constant	-	time series (A. Blauw)	time series (A. Blauw)
aggregation grid			van 14 naar 13 lagen	van 31 naar 8 lagen	-	2x2 (hor.agg)	-
bug-free run	x	x	x	x	x	x	x
continuity checked			x	x		.	x
salinity checked			x				
balance checked		x	x				x
calibrated		x	.	x			x

5.2 Oosterschelde bathymetry and GRID

For an integrated ecosystem model of the Oosterschelde, first a modeling grid is required. As the Oosterschelde-grid in the ZUNO-model does not have sufficient resolution, the ScalOost grid was used as a basis for the FLOW model. To decrease the resolution of the grid, it was aggregated with the standard option in RFGRID. The remaining grid consists of 3277 (active) cells (see Figure 5.3) in the horizontal, and 10 layers in the vertical.

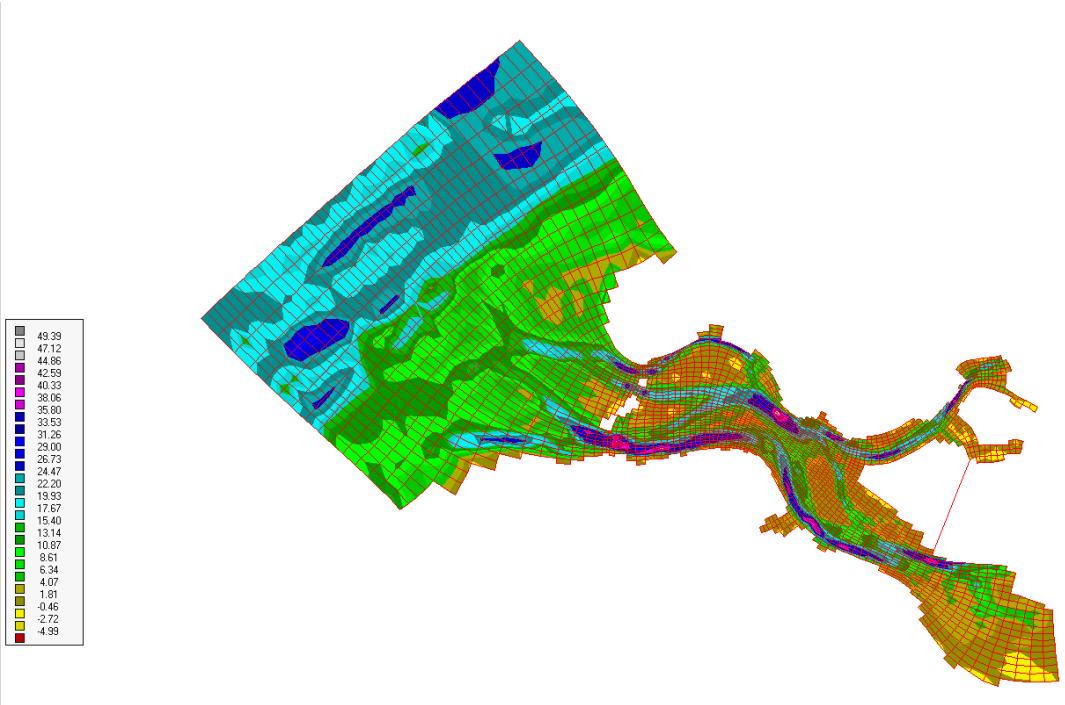


Figure 5.1. Grid and bathymetry of the Oosterschelde

5.3 Oosterschelde FLOW

Delft3D-FLOW software is used to calculate the hydrodynamics of the Oosterschelde. The redefined ScalOost grid was used as a basis for the FLOW model. The boundary water levels were obtained from the ZUNO-model using the nesting procedure provided through the Delft3D interface. Discharge quantities were obtained from the 1D-Deltamodel (Meijers et al 2007). The number and thickness of the layers is set equal to the ZUNO-model (N=10) to enable the nesting procedure. For the same reason, the wind-time series is copied from the ZUNO-model. Output options and physical parameters are set equal to those in the Grevelingen project (Nolte et al, 2008).

The coupling of the FLOW and the GEM was done on the fly (‘online’) with a time step of 1 minute. Too large flow-velocities were corrected by running the custom program “flow-check” which artificially adjusts the involved cell-volumes.

5.3.1 Validation by salinity

The FLOW model was validated by means of salinity only, since temperature measurements were not available. In Appendix A1, the simulated salinity

concentrations (ppt) resulting from the 3D FLOW model in the top layer at four locations are shown (Wissenkerke, HammenOost, Lodijksegat, and Zijpe), together with the measured concentrations.

The FLOW model was improved by including precipitation and evaporation. These two processes were found to be important to the salinity concentrations, especially since the simulated year (1998) was a very wet year. Evaporation increases the salinity concentration during summer, while precipitation decreases salinity concentrations in spring and fall. Both processes were included as an additional discharge at one central location per compartment (for evaporation also an additional discharge-location in the Voordelta was included). As a result of precipitation, the values in the eastern and northern compartment become quite variable, which is to some extent also present in the measurements.

Apart from its direct effects, precipitation also has some indirect influences. One of these is the inflow through the Krammer sluices. These sluices show peak-discharges at moments of large rainfall, which highly dominate the salinity concentrations in the Northern compartment. As a consequence, the salinity concentration gradient in the northern compartment is quite steep, and the results highly depend on the exact location that is chosen. When for instance choosing another location close to the original one, the periodicity disappears and the fit improves (blue curve).

Another indirect effect of precipitation is the inflow of freshwater from the Haringvliet through the northern boundary close to the coast. It was found that the salinity concentrations in the western and central compartment are highly dominated by the salinity concentrations of the water coming in over this boundary, and this inflow is greatly responsible for the large drop in salinity concentration in fall and winter in these two compartments. To reduce the initially overestimated decrease in salinity concentration during this period, the boundary conditions at the northern border close to the coast were moderated. This was done by using the concentrations from boundary NO2 for the range of boundaries NO3 – NO7. This improves the fit for salinity concentrations (as are shown), but slightly decreases the fit for nutrient and chlorophyll concentrations.

Appendix A.2 shows the salinity concentrations in case the FLOW model results are converted from 3D to 2D, so that they can be used in the 2D GEM. For this conversion, the FLOW results are averaged over the vertical. As a result, the concentrations become less variable, and in some cases the deviations between simulated and observed concentrations increase. This is especially the case for location Lodijkse Gat, where the salinity concentration during spring time is around 1 ppt too high. This is however considered to be acceptable for the water quality and ecology simulations that follow.

Another effect of converting the system from 3D to 2D is the periodicity emerging in the salinity concentrations in the northern compartment. This periodicity is the result from sampling over the tides. During high tides saline water flows into this compartment, while during low tide, the saline water retreats and the fresh water from the Krammer sluices penetrates further into the system. Because sampling occurs at fixed moments, and the tidal cycle is slightly longer than a day, the tide at which the sampling occurs shifts throughout the year.

5.4 Oosterschelde GEM set-up

Standard Delft3D-software was used to construct a Generic Ecological Model (GEM). The GEM calculates the concentrations of nutrients (nitrate, ammonium, phosphate, silica), dissolved oxygen and salinity, phytoplankton (diatoms, flagellates, dinoflagellates and *Phaeocystis*), and detritus. The following processes are simulated:

- phytoplankton processes: primary production, respiration and mortality.
- light extinction
- decay of organic matter in water and sediment
- nitrification and denitrification
- reaeration
- sedimentation and resuspension
- burial of organic material

5.4.1 Forcing functions

The GEM uses the hydrodynamics as calculated with Delft3D-FLOW. Other input for GEM consists of meteorological conditions, initial values, concentrations on the boundaries and discharges, en various other forcing functions. The polder loads and locations were obtained from the 1D Deltamodel (Meijers et al 2007). Meteorological conditions were obtained from KNMI, station Wilhelminadorp. Initial values of state variables are determined by carrying out a pre-run, the resulting values of which are used as initial values.

A time-dependent forcing function was used for inorganic matter. This function was based on measured concentrations, and describes a gradient from higher concentrations in the west to smaller concentrations in the east (Figure 5.2).

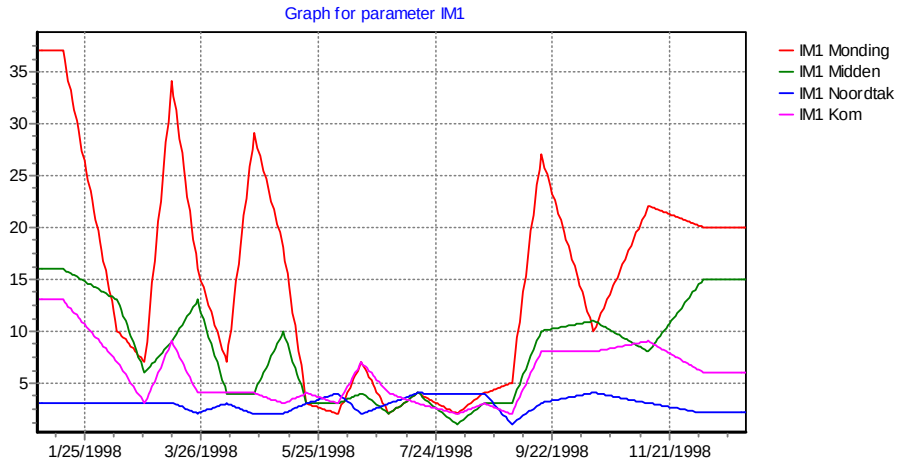


Figure 5.2 Inorganic matter forced in each of the compartments.

5.4.2 Validation measurements

- Nutrient and chlorophyll concentrations

For nutrient and chlorophyll concentrations, MWTL measurements are available for the locations Wissenkerke, HammenOost, Lodijkse gat, and Zijpe.

- Primary production

In some years (including 1998, which is the year that was simulated in the present study) the primary production of the phytoplankton has been measured by the NIOO-CEME at a bi-monthly interval at five locations in the Oosterschelde: Lodijksche Gat (east), Zandkreek (central), Zeelandbrug (central), Roompot (west), and the en de "Keeten-Krabbenkreek" (north). Measurements were converted to daily averages, which were linearly interpolated and summed to obtain the total yearly production (Wetsteyn et al 2003).

It should be noted that measurements of primary production are knowingly problematic. Various assumptions are needed in converting the measurements to daily averages. This gives them a different status then other variables that are measured much more straightforwardly (such as nutrient concentrations), and they should be considered with proper caution.

Also, measurements may vary with weather and other conditions such as the presence or absence of shellfish on the purging plots. Furthermore, the measurements were done at 5 locations only, and these were considered to be representative for the compartments in which they were located. No corrections were made for the intercompartmental variation in primary production. It is therefore recommended that these variables are measured more frequently and on more locations.

- Turnover time

The turnover rate of the phytoplankton was measured as the rate of nutrient cycling, from the moment of uptake by phytoplankton until return after mortality of the plankton. The turnover rate is therefore a measure for the growth rate of the phytoplankton, and can be used as an indication of the production rate of food for grazers. The turnover *time* is calculated as 1/(turnover rate). Measurements were done by the NIOO-CEME (Geurts van Kessel, 2003).

Like the measurements of primary production, those for turnover times are knowingly problematic. Again, various assumptions are needed to convert the raw measurements into turnover times. This gives them a different status then other variables that are measured much more straightforwardly (such as nutrient concentrations), and they should be considered with proper caution. Furthermore, the available values are yearly averages, which do not provide any information on the interannual variation.

5.5 Oosterschelde GEM without grazing

As a reference run, the GEM was first run without grazer module.

5.5.1 Validation by nutrient concentrations

Nutrient and chlorophyll concentrations as resulting from the 2D and 3D GEM are shown in Appendix A.3 to A.6. Simulated and measured concentrations are shown for four locations. Overall, the agreement of the model results with the observed concentrations is better at the western locations than at the eastern ones. This is probably partly due to the fact that the east is more dominated by grazers than the

west, while grazers are totally absent in the model. It was decided to wait with further calibration until grazing has been included in the model.

- Chlorophyll concentration (Appendix A.3)

The modeled chlorophyll concentrations in the western part of the Oosterschelde coincide well with the measured concentrations. Although the nutrient concentrations in the east are comparable to those in the west, the chlorophyll concentrations in the east are too high and the bloom starts too early. The difference between east and west may be caused by the shallowness of the eastern Oosterschelde, where the algae are not mixed to deep waters and thus suffer less from light limitation. Another reason may be the smaller concentration of inorganic matter in the east. The deviation between simulated and observed concentrations may be esplained by the nutrient concentrations that are too high (see next bullet). Another obvious reason may be that grazing is not yet included in the model.

- Nitrate concentration (Appendix A.4)

Like the chlorophyll concentrations, the simulated nitrate concentrations in the west coincide well with the observed concentrations, whereas the fits of the concentrations in northern and eastern locations are not good. Specifically, the concentrations at location Zijpe are too high. A reason for this mismatch may be that the nutrient concentrations of the discharges at the Phillipsdam are not known. Instead, nutrient concentrations measured at location Steenbergen are used in the simulation. It is possible that these concentrations are too high. Although a decrease by 20% in these concentrations did lead to an overall decrease in the nitrate concentrations simulated in the northern compartment, it did not lead to an improved fit with the observations. Therefore, it seems that the inflowing concentrations are not only too high, but also that their trend is wrong.

Furthermore, the curve at location Zijpe shows a periodic pattern. Similar to the simulated periodicity in salinity (Section 5.3.1), the periodicity in nitrate concentration is the result from sampling over the tidal spring-neap cycle. During high tides, saline water with small nitrate concentrations flows into the northern compartment, while during low tide, the saline low-nitrate water retreats and the fresh water with high nitrate concentrations from the Krammer sluices penetrates further into the northern compartment. As with salinity, the nitrate gradient in the northern compartment is quite steep, and the results are sensitive to the exact location that is chosen.

- phosphate concentration (Appendix A.5)

Simulated phosphate concentrations in the shallow eastern areas are clearly too low during summer. This is a known modeling problem that is related to the remineralization of PO4 from the sediment, which process is difficult to model deterministically. The fit may be increased by introducing a forcing function that supplies PO4 during the summer. However, also the inclusion of grazing will affect the PO4 concentration, because respiration of the shellfish will lead to nutrient recycling. Therefore, it was decided not yet to include a forcing function on PO4 remineralization, but to wait with calibration until grazing has been included.

- ammonium concentration (Appendix A.6)

Ammonium is a substance mainly produced by higher organisms, and thus is suitable for validating grazing. In contrast to measured ammonium concentrations, the simulated concentrations in absence of grazers remain very small throughout the year.

5.6 Oosterschelde GEM with generalized grazer

As a next step towards a more realistic GEM, we include a generalized grazer that is constant in size (V1-morph). The parameter values are based on those of mussels, as is explained in Section 3.4.

5.6.1 Initial grazer distribution

The initial distribution is based on the estimated shellfish stocks around the year 2000 (see Table 5.2, after Geurts van Kessel et al., 2003). This distribution includes mussels, cockles, sublitoral and litoral oysters. In the eastern compartment, an additional density of cultivated mussels is included to take into account the mussels that are brought there to purge (‘verwater’).

Cockles were restricted to areas on or close to mudflats (areas with a depth < 2m), see Figure 5.3A. Mussels were located on dedicated mussel farms, which were appointed in the schematization on basis of figure 1.1 (see Figure 5.2B). Sublitoral and litoral oysters were restricted to areas with a depth<1, and litoral oysters to areas with a 15<depth<1 (see Figure 5.3C).

The AFD-weights (10^6kg) were converted into grams by multiplying them by a factor 10^9, and then converted to gWW by multiplying them by a factor 8.3. Next, one third of the weight was subtracted to cover for the weight by energy reserves and reproductive material. Finally, the weights were multiplied by a factor 0.8 to take into account the smaller winter weights. It was assumed that 1 gWW is equal to 1 cm^3 of volume.

The densities were calculated per species by dividing their total biomass per compartment by the total surface area at which they occur. Then, the densities of the separate species were summed, resulting in a total initial density for a generalized grazer as shown in Figure 5.4.

Table 5.2. Total biomass (million kg AFDW), after (Geurts van Kessel et al., 2003).

	West	Central	North	East
mussels	0.8	1.05	0.19	0.5
cockles	0.25	0.3	0.19	0.1
litoral oysters	0.4	0.4	0.28	0.35
sublitoral oysters	0.6	0.5	0.14	0.35
total	2.05	2.25	1.00	1.30

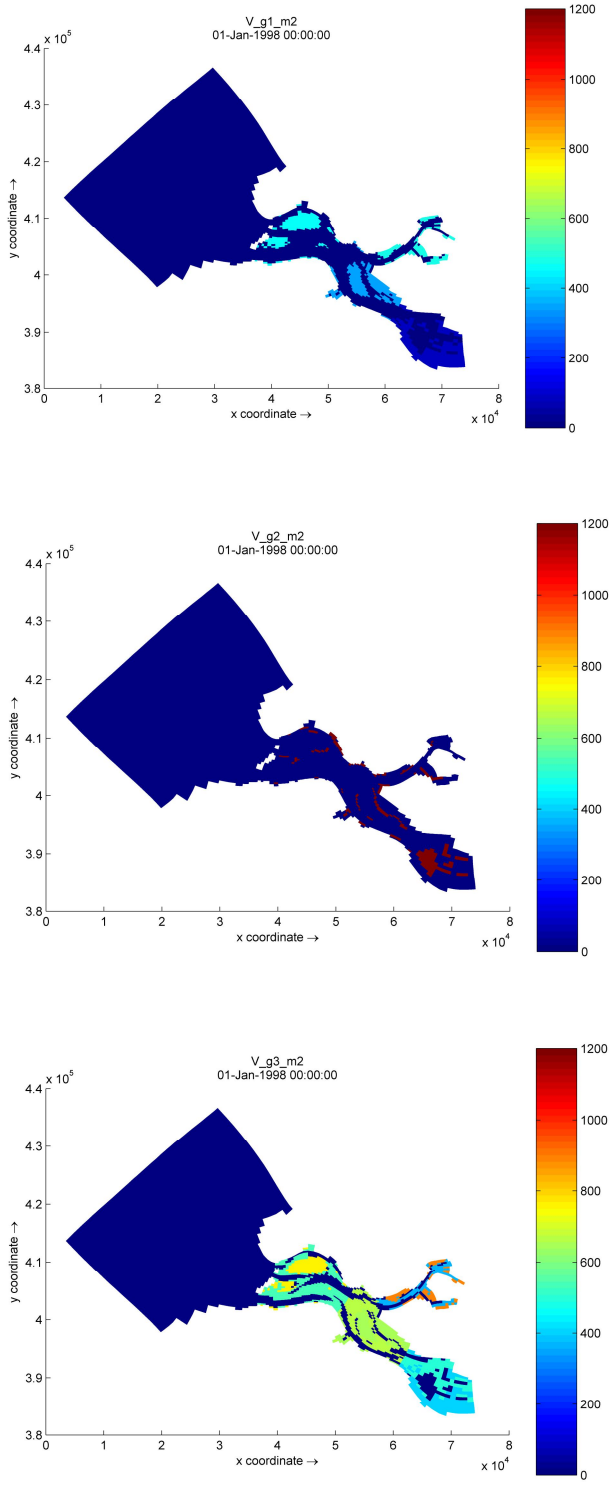


Figure 5.3. Initial densities (gWW/m2) of cockles (A), mussels (B), and oysters (C).

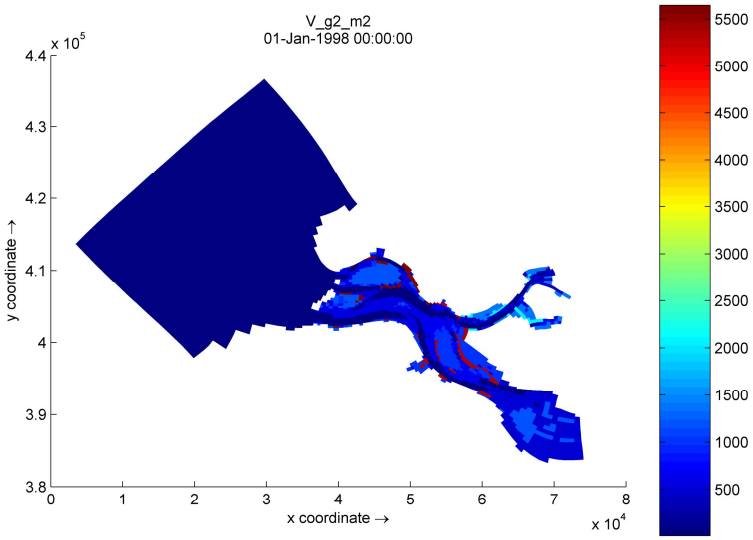
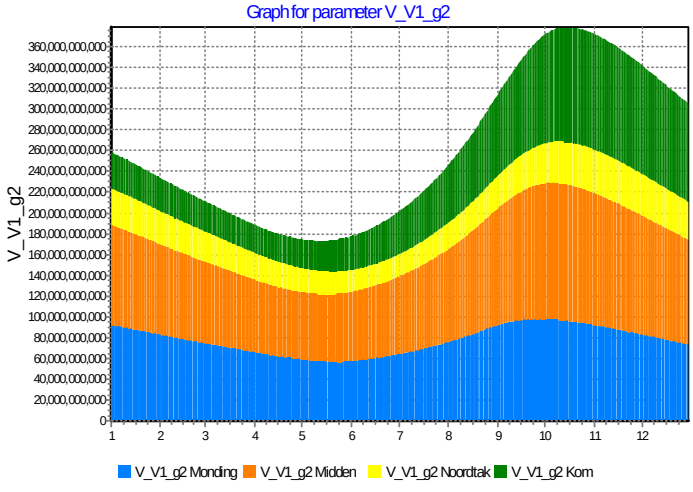


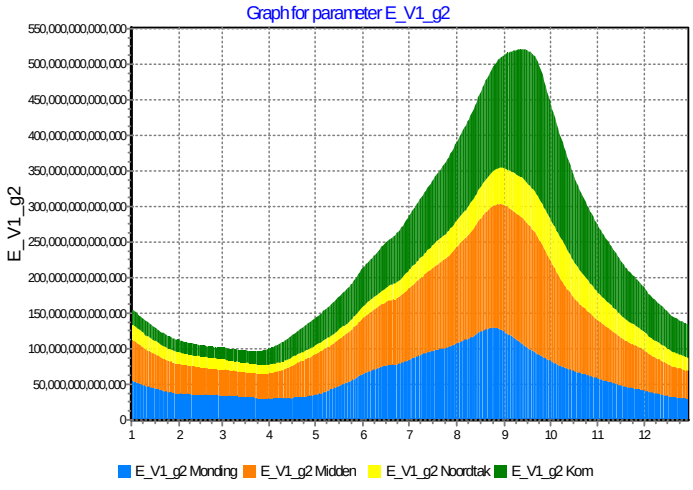
Figure 5.4. Initial distribution (structural volume gWW/m2) for a generalized grazer.

5.6.2 Results for a generalized grazer

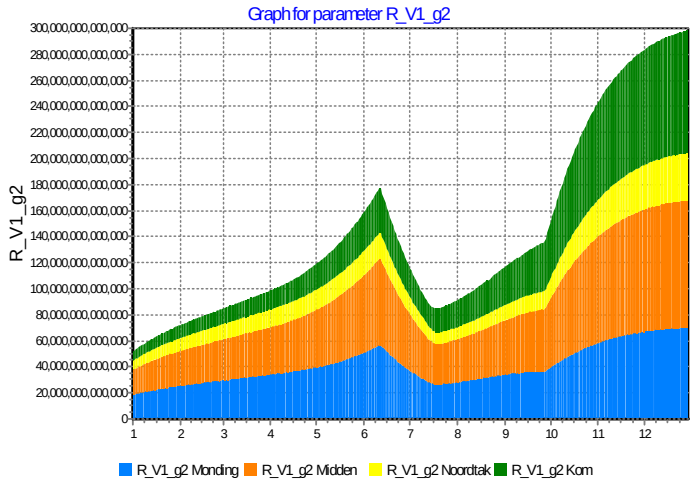
To give an impression of how the generalized grazer develops over time, Figure 5.5 shows its simulated state variables.



A



B



C

Figure 5.5 Simulated structural volumes (gWW) (A), energy reserves (Joules) (B) and reproductory volumes (Joules) (C) in the western (blue), central (orange), northern (yellow) and eastern (green) compartment.

5.6.3 Validation by nutrient concentrations

In appendices A.3 to A.6, the nutrient concentrations from the GEM including grazers are compared to those resulting from the GEM without grazers, as well as to observed nutrient concentrations.

The figures (A3.-A.6) clearly show that the overall model fit improves considerably, especially at the eastern (Lodijkse gat) and northern locations (Zijpe). This is also clear from Figure 5.6, where the fraction of the number of measurements is shown at which plankton bloom absence (chl-conc<5) and presence (chl-conc>5) were predicted correctly. Chlorophyll concentrations in western (Wissenkerke) and central locations (Hammen Oost), however, change only slightly, and their fit with observations slightly deteriorates when including grazers.

A similar pattern goes for ammonium. Nitrate concentrations improve for Lodijkse gat, but the concentrations in Zijpe are still too high. Phosphate concentrations do not fit the observations nutrients have not changed much.

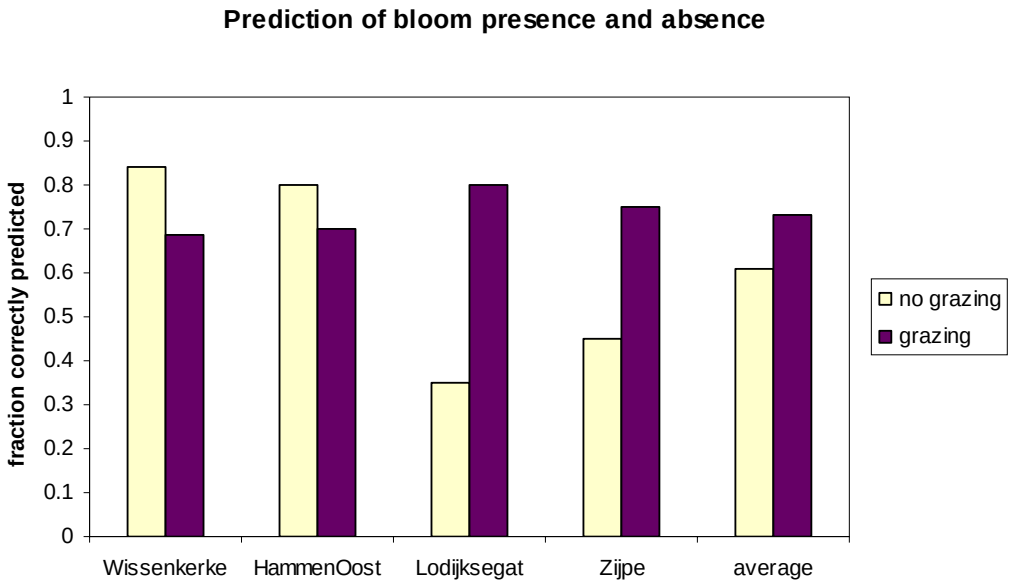


Figure 5.6 Fraction of measurements at which plankton bloom presence and absence was predicted correctly, for four locations in the Oosterschelde using the model with grazing (purple) and without grazing (yellow).

5.6.4 Effects of grazing on carrying capacity

In almost all compartments, the predicted primary productions are too small and turnover times are too large when not taking into account grazing (Figures 5.7 and 5.8). When including grazing, primary productions increase and turnover times decrease. Overall, the prediction of the system's carrying capacity thus improves considerably when including grazers.

In the northern and eastern compartment, however, the predictions of primary production and turnover times deteriorate when taking into account grazing. This is

surprising, as the chlorophyll concentrations in specifically these two compartments improve when including grazers. An explanation for this deviation could be that the measured values themselves are not very reliable. These variables are knowingly difficult to measure and require various assumptions to convert them into yearly numbers. This gives them a different status then other variables that are measured much more straightforwardly (such as nutrient concentrations), and they should be considered with proper caution (see also section 5.4.2).

The inclusion of an additional initial grazer biomass in the eastern compartment (on the purging plots ('verwater percelen')) results in a later phytoplankton bloom and greatly improves the fits of predicted chlorophyll concentrations with measurements. The fact that the system is sensitive to such an additional biomass, indicates that the grazing pressure is high. Yet, the predicted primary productions in this compartment are much larger than measured.

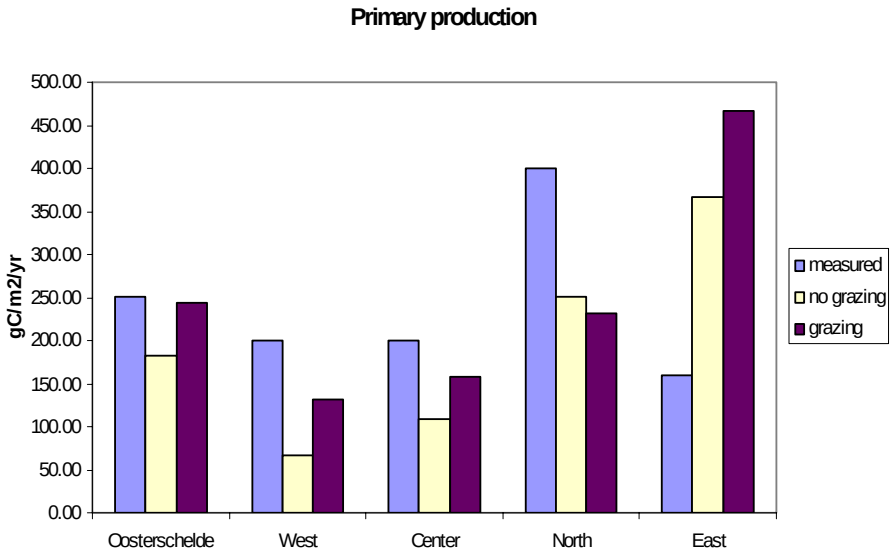


Figure 5.7. Primary production in each of the four compartments of the Oosterschelde, measured (blue) and simulated by the model with (purple) and without grazing (yellow).

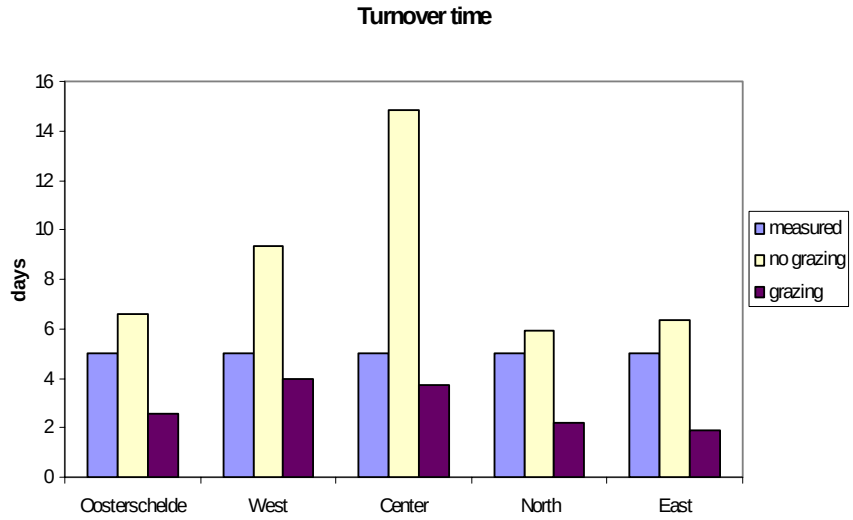


Figure 5.8. Turnover times in each of the four compartments of the Oosterschelde, measured (blue) and simulated by the model with (purple) and without grazing (yellow).

5.6.5 Nutrient balances

Tables 5.3 and 5.4 present the water balances for nitrogen fluxes in and out of the watercolumn and the sediment.

Table 5.3. Nitrogen fluxes in and out of the watercolumn for the whole Oosterschelde

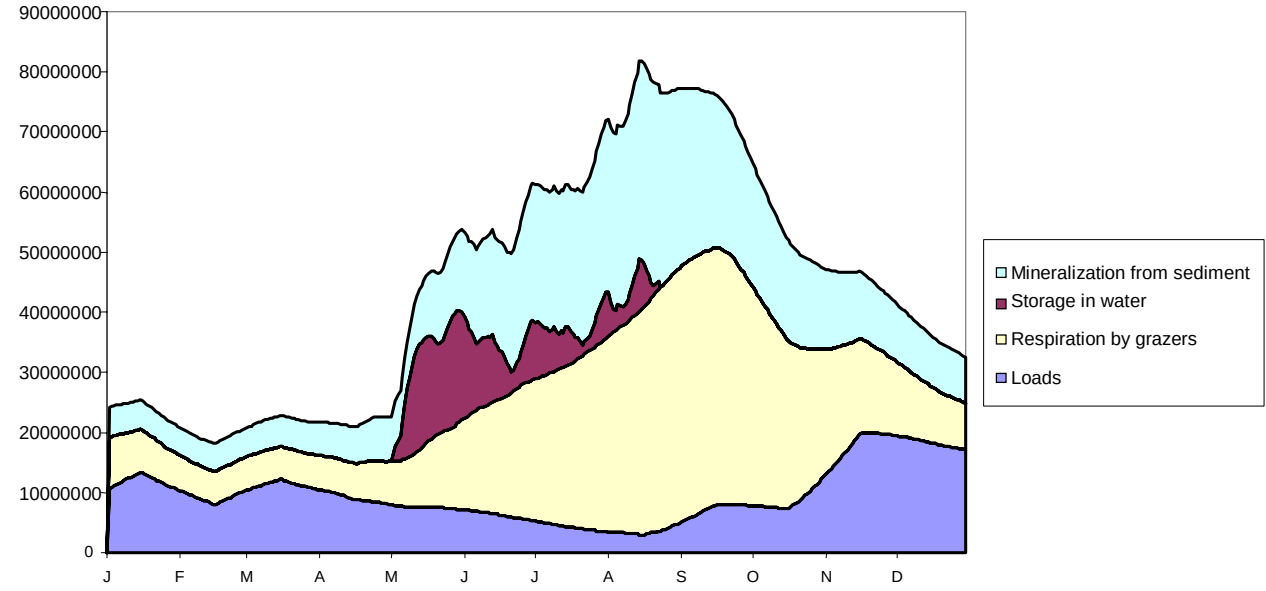
Nitrogen in water	Sources/Inflows	Sinks/Outflows	Sources/Inflows	Sinks/Outflows
	gN/yr	gN/yr	%	%
loads	3,47E+09	0,00E+00	25	0
transport in/out NorthSea	0,00E+00	-2,14E+09	0	16
denitrification	0,00E+00	-4,74E+08	0	3
mineralisation sediment	3,82E+09	0,00E+00	28	0
sedimentation	0,00E+00	-3,09E+09	0	23
grazing	0,00E+00	-7,70E+09	0	57
respiration grazers	6,34E+09	0,00E+00	47	0
storage in water	0,00E+00	-1,50E+08	0	1
total	**1,36E+10**	**-1,36E+10**	100	100

Table 5.4. Nitrogen fluxes in and out of the sediment for the whole Oosterschelde

Nitrogen in sediment	Sources/Inflows	Sinks/Outflows	Sources/Inflows	Sinks/Outflows
	gN/yr	gN/yr	%	%
mineralisation	0,00E+00	-3,82E+09	0	35
sedimentation	3,09E+09	0,00E+00	28	0
grazing	7,70E+09	0,00E+00	70	0
respiration grazers	0,00E+00	-6,34E+09	0	58
storage in grazer	0,00E+00	-7,27E+08	0	7
storage in sediment	2,47E+08	0,00E+00	2	0
loss grazer (spawning&harvest)	0,00E+00	-1,53E+08	0	1
total	**1,10E+10**	**-1,09E+10**	100	100

Figure 5.9 shows the nitrogen fluxes into (A) and out of (B) the water column (including NO3, NH4, algae, and detritus) as they occur throughout time. Positive fluxes include the river loading of nutrients into the system, the mineralization flux from the sediment, the respiration flux from the grazers into the water column, and (net) storage in the water column. Negative fluxes include the (net) transport to the North Sea, the sedimentation fluxes of algae and detritus, denitrification in the water column, and the grazing of algae and detritus.

A



B

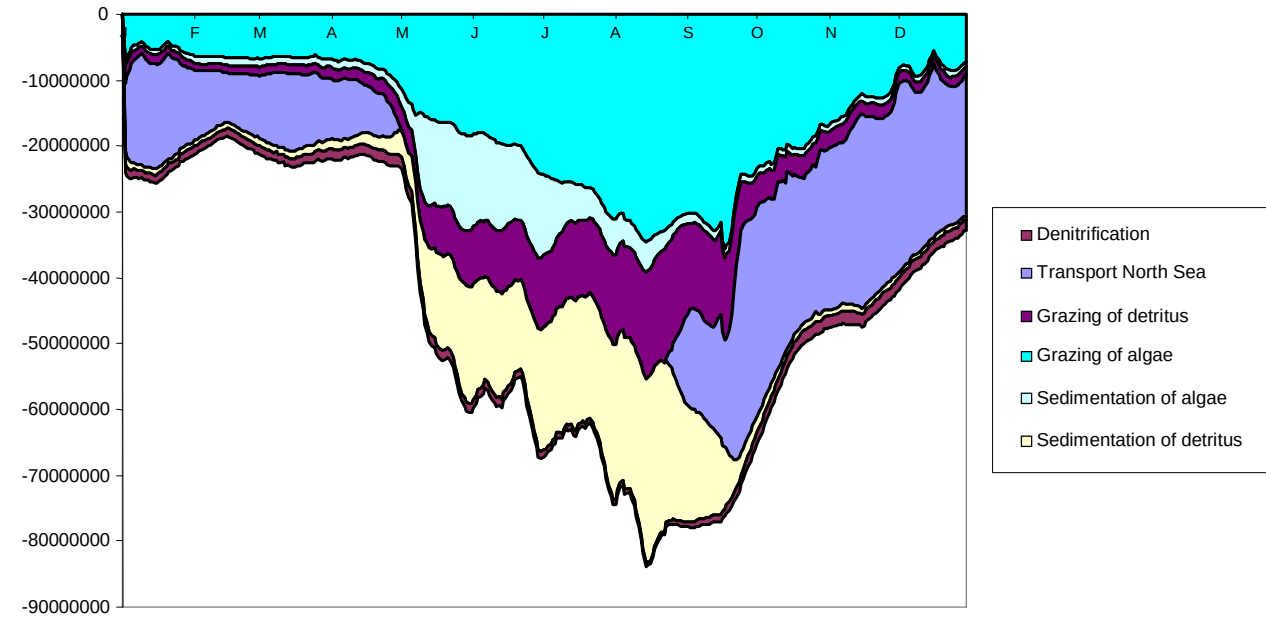


Figure 5.9. Nitrogen fluxes into (A) and out of (B) the water column (including NO3, NH4, algae, and detritus) as they occur throughout time.

5.7 Oosterschelde GEM with size-structured grazing

In this section, three models with generalized grazers of increasing complexity are compared. More specifically, the model with one unstructured grazer (V1-morph) is compared to a model including three size-classes of V1-morphs, and to a model including three size-classes of iso-morphs.

5.7.1 Initial grazer distribution

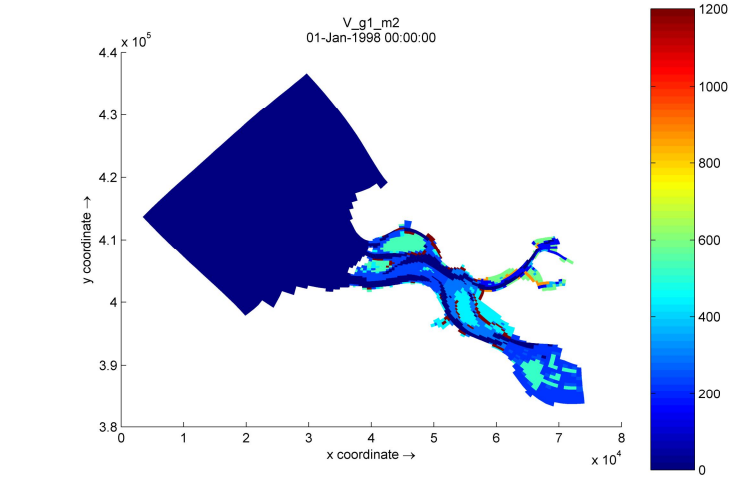
Like before, the total biomass was based again on the estimations in Geurts van Kessel et al. (2003), and the conversion from biomasses into densities is also as explained in section 5.6.1.

The three size-classes were initialized by assuming that small and medium individuals each constitute 43% of the total biomass, while large individuals constitute 14%. These percentages were based on those used in the Keyzones-project (Blauw et al, 2007).

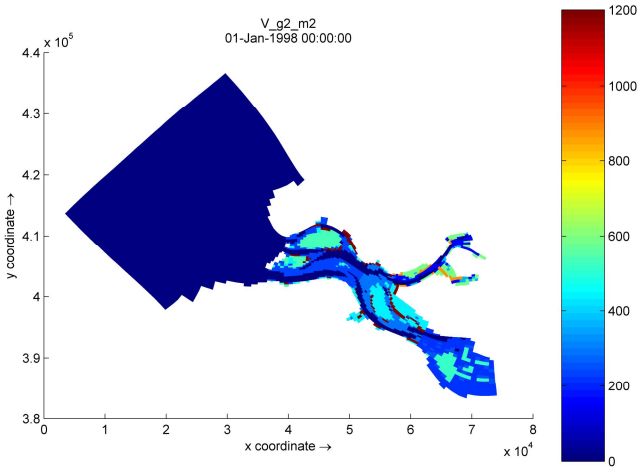
The resulting initial biomasses per size class are given in Table 5.5, the initial distributions are shown in Figure 5.10.

Table 5.5. Estimated initial biomass per size class and compartment (million kg AFDW).

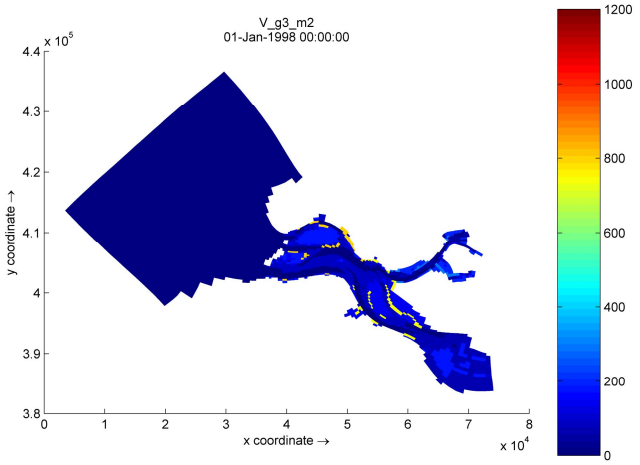
	West	Central	North	East
small	0.88	0.97	0.43	0.56
medium	0.88	0.97	0.43	0.56
large	0.29	0.32	0.14	0.18
total	2.05	2.25	1.00	1.30



A



B



C

Figure 5.10. Initial densities (gWW/m2) of small (A), medium (B), and large (C) grazers.

5.7.2 Size-structured results

Figure 5.11 gives an impression of the change in density of the size-structured grazers as resulting from the model. The grazers from the large size class are all harvested at the beginning of the year, while the biomass of small grazers increases considerably. The biomass of medium sized grazers stays more or less constant during the year.

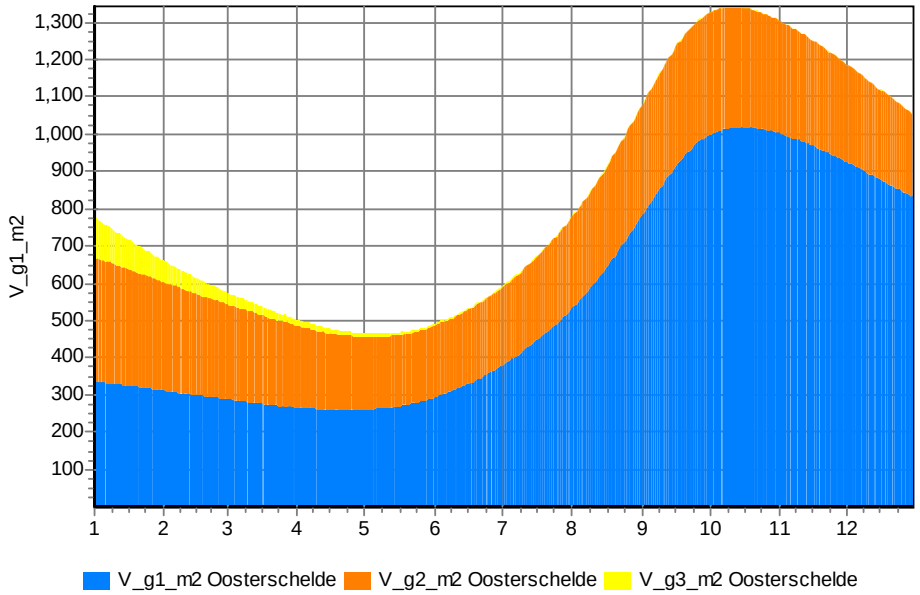


Figure 5.11. Density of structural volume averaged over the Oosterschelde of small (blue), medium (orange), and large (yellow) grazers as resulting from the size structured model.

5.7.3 Validation by nutrient concentrations

Nutrient concentrations resulting from the size-structured models do not differ much from those predicted by the unstructured model (Appendix A7 – A10). The largest difference occurs with regard to chlorophyll concentrations in location Zijpe, which are reduced to almost zero, indicating that the system is being overgrazed. This is probably due to the small grazers which put relatively more grazing pressure on the system.

This deterioration of the model performance is also shown in Figure 5.12, which shows the fraction of correct predictions with regard to the absence or presence of an algal bloom.

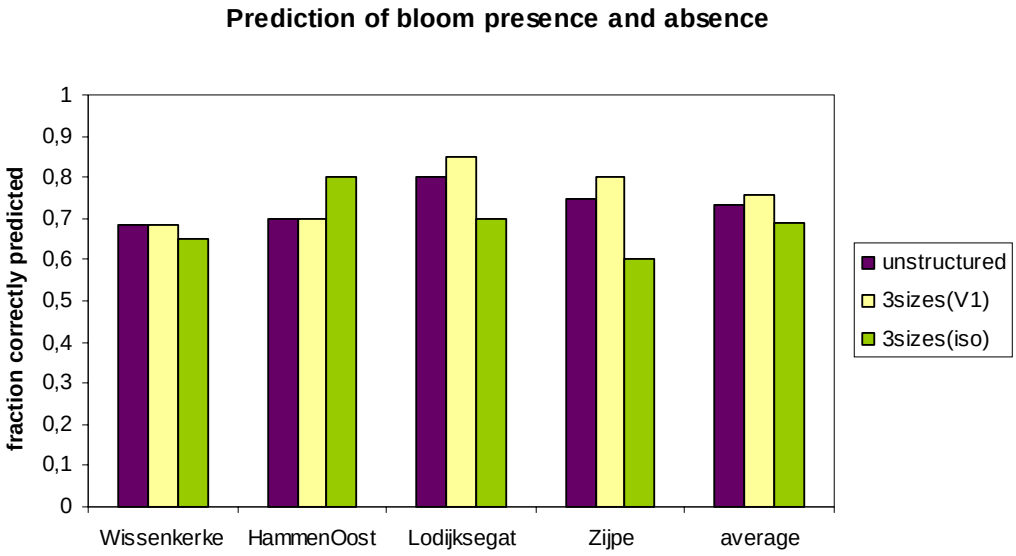


Figure 5.12. Fraction of measurements at which plankton bloom presence and absence was predicted correctly, for four locations in the Oosterschelde using the model without size-structure (purple), with V1-morphs (yellow) and with isomorphs (green).

5.7.4 Effects of size structure on carrying capacity

Including size structure in the model does not seem to have much impact on the predicted carrying capacity of the system. The size structured model variants slightly reduce primary production (Figure 5.13) as well as turnover times (Figure 5.14). Again, this is probably due to the small grazers which put relatively more grazing pressure on the system.

The apparent absence of a large effect of taking into account size-structure, indicates that the unstructured grazer module is robust with regard to these variables, and can thus well be used to predict these variables at the scale of individual compartments. This is advantageous, since the unstructured grazer is described by only three state variables for which it requires initialization. In contrast, the structured models are described by three or four state variables per size-class (depending on the model variant) and are thus more computational intensive and more difficult to initialize. Yet, these structured models may be useful when size-class specific considerations become important, such as when smaller and larger individuals are spatially separated. This is for instance the case in when studying the effect of mussel seed capture devices (MZI's) on carrying capacity.

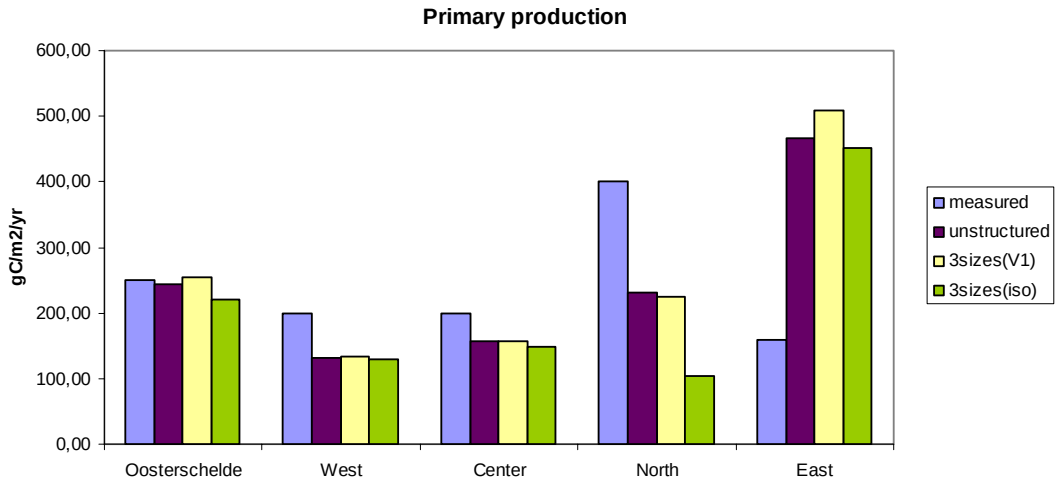


Figure 5.13. Primary production in each of the four compartments of the Oosterschelde resulting from the models, as measured (blue), without size-structure (purple), with V1-morphs (yellow) and with isomorphs (green).

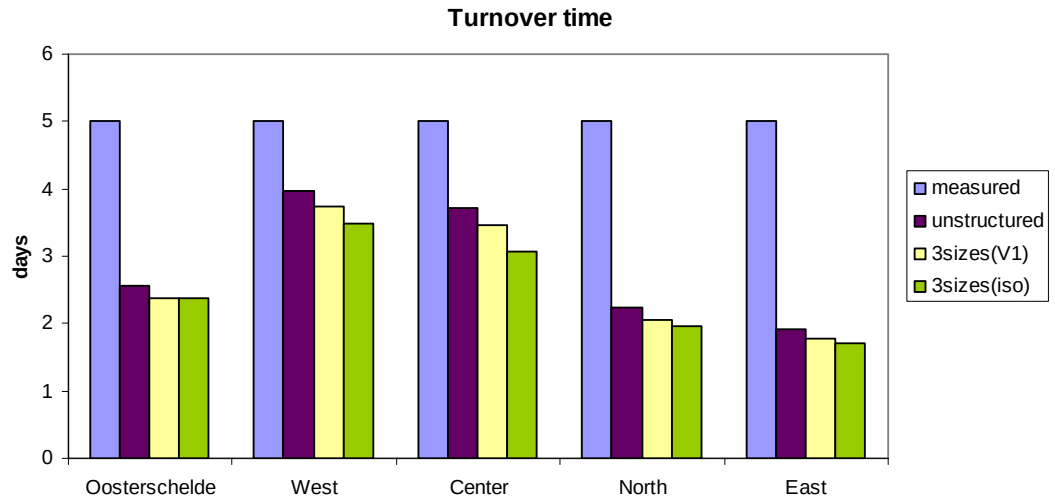


Figure 5.14. Turnover times in each of the four compartments of the Oosterschelde resulting from the models, as measured (blue), without size-structure (purple), with V1-morphs (yellow) and with isomorphs (green).

5.8 Oosterschelde GEM with grazing by different species

In order to show the sensitivity of the grazer model for different parameters, the model is parameterized for three different species (cockles, mussels, and oysters). To make the results comparable, the model was initialized totally equally in all three cases: the three species were started up with the same densities and in the same areas. These simulation runs do thus not represent realistic situations, but they do give insight in the effect that the differences in parameter values may have on the population growth and system behaviour. In addition, a simulation is done in which the three species were included simultaneously. In this case they were initialized as realistically as possible. In all these runs, only the model for V1-morphs was used.

5.8.1 Initial grazer distribution

Cockle, mussel, and oysters densities were initialized just like the density of the generalized grazer (see Figure 5.4), such that the total initial biomass and its distribution in each of the three cases was fully equal. Obviously, this does not correspond to the real situation, and habitat preference is not taken into account at all.

In the simulation with all three species, these species were initialized as realistically as possible (as is shown in Figure 5.3). Habitat preference was not included explicitly in the model, but was included implicitly by only allowing the populations to grow in areas where they were initialized.

5.8.2 Species-specific results

As expected, the grazers react differently when parameterized for different species. Cockles, being relatively small, can reach high growth rates, but also have relatively small energy reserves. As a results, their biomass is relatively variable throughout the year (Figure 5.15A). In contrast, oysters are large, and are less variable (Figure 5.15C). The growth of mussels lies in between that of oysters and cockles (Figure 5.15B), which is why the mussel-parameters were used to parameterize the generalized grazer.

When simulating the three species simultaneously, the resulting structural biomass resembles that of mussels quite closely (Figure 5.16).

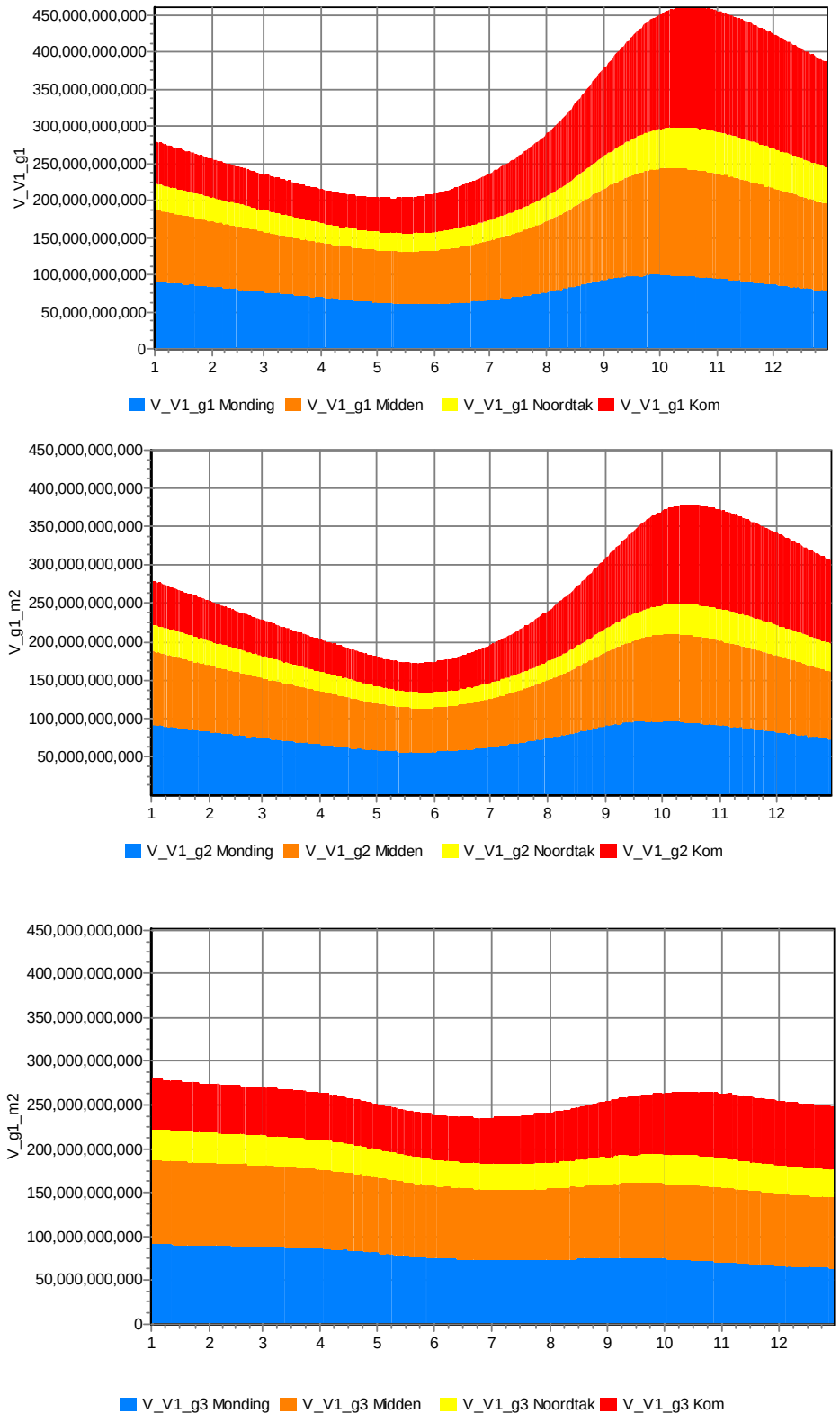


Figure 5.15. Structural grazer volume when the model is parameterized for cockles (A), mussels (B), and oysters (C) in each of the four compartments.

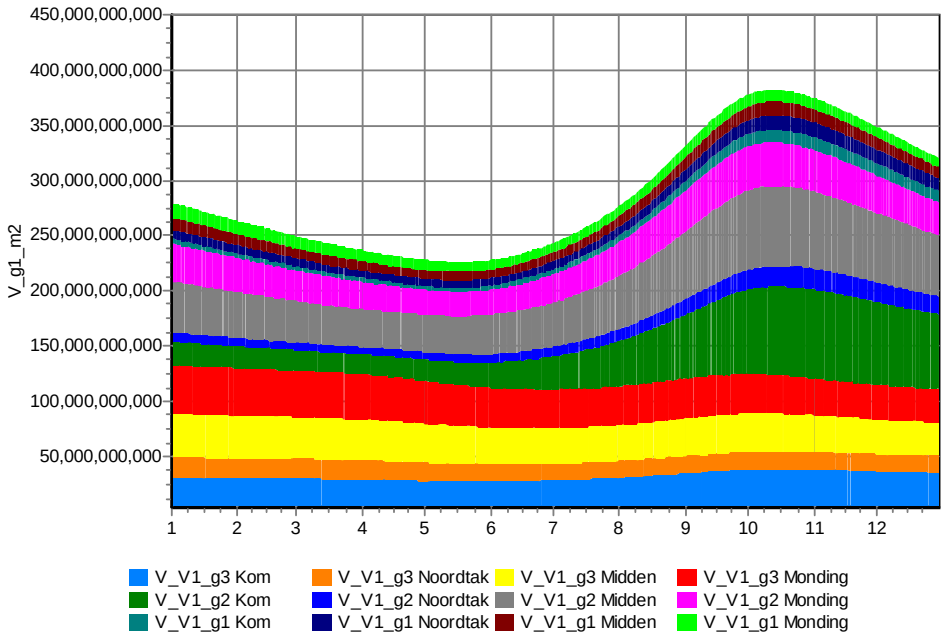


Figure 5.16. Structural grazer volume in each of the four compartments when the three species are simulated simultaneously (In the legend, ‘g1’ indicates cockles, ‘g2’ indicates mussels, and ‘g3’ indicates oysters).

5.8.3 Validation by nutrient concentrations

Resulting nutrient concentrations are shown in the Appendices A.11-A.14. Concentrations do not differ much for location Wissenkerke and HammenOost, the largest differences can be found in location Lodijkse gat and Zijpe.

On average, the oyster model performs the worst, while the mussel model performs the best. This is also the impression given by the fraction of correct predictions of algal bloom presence and absence as shown in Figure 5.17.

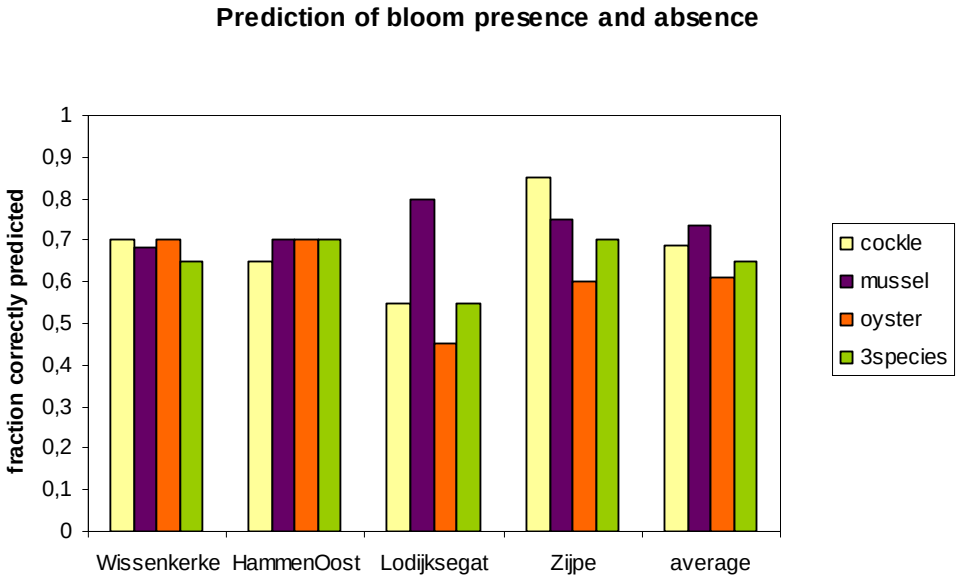


Figure 5.17. Sum of squares of chlorophyll concentrations at four locations in the Oosterschelde resulting from the model with grazers parameterized as cockles (yellow), mussels (purple), oysters (orange) and as all three species combined (green).

5.8.4 Effects of interspecific variation on carrying capacity

The oyster model leads to the smallest primary productions (Figure 5.18) and the largest turnover times (Figure 5. 19). The cockle model leads to the largest primary production rates, while the mussel model leads to the smallest turnover times. As expected, the results of the model with 3 simultaneously simulated species variation stay within the variation of the three other models, and resemble those of the mussel model.

Overall, it seems that taking into account interspecific variation into the grazing module does not affect the model behaviour much with regard to the system's carrying capacity. Interspecific variation may however become important when questions arise with regard to a specific species or location, or to interspecific competition, such as the questions regarding the invasion of *Ensis*.

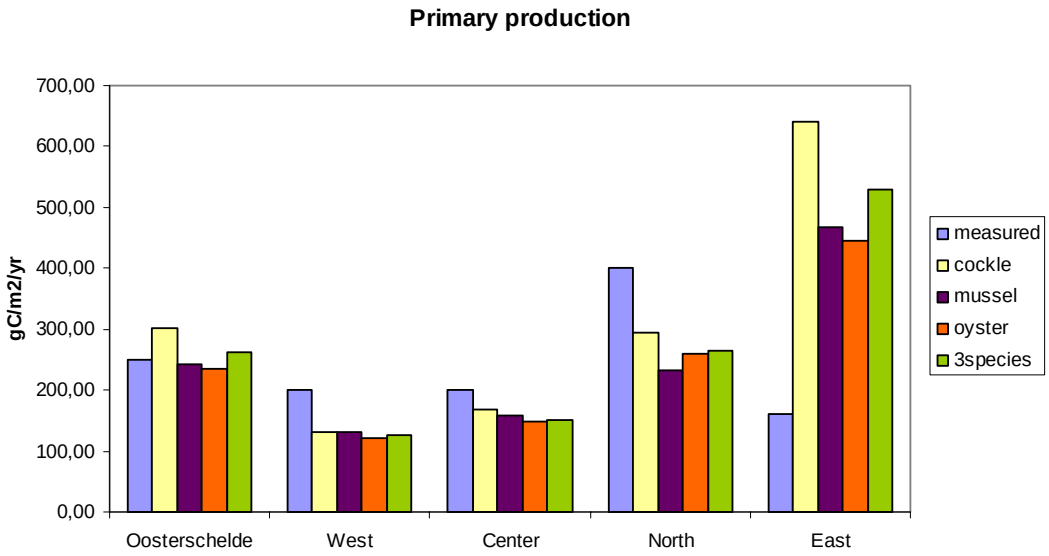


Figure 5.18. Primary production in each of the four compartments of the Oosterschelde as measured (blue), and as resulting from the model with grazers parameterized as cockles (yellow), mussels (purple), oysters (orange) and as all three species combined (green).

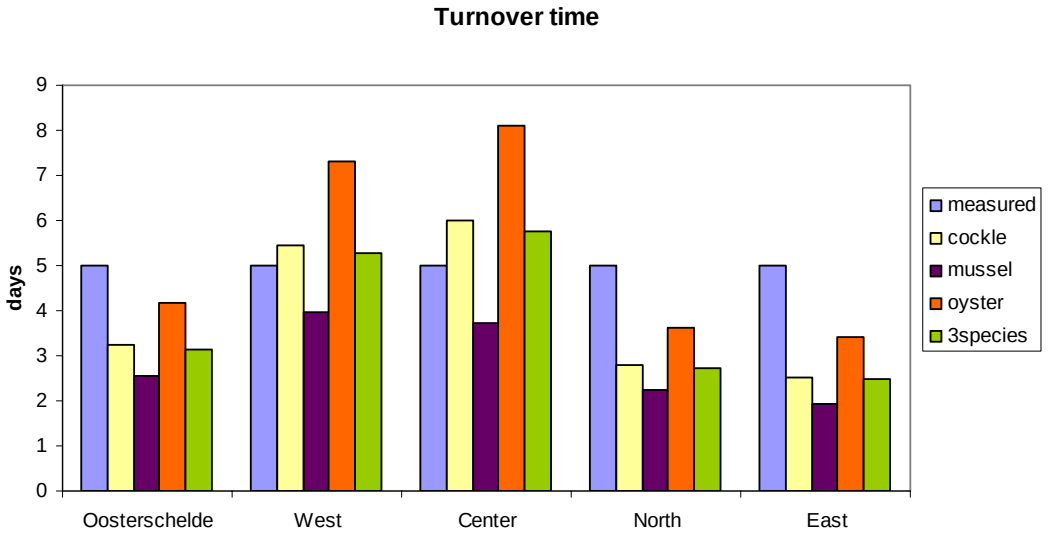


Figure 5.19. Turnover times in each of the four compartments of the Oosterschelde as measured (blue), and as resulting from the model with grazers parameterized as cockles (yellow), mussels (purple), oysters (orange) and as all three species combined (green).

5.9 Case study

In order to show how the model may contribute to assessing the effects of a management measure on the carrying capacity of a system, the grazer model is tentatively applied in a case-study considering the effects of opening the Krammer-sluices. Note that the case-study was performed in a “quick and dirty” manner, and that its results should therefore be considered with caution.

For this case-study, the Oosterschelde FLOW and GEM were adjusted to include the larger capacity of the Phillipsdam. In- and outflow of the Phillipsdam, as well as the local nutrient concentrations were based on the results of the Deltamodel corresponding to the so-called “300 m^3/s (100%) variant” (Meijers et al 2008). In this variant, a daily mean influx and outflux from the Oosterschelde to Lake Volkerak were assumed of 204 m^3/s and 268 m^3/s, respectively.

The outflowing nutrient concentrations are those calculated by the Oosterschelde model itself, while the nutrient concentration of the inflowing water are set to those calculated by the Deltamodel. The nitrogen concentrations discharged into the Oosterschelde through the Krammersluices, as used in the baseline model and in the case study are shown in Figure 5.20. (Note that the nutrient concentrations in the baseline study were based on measurements at Steenbergen). The resulting nitrogen fluxes going through the Krammersluices in the case study are shown in Figure 5.21. As can be seen in this figure, in the case study the net nitrogen flux is negative, meaning that nitrogen is flowing out of Oosterschelde.

The adjusted FLOW and GEM were run for a year. To make the results more comparable to those of the baseline case, these models were initialized equal to those in the reference case, so no additional spin up period was used. The system was thus not yet in “steady state” with the opened sluices.

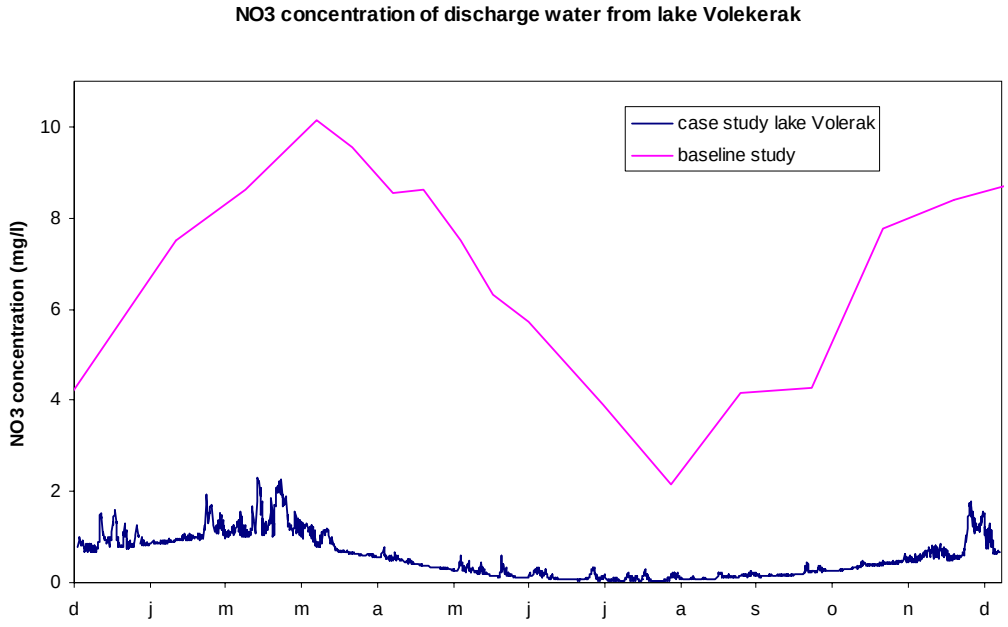


Figure 5.20. Discharged nitrogen concentrations as used in the baseline study and in the case study.

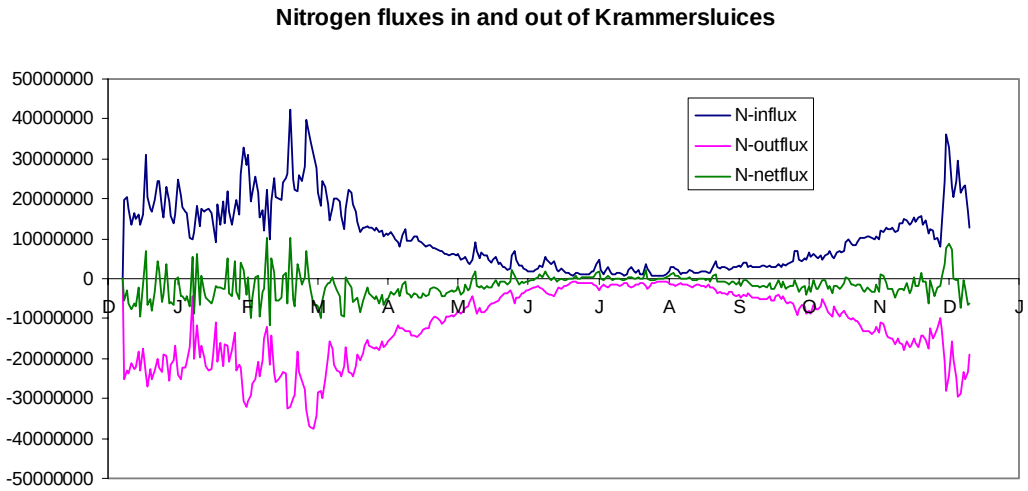


Figure 5.21. Nitrogen fluxes through the Krammersluices in the case study.

5.9.1 Initial grazer distribution

In both the default scenario as in the case study, the same initial grazer distribution was used (see Section 5.6.1).

5.9.2 Preliminary results

Figure 5.22 shows the predicted grazer structural biomass in each of the four compartments and for the total Oosterschelde. These results suggest that the grazer biomass slightly increases in all compartments when opening the Krammer sluices, and thus also in the total Oosterschelde. However, the differences between the results of the baseline run and those of the case-study are very small and fall well within the uncertainty ranges of the model.

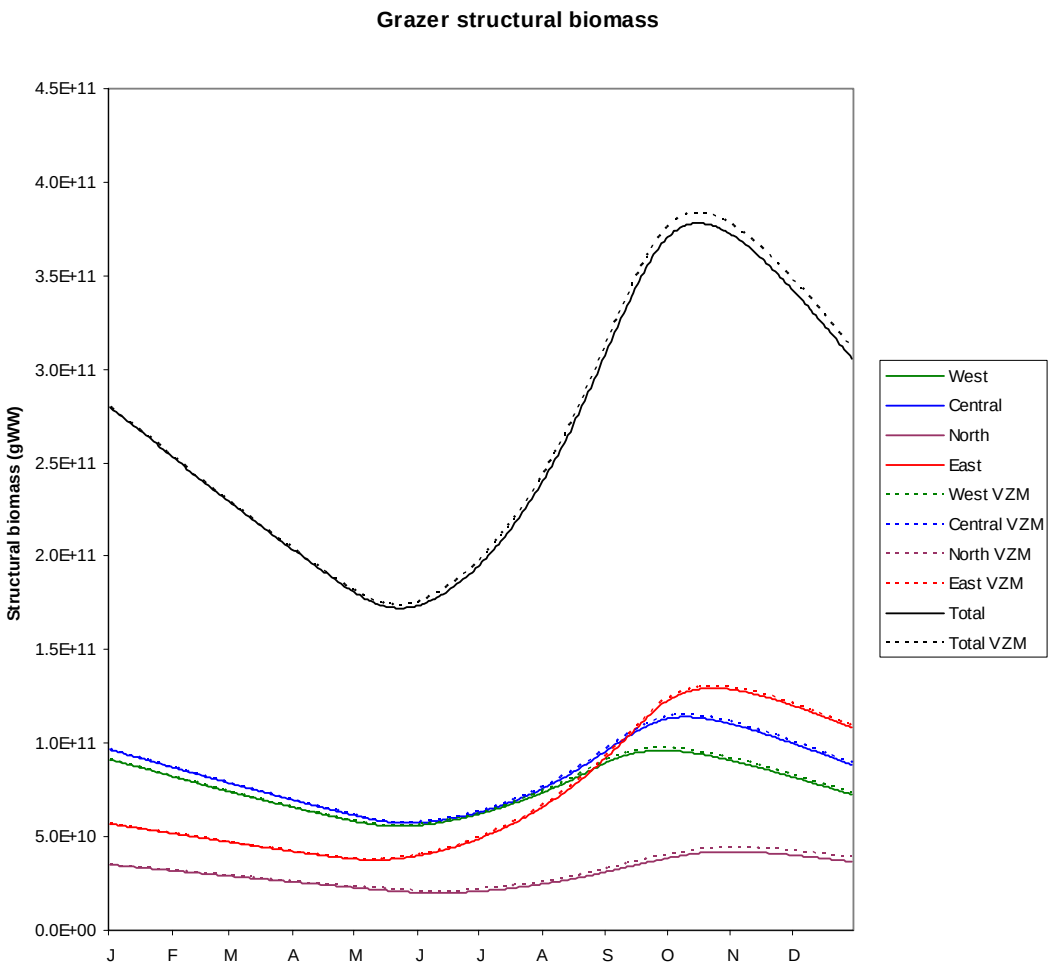


Figure 5.22 Predicted structural grazer volume

5.9.3 Preliminary effects on nutrient concentrations

Resulting nutrient concentrations are shown in Appendices A.15 – A.18. As expected, the largest differences can be found in the northern compartment (location Zijpe). Especially, the nitrogen concentration in location Zijpe is much lower than in the baseline study. Despite of the differences in nitrogen concentration, chlorophyll concentrations do not differ much between the baseline and the case study.

The salinity concentrations are only slightly affected by the opening of the Krammer sluices. Though more water flows into the Oosterschelde, this is no longer fresh water, and the resulting salinity concentrations are only slightly smaller than those in the baseline study (results not shown).

5.9.4 Preliminary effects on carrying capacity

As shown in Figure 5.21 a net nutrient flux is leaving the Oosterschelde. One could therefore expect that the carrying capacity would decrease. Yet, the primary production slightly *in*creases in all compartments (Figure 5.23), while turnover times slightly *de*crease (Figure 5.24). Although the opening of the Krammer sluices does have a considerable effect on the predicted nutrient concentrations in the northern

compartment, it thus seems that the effects on the carrying capacity and the sustainable biomass are only small. However, the differences between the results of the default run and those of the case-study are very small and fall well within the uncertainty ranges of the model.

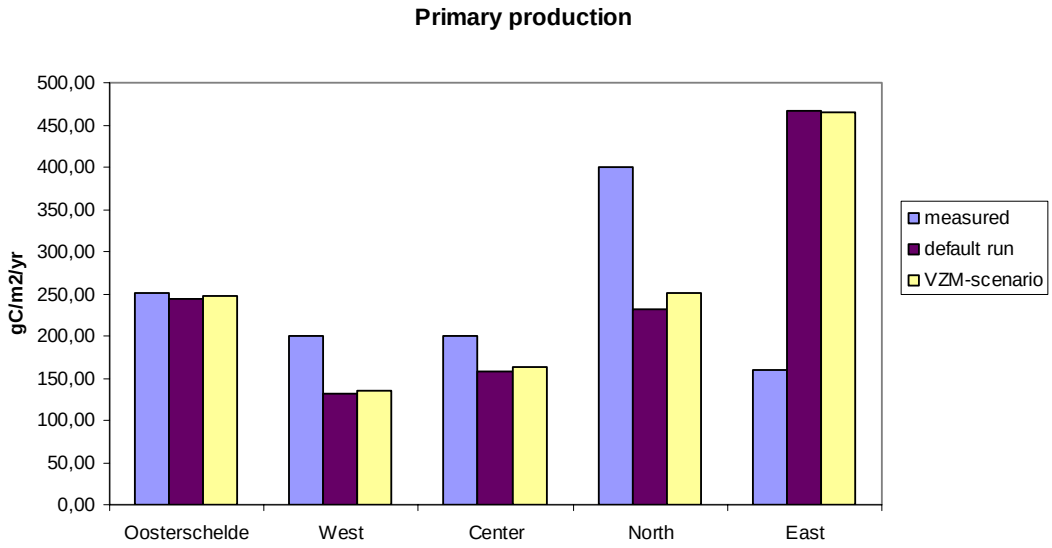


Figure 5.23. Primary production in each of the four compartments of the Oosterschelde as measured (blue), and as resulting from the model with the default settings (purple), and from the case study (yellow).

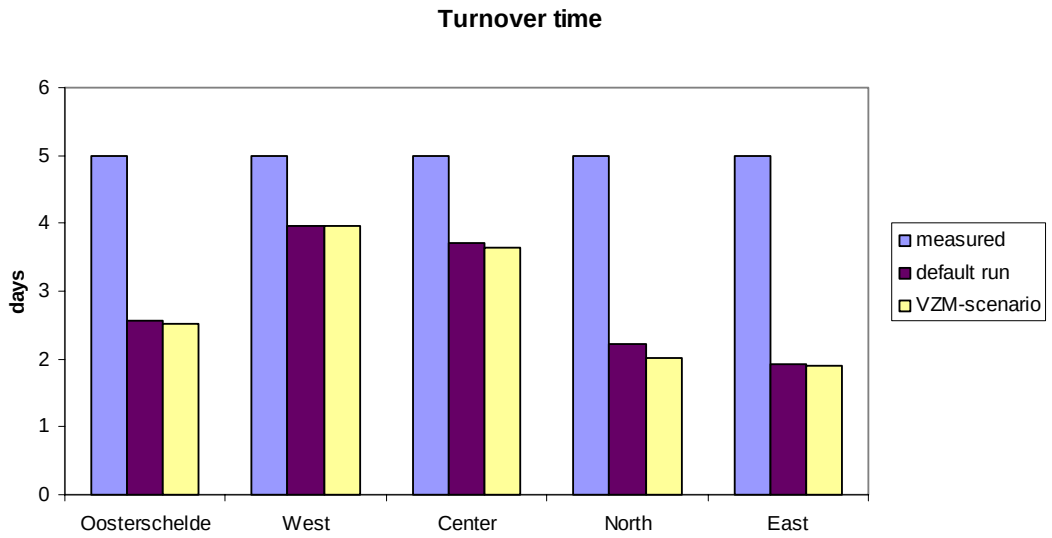


Figure 5.24. Turnover times in each of the four compartments of the Oosterschelde as measured (blue), and as resulting from the model with the default settings (purple), and from the case study (yellow).

5.10 Overall model comparison and validation

Figure 5.25 and 5.26 show a comparison of the performance of the various model variants that were studied in this report. Shown are the averaged sums of squares for primary production and turnover times (Figure 5.25), and the fraction of correctly predicted bloom presences and absences (Figure 5.26).

Including grazing greatly improves the model, especially with respect to turn over times. Furthermore, the mussel model and the oyster models perform best with regard to primary production. The mussel and the 3sizes(V1) models predict bloom presences and absences best, while the cockle and the 3-species models perform best with regard to turnover time.

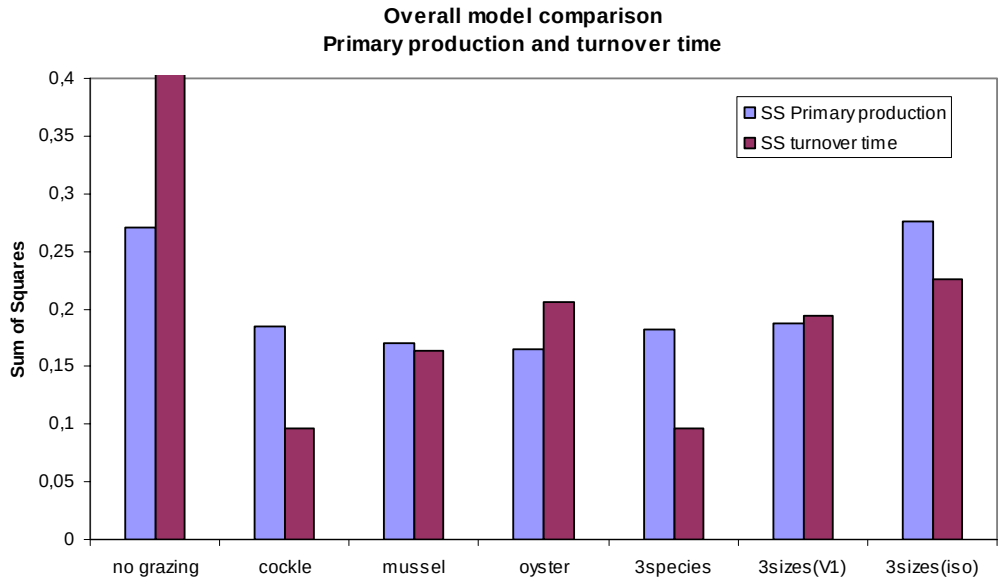


Figure 5.25. Averaged sum of squares for primary production (blue) and turnover times (red) for all model variants. Sum of squares of turnover time for the model without grazing has a value of 1.28.

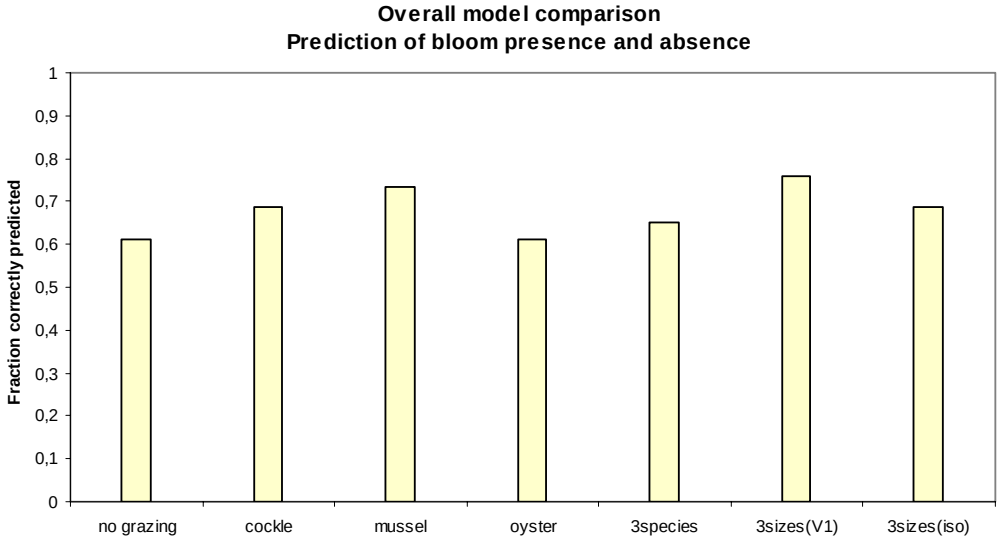


Figure 5.26. The fraction of correctly predicted bloom presences and absences for all model variants.

Another way to compare the model variants is shown in Figure 5.27. This so-called target diagram shows the normalized bias *B** and the normalized unbiased root-mean-squared difference *RMSD'**. *B** is indicative of the match between model and observations in the annual mean sense, *RMSD'** is a measure for the match between the residues of the time series after removeal of the bias (Los & Blaas, 2010). The circle with radius 1 on the target diagram corresponds to a total normalized *RMSD* equal to the standard deviation of the observations. All points outside this circle may be considered as 'poor'.

The target diagram shows that the fit of the various model variants is reasonable to good, with most points lying within the circle, and their bias close to zero. The few points lying far outside the circle (those with *B**>0,5) all correspond to location Lodijkse gat in the eastern compartment (Table 5.6), indicating that the model performance is poor for this part of the Oosterschelde.

Furthermore, the target diagram shows that in most cases the inclusion of grazing in the model reduces its bias and *RMSD'**. Also, for locations other than Lodijkse gat, the model variants score a negative *RMSD*,* implying that the standard deviation of the model is smaller than that in the observations, which is a common phenomenon in modeling.

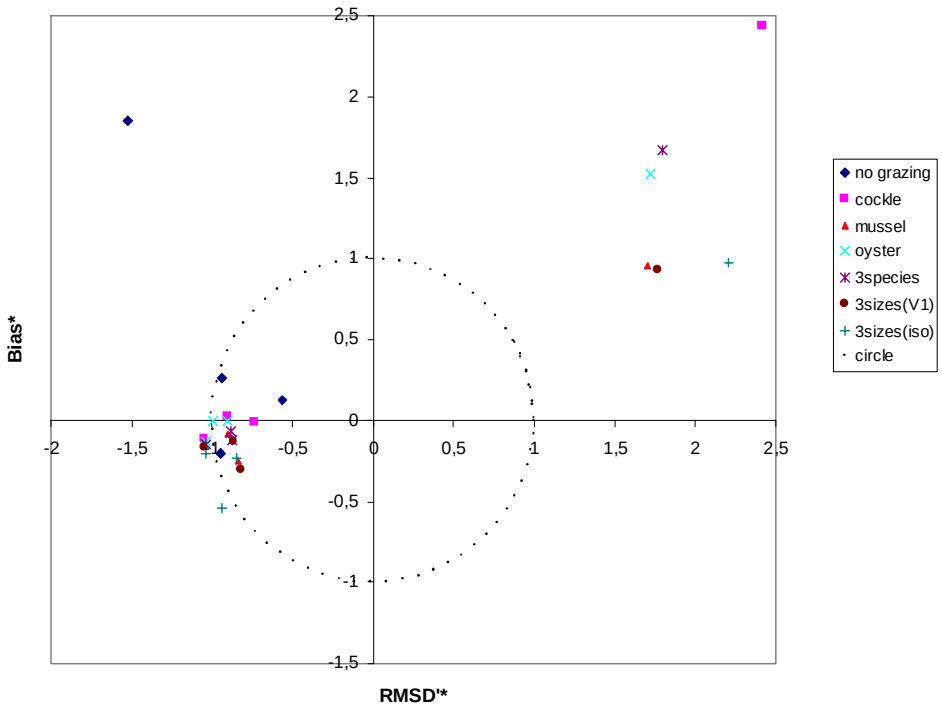


Figure 5.27. Target diagram for chlorophyll predictions per location by all model variants, showing the normalized bias *B** and the normalized unbiased root-mean-squared difference *RMSD'**. The circle with radius 1 corresponds to a total normalized *RMSD* equal to the standard deviation of the observations.

Table 5.6. The normalized bias *B** and the normalized unbiased root-mean-squared difference *RMSD'** per model version and location.

		Wissenkerke	HammenOost	Lodijksegat	Zijpe
B*	no grazing	-0,21	0,13	1,85	0,26
	cockle	-0,11	0,02	2,44	-0,02
	mussel	-0,15	-0,09	0,96	-0,25
	oyster	-0,13	-0,01	1,52	0,00
	3species	-0,14	-0,07	1,66	-0,12
	3sizes(V1)	-0,16	-0,13	0,93	-0,30
	3sizes(iso)	-0,20	-0,23	0,97	-0,54
RMSD'*	no grazing	-0,95	-0,56	-1,53	-0,94
	cockle	-1,05	-0,90	2,42	-0,74
	mussel	-1,05	-0,90	1,71	-0,84
	oyster	-1,04	-0,90	1,72	-0,99
	3species	-1,04	-0,88	1,79	-0,88
	3sizes(V1)	-1,05	-0,87	1,77	-0,82
	3sizes(iso)	-1,04	-0,85	2,21	-0,94

6 Discussion and conclusions

In this study, the carrying capacity of the Oosterschelde is calculated using an ecosystem model that includes grazers. The grazer module is developed and improved in various steps. Starting point is the model for individual cockle growth which was scaled up to a population growth model and incorporated in a fully integrated ecosystem model. The model was applied to simulate a 'generalized' grazer, as well as grazers of various species or size-classes. The results of the various model variants are compared and validated by nutrient concentrations and carrying capacity-related variables such as primary production and turnover times.

When incorporating grazers in the model, the fit of nutrient concentrations with measurements improves considerably, especially in the northern and eastern compartments. However, the fits with carrying capacity-related variables in these compartments remain poor. Probably, these poor fits are due to the fact that determining these variables requires a range of assumptions. The underlying parameters are also difficult to measure, and vary with weather and other conditions such as the presence of shellfish on the purging plots. It is therefore recommended that they are measured more frequently and on more locations.

In addition to the measurements, also the model itself may be imperfect. For instance, in the northern compartment the water is highly dominated by the inflow from the Krammer sluices, while the inflowing nutrient concentrations are not known. A problem of modeling the eastern compartment is that it is highly dominated by grazers, while grazer densities (especially of wild sublitoral oysters and the shellfish on the purging plots) are not well known.

The importance of initializing the grazers properly, was shown by adding biomass representing the shellfish on the purging plots. When including this additional biomass in the eastern compartment, the simulated phytoplankton bloom started later and this greatly improved the fit of predicted chlorophyll concentrations with measurements. The fact that the system is sensitive to such an additional biomass, indicates that the grazing pressure in the system is quite high. Yet, the predicted primary production in this compartment is much larger than measured.

Also, the importance of taking into account size structure in the grazer model was investigated. It should be noted that comparing the size-structured model with the unstructured model was quite difficult, as the end results highly depend on the chosen initial biomass distribution over the various size classes, the initial size of the shellfish in each of these classes, and their mortality rates.

So far, however, results show that taking into account size structure in the grazer model does not have much impact on the predicted primary production and turnover times. This indicates that the grazer model is robust with regard to these variables, and that the simpler unstructured model can thus be well used to predict these variables at the scale of individual compartments. This is advantageous, since the unstructured grazer is described by only three state variables for which it requires initialization. In contrast, the structured models are described by three or four state variables per size-class

(depending on the model variant) and are thus more computational intensive and much more difficult to initialize. Yet, these structured models may be useful when size-class specific considerations become important, such as when smaller and larger individuals are spatially separated. This is for instance the case in when studying the effect of mussel seed capture devices (MZI's) on carrying capacity.

Furthermore, the sensitivity of the grazer model for its parameter values was analysed by parameterizing the model for three different species. The goal of this exercise was not to simulate realistic situations, but to give insight in the effect that the differences in parameter calues may have on the population growth and system behaviour. In addition, also a simulation run was doen in which the three species were modeled simulateously. In this case the densitiy distribution of each of the species was done as realistically as possible.

Taking into account this interspecific variation into the grazing module did not seem to affect the model behaviour much. Again, this shows the robustness of the simple unstructured grazer module. Interspecific variation may however become important when questions arise with regard to a specific species or to interspecific competition, such as the questions regarding the invasion of *Ensis*.

Although model results are not yet perfect, the model was applied to study the effects of opening of the Krammer sluices. Preliminary results suggest that the carrying capacity of the system may increase slightly in this case, but the difference is small and falls well into the uncertainty range of the model.

6.1 Recommendations

(1) including pseudofaeces production

A large improvement regarding the feedback of grazers on their environment can be made by separating the model processes for pseudofaeces production and those for faeces production, since faeces and pseudofaeces are thought to have a different composition and mineralization rate. This will also enable a dynamic modeling of the effect of filtration on the suspended particulate matter.

(2) More and improved measurements on carrying capacity related parameters

More frequent measurements of carrying capacity related variables such as primary production and turnover times are necessary to better validate or calibrate the grazer model.

(3) Model results in eastern and northern compartment

Model results in the eastern and northern compartment could be improved by moderating the initial grazer biomass and inflowing nutrient concentrations, respectively.

(4) Adjustments to model MZI's

To be able to model MZI's, the model should be run in 3D mode, to take into account the fact that the MZI's are placed in the water column, and not on the bottom. For this, the grazer model has to be adjusted as well to allow grazer growth in the water column.

7 Reference list

Bacher, C., Gangnery, A. 2006 Use of dynamic energy budget and individual based models to simulate the dynamics of cultivated oyster populations. *J. Sea Res.* **56**,140–155.

Beukema, J. J., Dekker, R. 2005. Decline of recruitment success in cockles and other bivalves in the Wadden Sea: possible role of climate change, predation on post-larvae and fisheries. Marine Ecology-Progress Series 287: 149-167.

Blauw, A. N., van Beek, J.K.L., Troost, T.A., Desmit, X., Zijl, F., Los, F.J. 2007. Keyzones, ecosystem scale modelling: WP4: deliverable 12. WL|Delft Hydraulics z3557.

Geurts van Kessel, A.J.M., Kater, B.J. , and Prins, T.C. 2003. Veranderende draagkracht van de Oosterschelde voor kokkels. Rapport RIKZ/2003.043, RIVO rapport C062/03.

Kamermans P., Kestloo, J., Baars, D. 2003. Deelproject H2: Evaulatie van de geschatte omvang en ligging van kokkelbestanden in de Waddenzee, Oosterschelde en Westerschelde. RIVO rapport C054/03.

Kooijman, S.A.L.M., 2000. Dynamic Energy and Mass Budgets in Biological systems. Cambridge University Press, 2nd edition.

Los F. J. and M. Blaas. 2008. Complexity, accuracy and practical applicatbility of different biogeochemical model versions. Journal of Marine Systems AMEMR 2008 Special Issue.

Meijers E.M., Groot, S., Haasnoot, M., van Wesenbeeck, B.K., de Vries, I. 2008. Waterkwaliteit en ecotopen in een zout Volkerak-Zoommeer. Deltares, Q4448.

Meijers E.M., Groot, S. 2007. Deltamodel – hulpmiddel ter ondersteuning van het beheer en beleid van de zuidwestelijke Delta. WL|Delft Hydraulics Q4435

Pouvreau, S., Bourles, Y., Lefevre, S., Gangnery, A., Alunno-Bruscia, M. 2006 Application of a dynamic energy budget model to the Pacific oyster, *Crassostrea gigas*, reared under various environmental conditions. *J. Sea Res.* **56**, 156–167.

Rosland, R., Strand, Ø., Alunno-Bruscia, M., Bacher, C., Strohmeier, T. 2009 Applying Dynamic Energy Budget (DEB) theory to simulate growth and bio-energetics of blue mussels under low seston conditions. *J. Sea Res.* **62**, 49–61.

Van der Veer H.W., Cardoso, J.F.M.F., van der Meer, J. 2006. In: thesis Cardoso, Chapter 7.

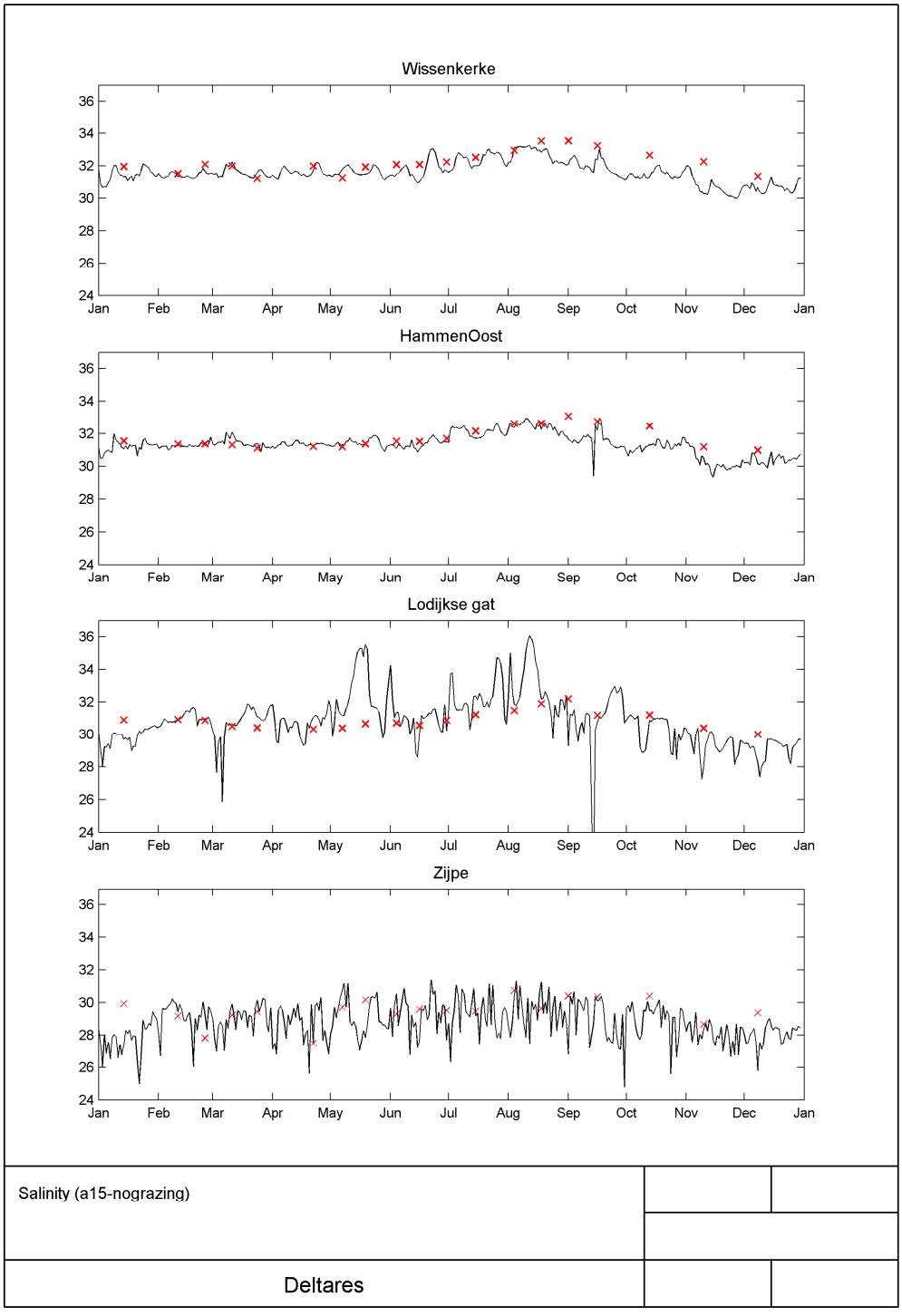
Wetsteyn, L.P.M.J.; Duin, R.N.M. *et al.* (2003). Exploration capacity Oosterschelde; study towards changes and trends in the Oosterschelde in the period 1990 till 2000 [Verkenning draagkracht Oosterschelde; onderzoek naar veranderingen en trends in de Oosterschelde in de periode 1990 t/m 2000]. *Rapport RIKZ*, 2003.049. RIKZ/Rijksinstituut voor Kunst en Zee/Ministerie van Verkeer en Waterstaat, Directoraat-Generaal Rijkswaterstaat: the Netherlands. 106 pp.

Wijsman, J.W.M. Een generiek model voor hogere trofie Niveaus. 2004. WL|Delft Hydraulics Z3515.

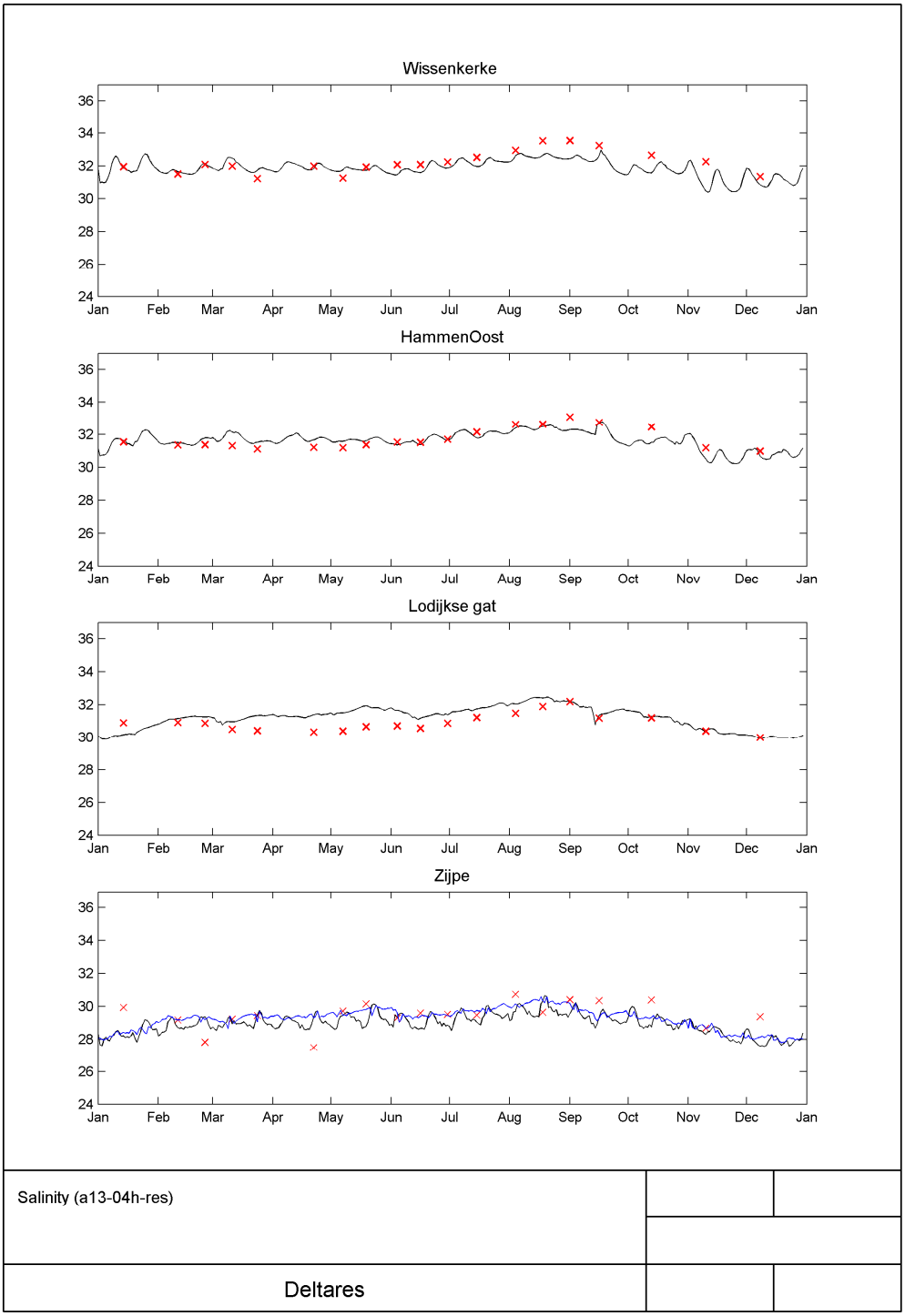
Wolfshaar, van de, K. 2007. A biomass-based stage-structured model for higher trophic levels; A grazer module for *Mytilus edulis*. WL|Delft Hydraulics.

A Figures

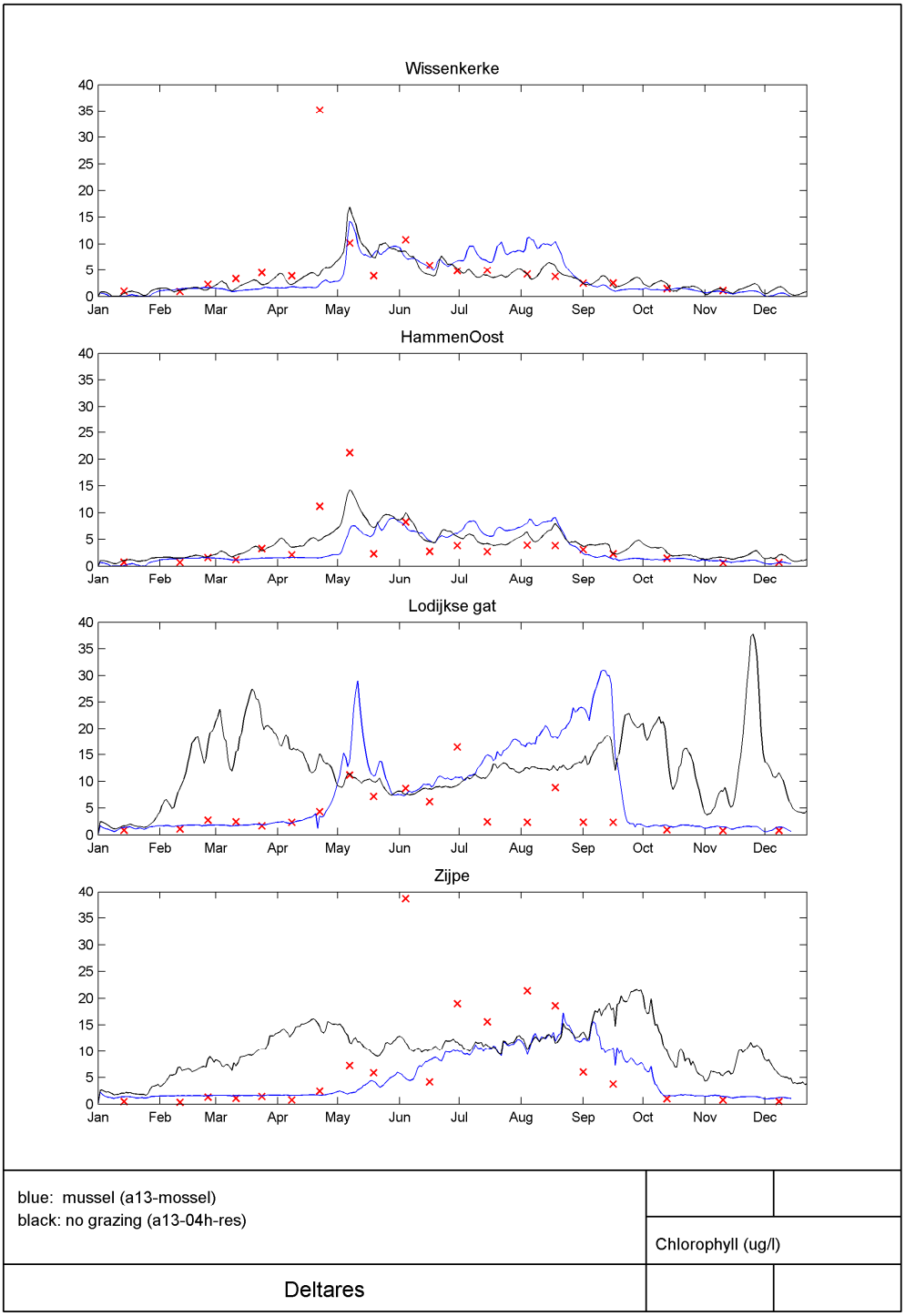
A.1 Salinity levels (ppt) simulated with the 3D FLOW model (black curve) and measured (red crosses) salinity levels in the surface layer at four locations in the Oosterschelde (Wissenkerke, HammenOost, Lodijksegat, and Zijpe).



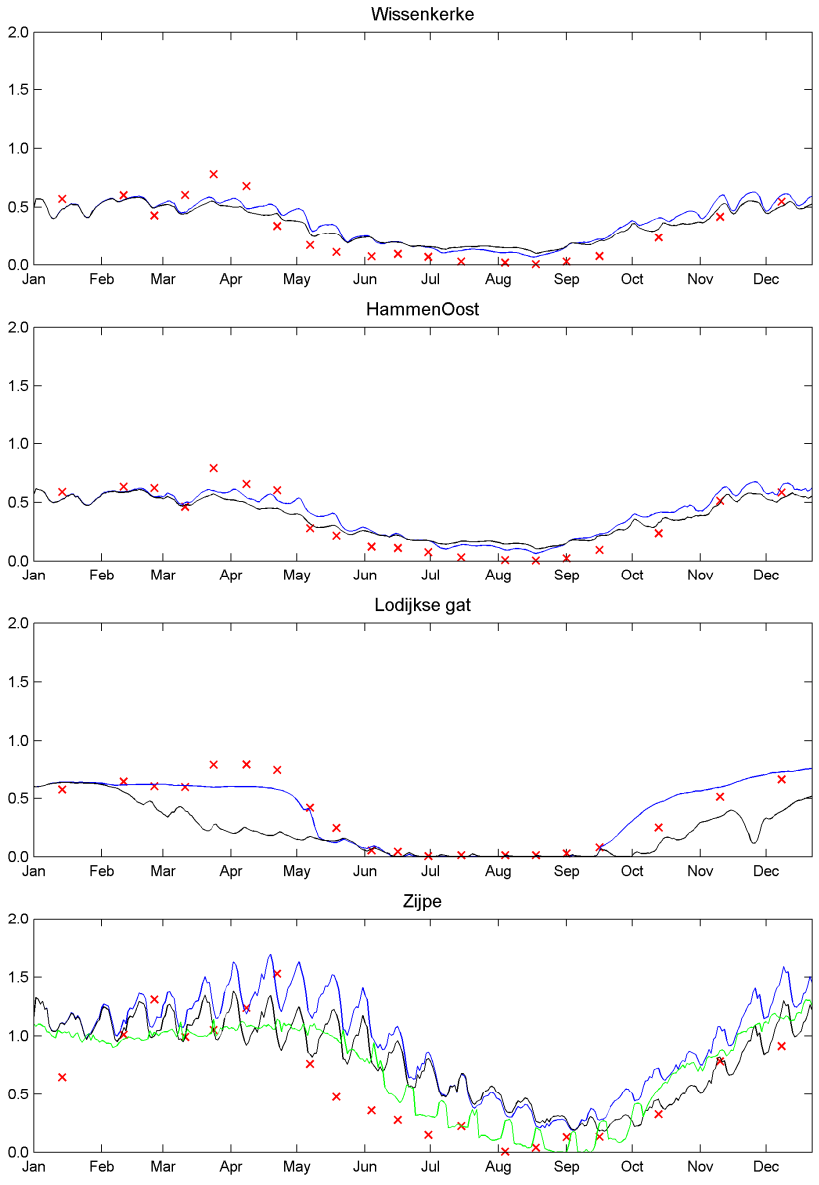
A.2 Salinity levels (ppt) simulated with the 3D FLOW model (black curves) converted to 2D in four locations in the Oosterschelde (Wissenkerke, HammenOost, Lodijksegat, and Zijpe). Red crosses indicate measured salinity levels. The blue curve corresponds to a location close to Zijpe.



A.3 Chlorophyll: simulated chlorophyll concentrations of the 2D GEM with (blue curve) and without grazing (black curve) at four locations in the Oosterschelde. Red crosses are measured concentrations.

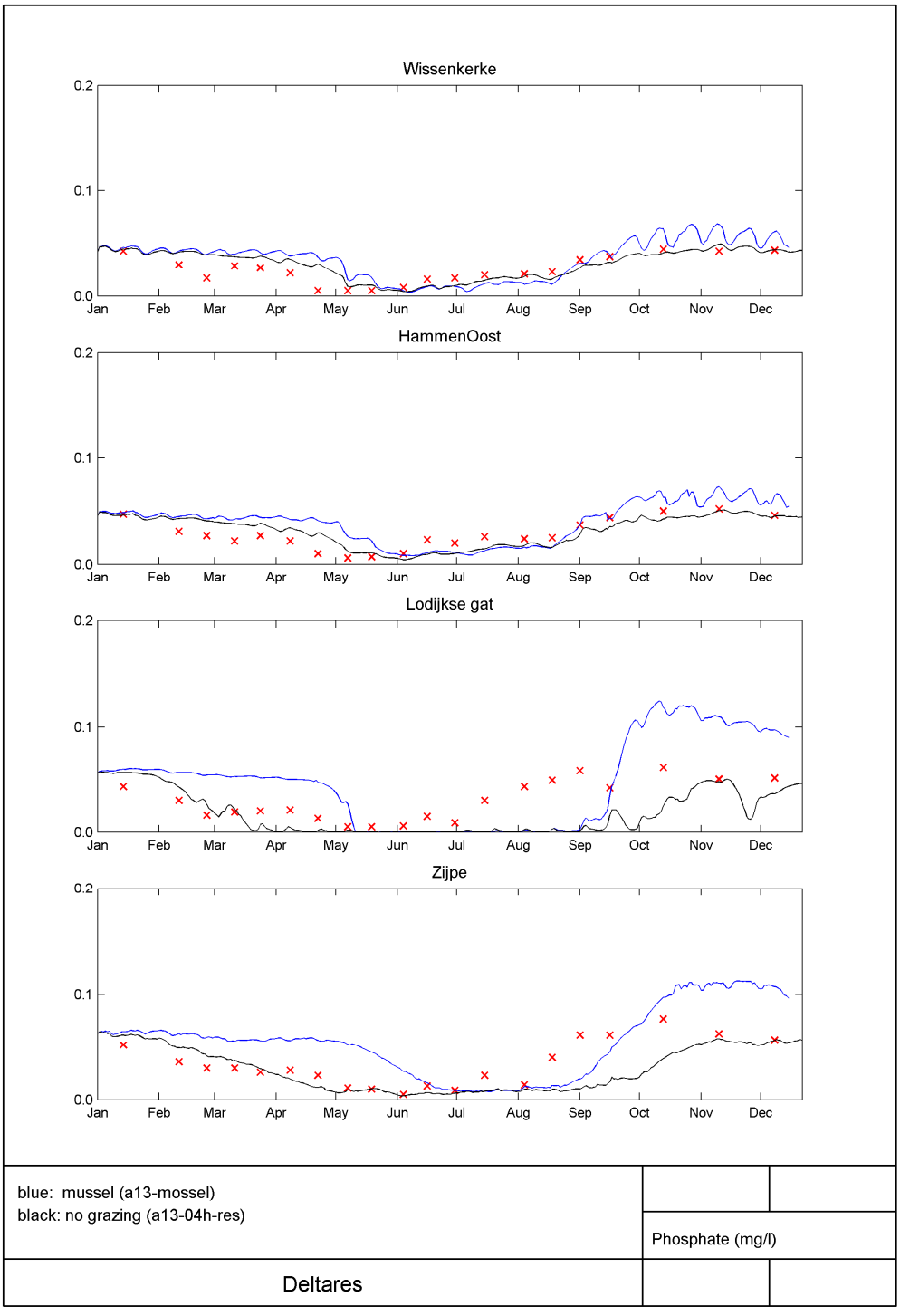


A.4 Nitrate: simulated NO3 concentrations of the 2D GEM with (blue curve) and without grazing (black curve) at four locations in the Oosterschelde. Red crosses are measured concentrations. The green curve corresponds to the model with grazing for a location close to Zijpe.

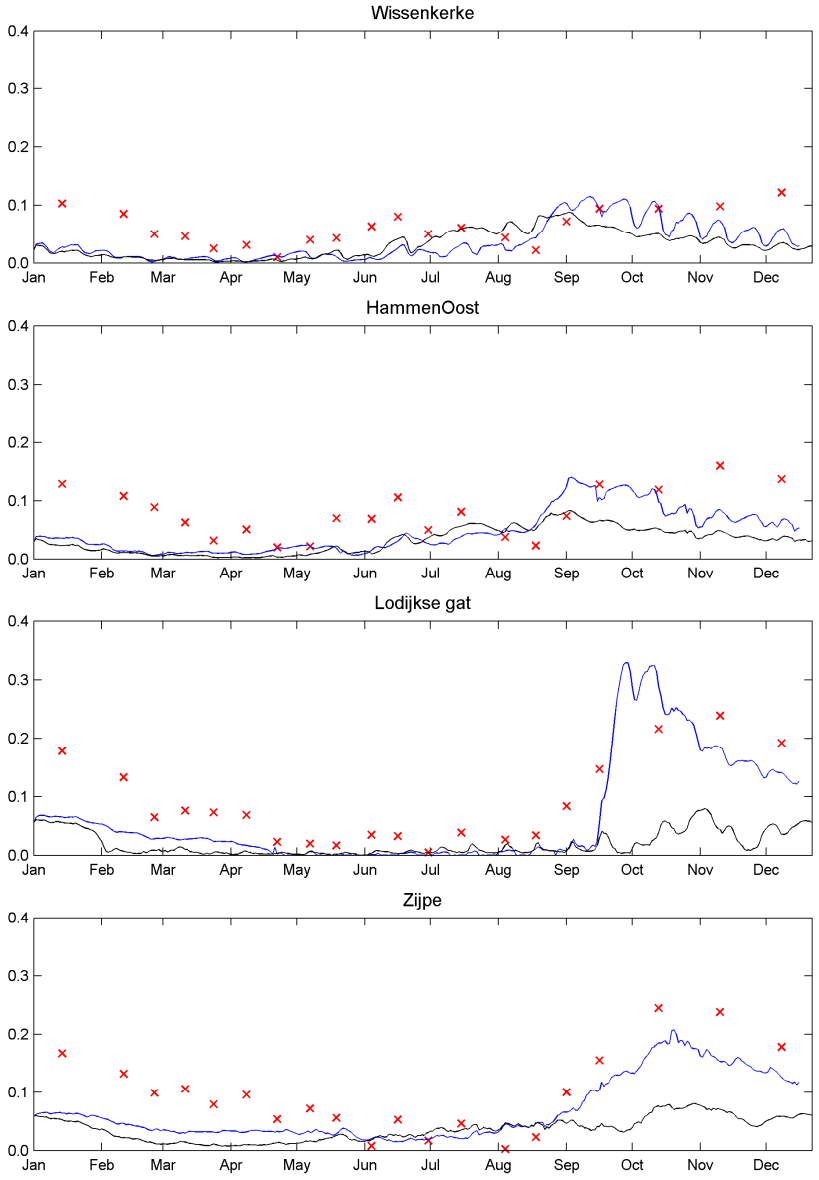


blue: mussel (a13-mossel) black: no grazing (a13-04h-res)		
	Nitrate (mg/l)	
Deltares		

A.5 Phosphate: simulated PO4 concentrations of the 2D GEM with (blue curve) and without grazing (black curve) at four locations in the Oosterschelde. Red crosses are measured concentrations.



A.6 Ammonium: simulated NH4 concentrations of the 2D GEM with (blue curve) and without grazing (black curve) at four locations in the Oosterschelde. Red crosses are measured concentrations.

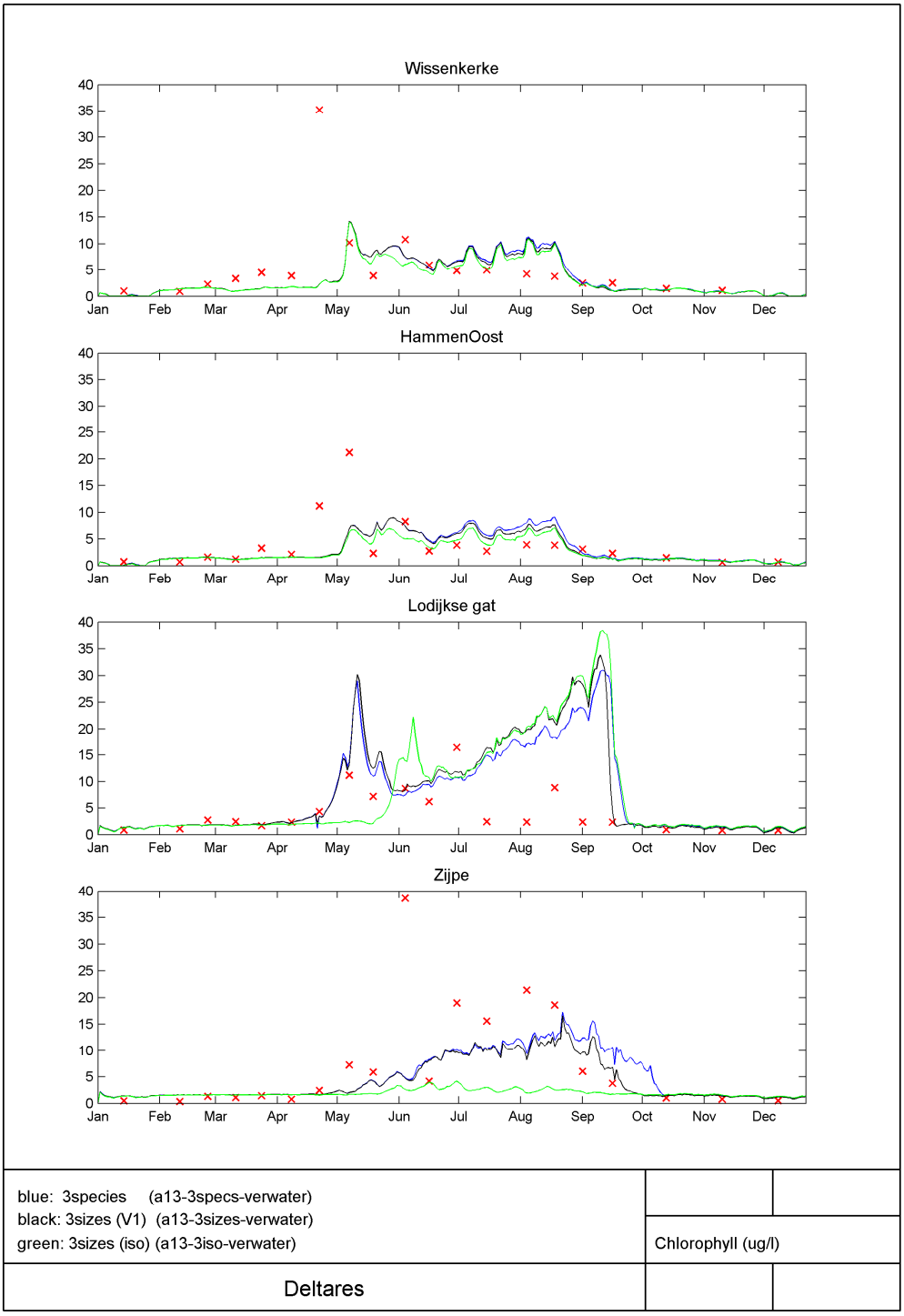


blue: mussel (a13-mossel)
black: no grazing (a13-04h-res)

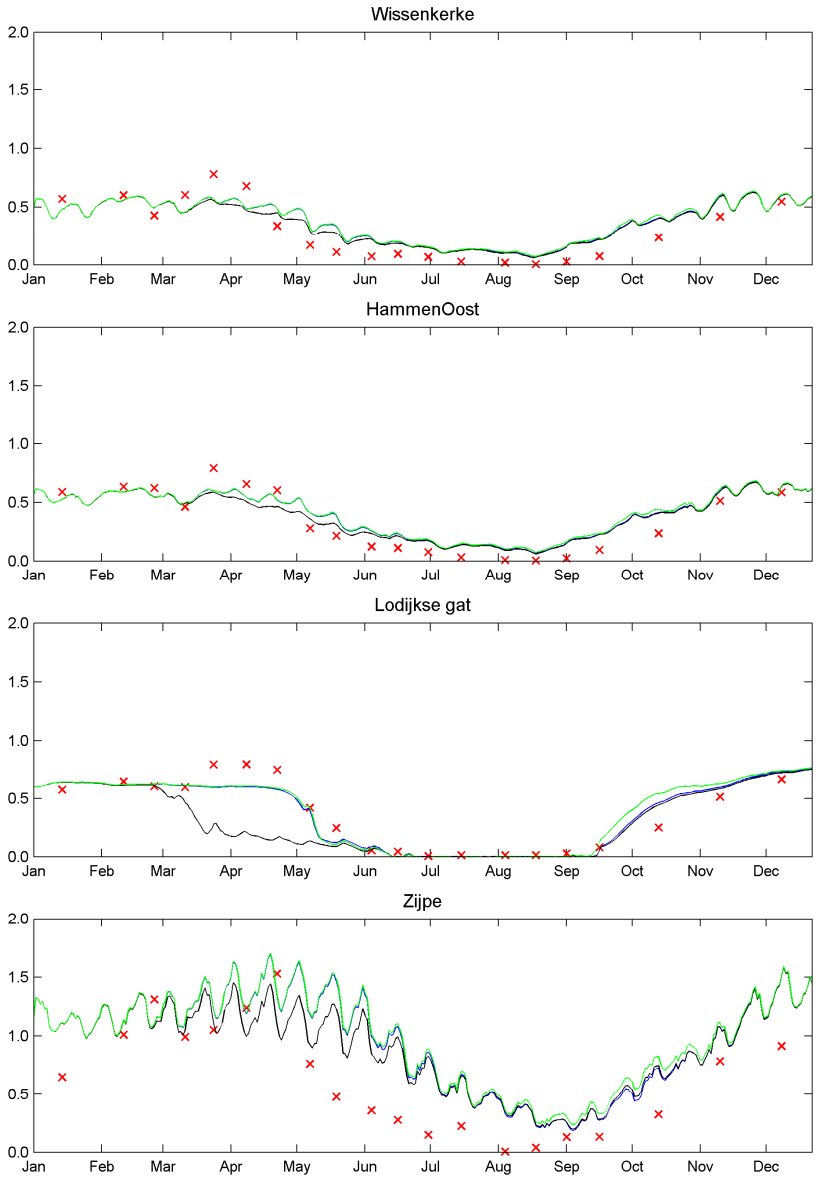
Ammonium (mg/l)

Deltares

A.7 Chlorophyll: simulated chlorophyll concentrations of the 2D GEM with unstructured grazing (blue curves) and size-structured grazing (V1-morphs: black curves, iso-morphs: green curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.

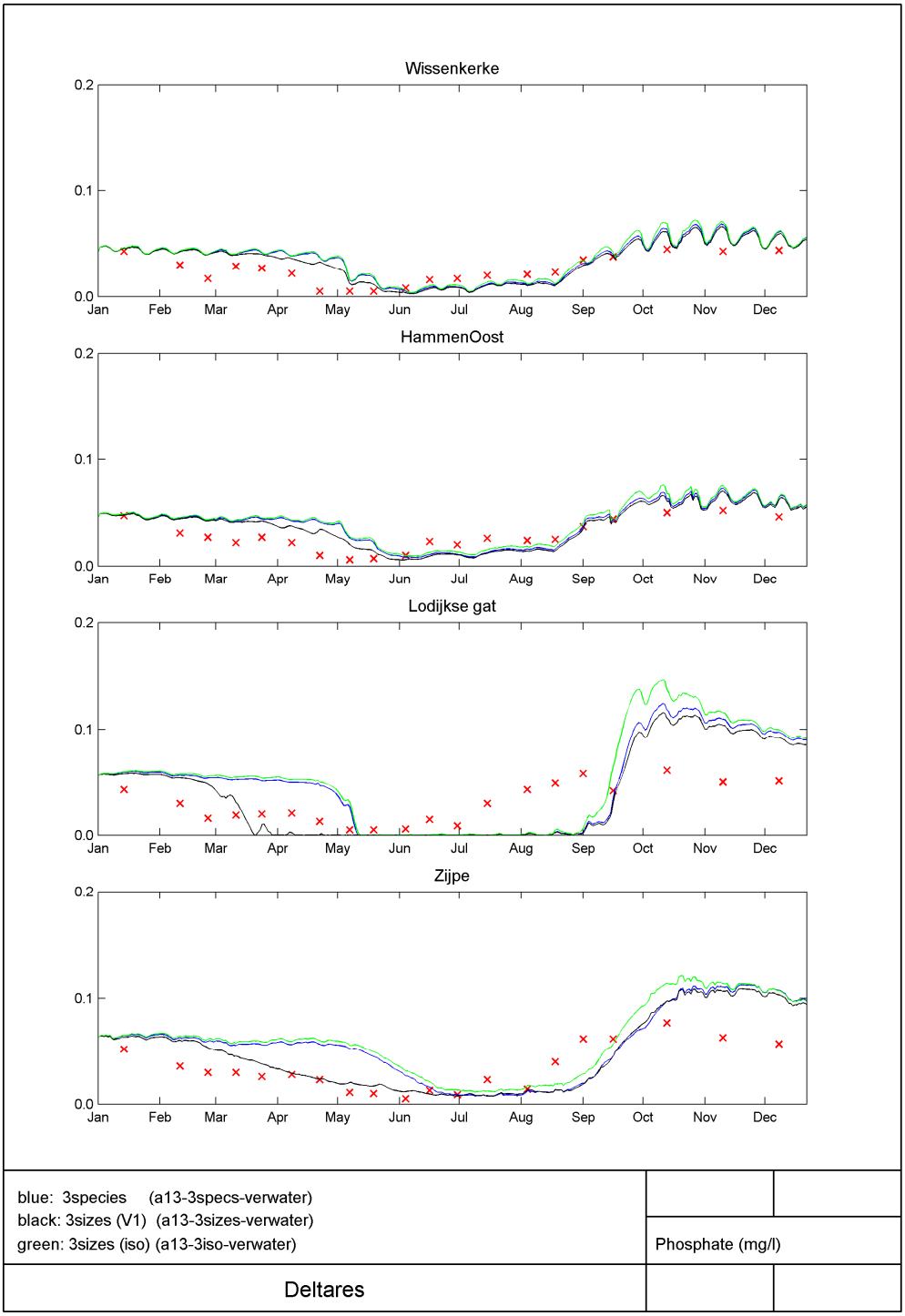


A.8 Nitrate: simulated NO3 concentrations of the 2D GEM with unstructured grazing (blue curves) and size-structured grazing (V1-morphs: black curves, iso-morphs: green curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.

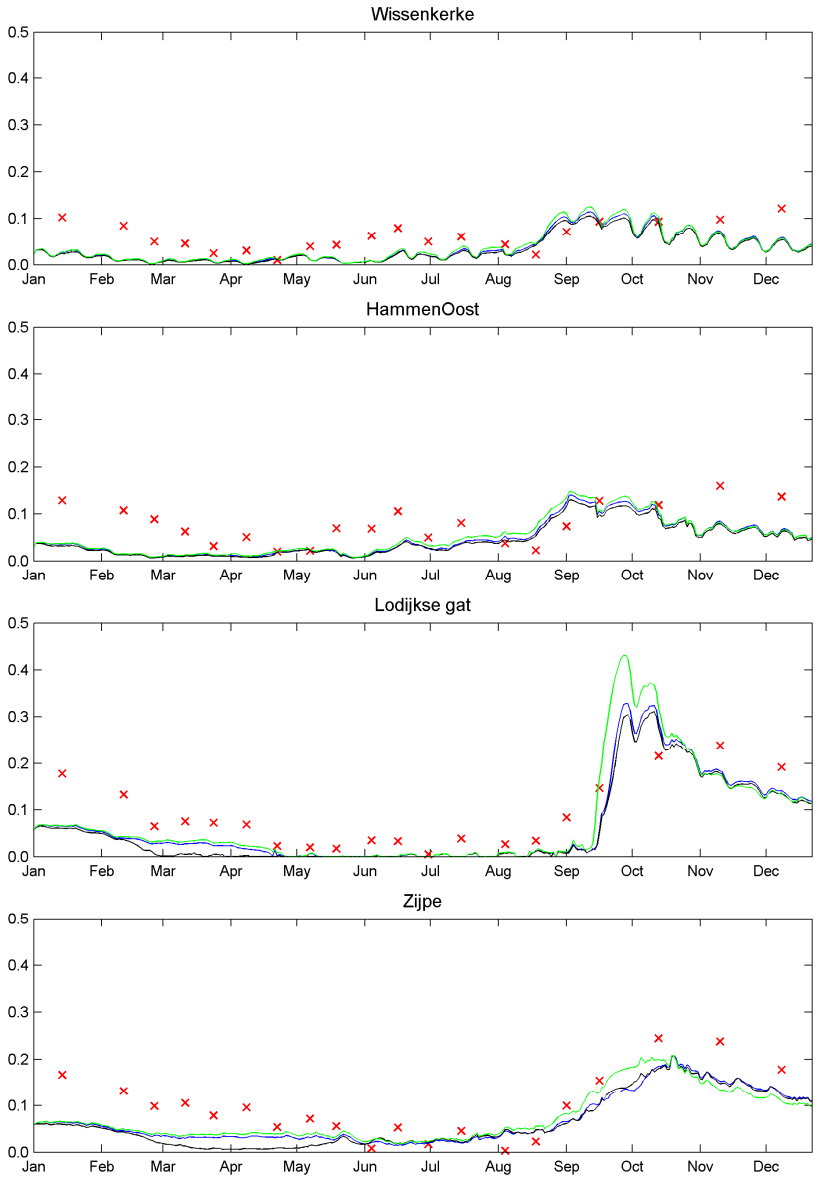


blue: 3species (a13-3specs-verwater) black: 3sizes (V1) (a13-3sizes-verwater) green: 3sizes (iso) (a13-3iso-verwater)		
	Nitrate (mg/l)	
Deltares		

A.9 Phosphate: simulated PO4 of the 2D GEM with unstructured grazing (blue curves) and size-structured grazing (V1-morphs: black curves, iso-morphs: green curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.

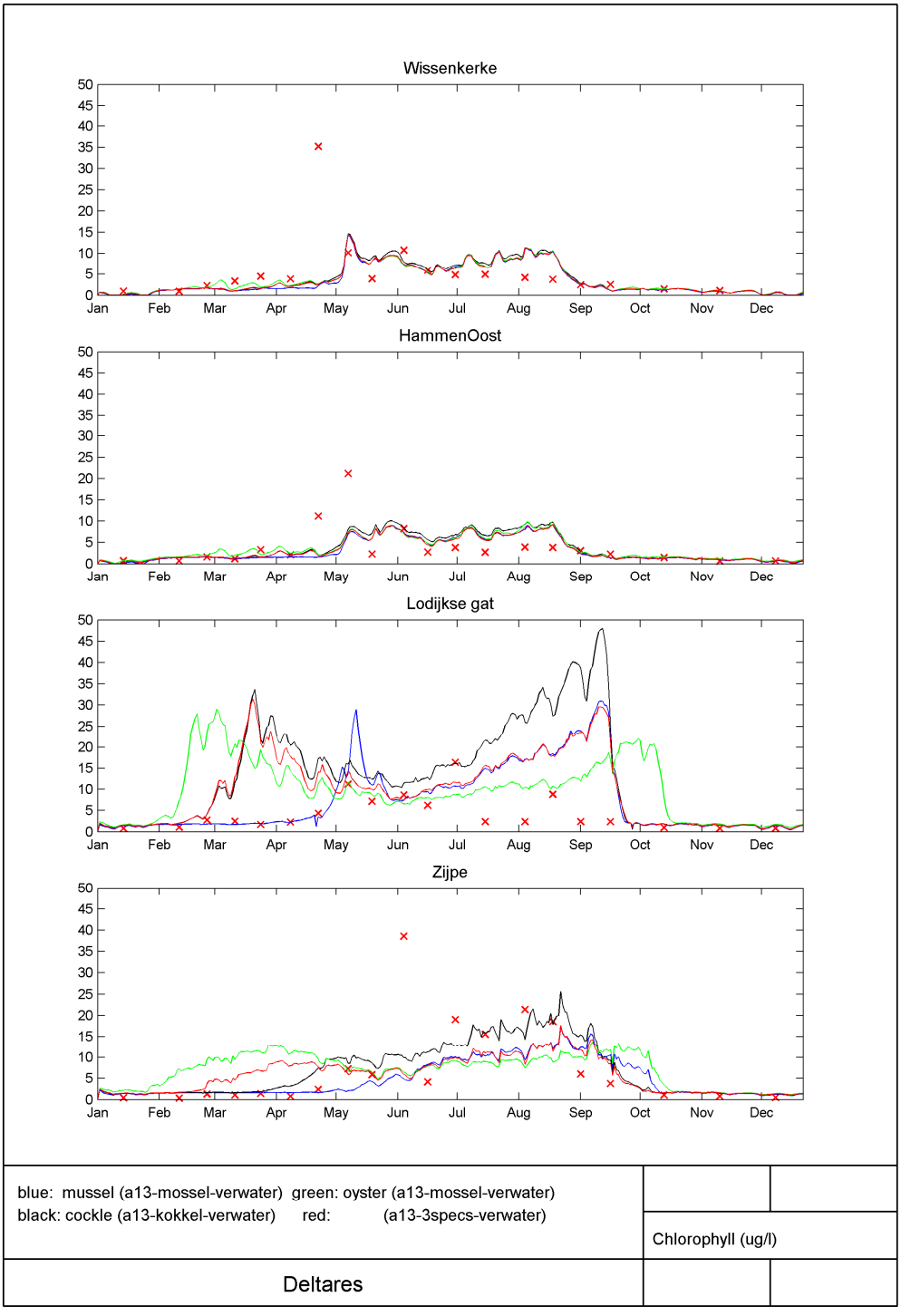


A.10 Ammonium: simulated NH4 concentrations of the 2D GEM with unstructured grazing (blue curves) and size-structured grazing (V1-morphs: black curves, iso-morphs: green curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.

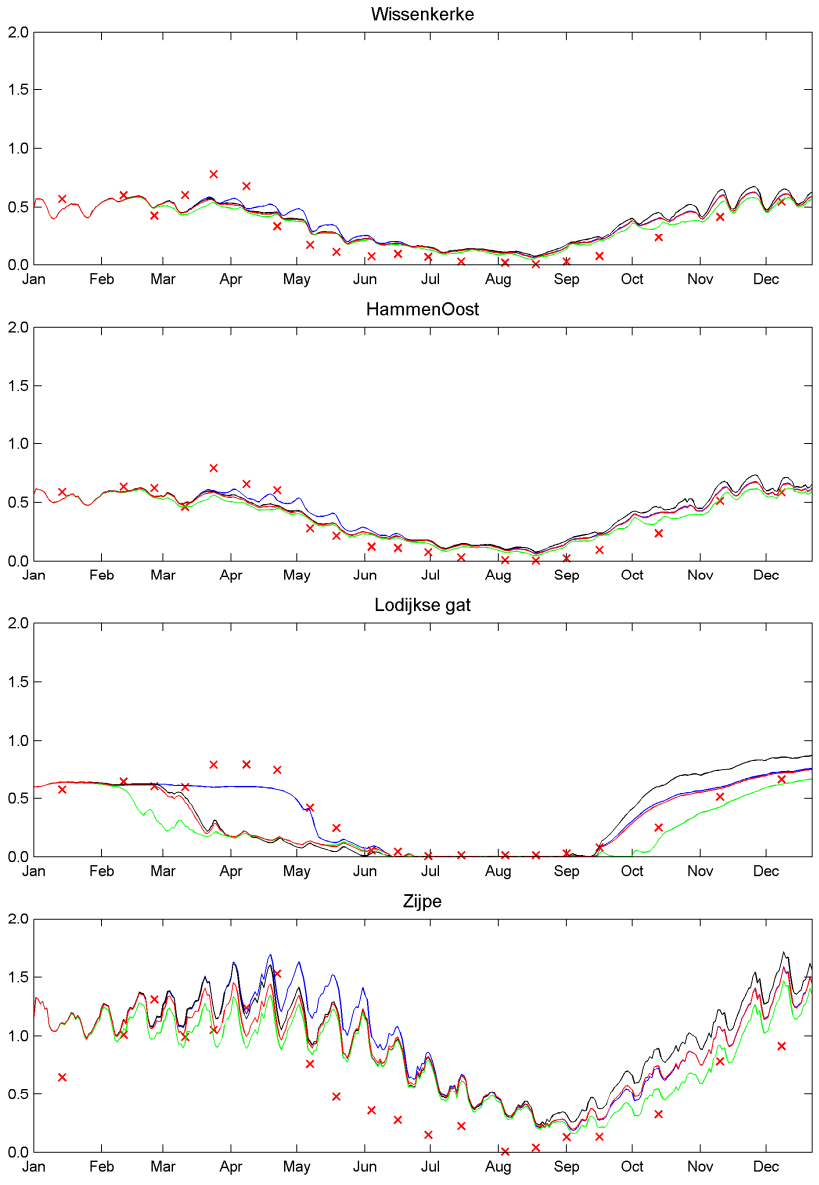


blue: 3species (a13-3specs-verwater) black: 3sizes (V1) (a13-3sizes-verwater) green: 3sizes (iso) (a13-3iso-verwater)		
	Ammonium (mg/l)	
Deltares		

A.11 Chlorophyll: simulated chlorophyll of the 2D GEM with grazing of mussels (blue curves), cockles (black curves), oysters (green curves) and the three species simultaneously (red curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.

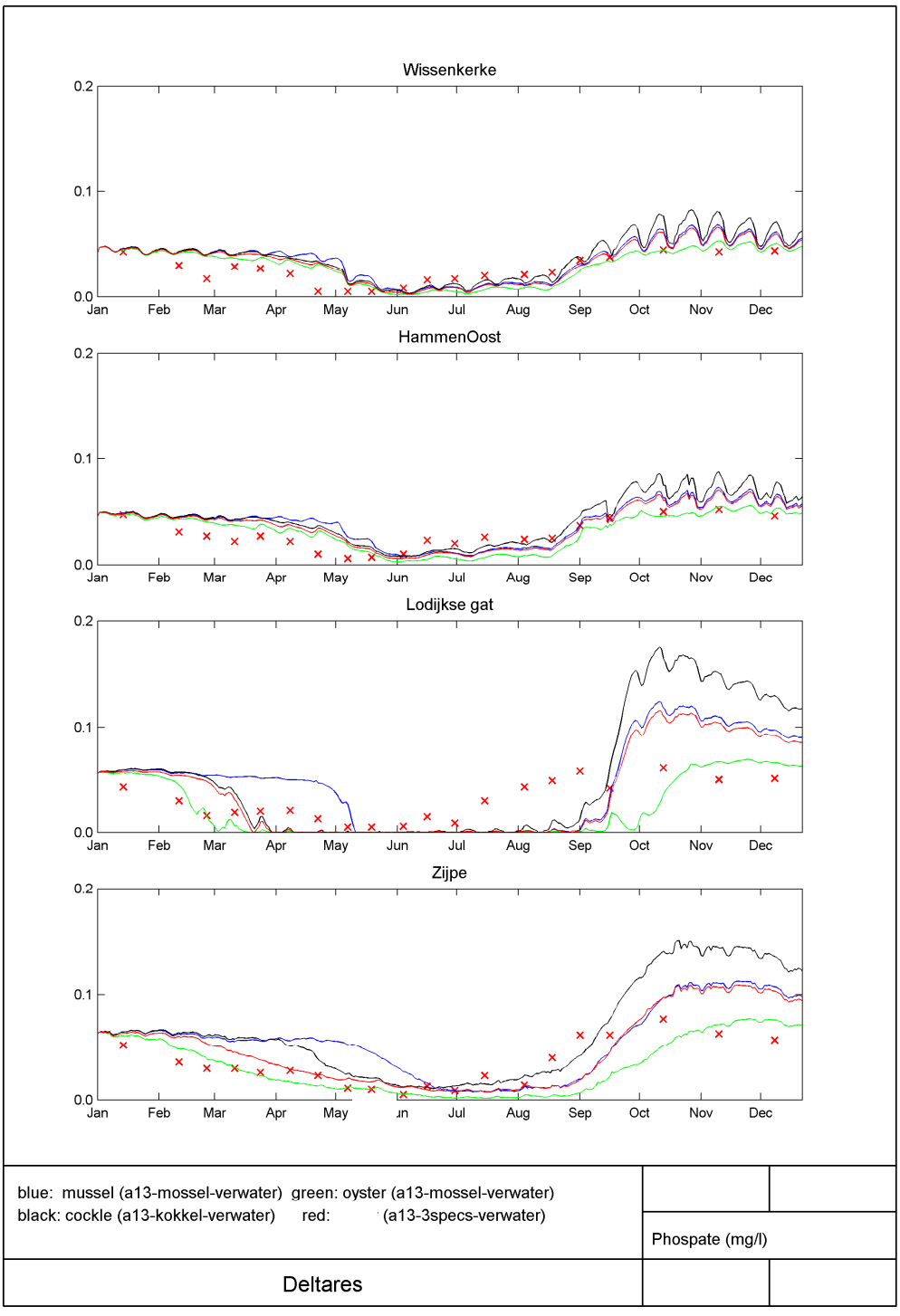


A.12 Nitrate: simulated NO3 of the 2D GEM with grazing of mussels (blue curves), cockles (black curves), oysters (green curves) and the three species simultaneously (red curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.

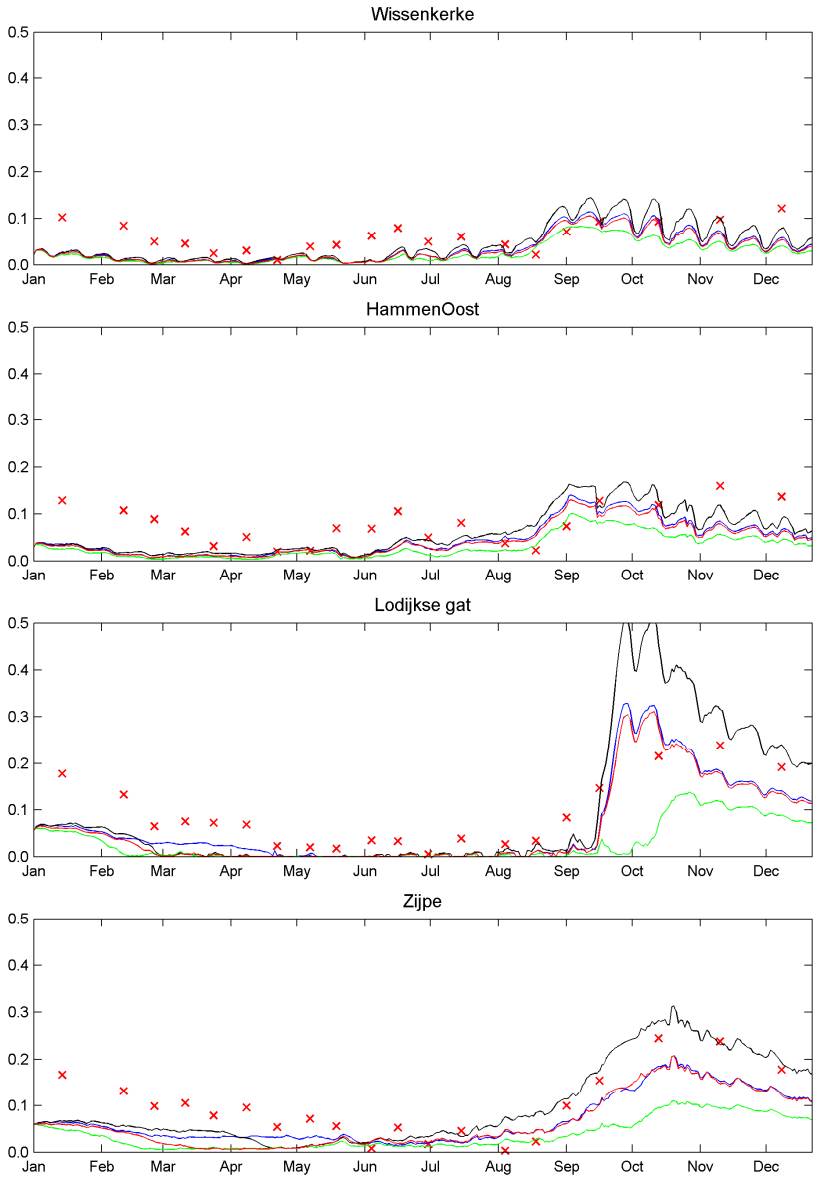


blue: mussel (a13-mossel-verwater) green: oyster (a13-mossel-verwater) black: cockle (a13-kokkel-verwater) red: oyster (a13-3specs-verwater)		
	Nitrate (mg/l)	
Deltares		

A.13 Phosphate: simulated PO4 concentrations of the 2D GEM with grazing of mussels (blue curves), cockles (black curves), oysters (green curves) and the three species simultaneously (red curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.



A.14 Ammonium: simulated NH4 concentrations of the 2D GEM with grazing of mussels (blue curves), cockles (black curves), oysters (green curves) and the three species simultaneously (red curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.

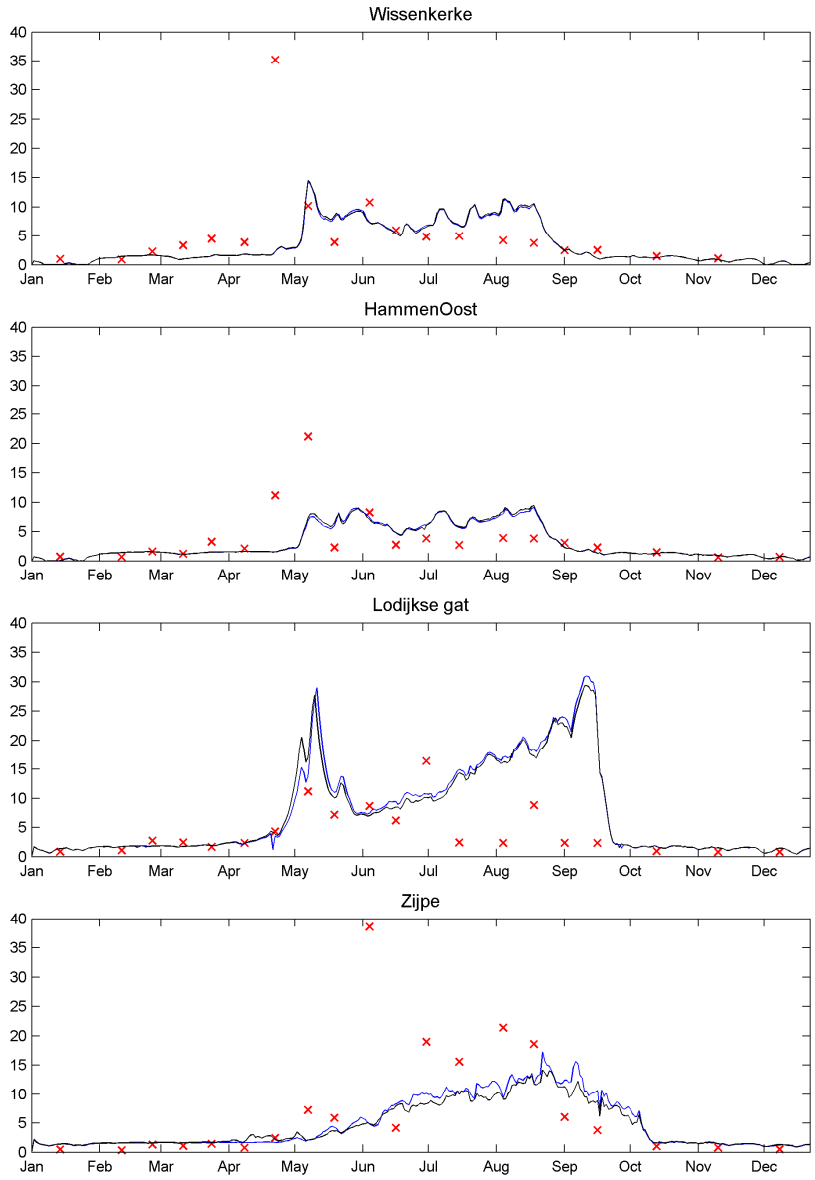


blue: mussel (a13-mossel-verwater) green: oyster (a13-mossel-verwater)
black: cockle (a13-kokkel-verwater) red: (a13-3specs-verwater)

Ammonium (mg/l)

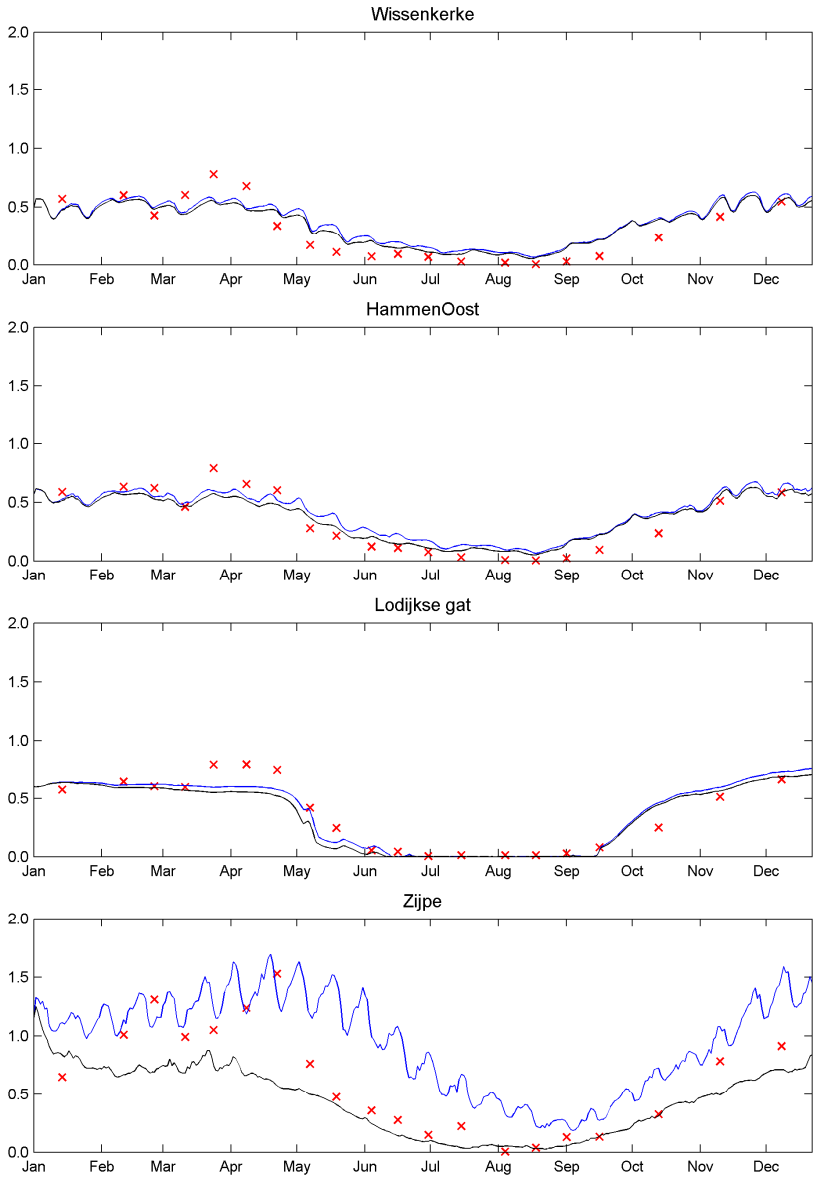
Deltares

A.15 Chlorophyll: simulated chlorophyll of the 2D GEM with grazing in the default scenario (blue curves), and in case of an increased discharge through the Krammer-sluices (black curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.



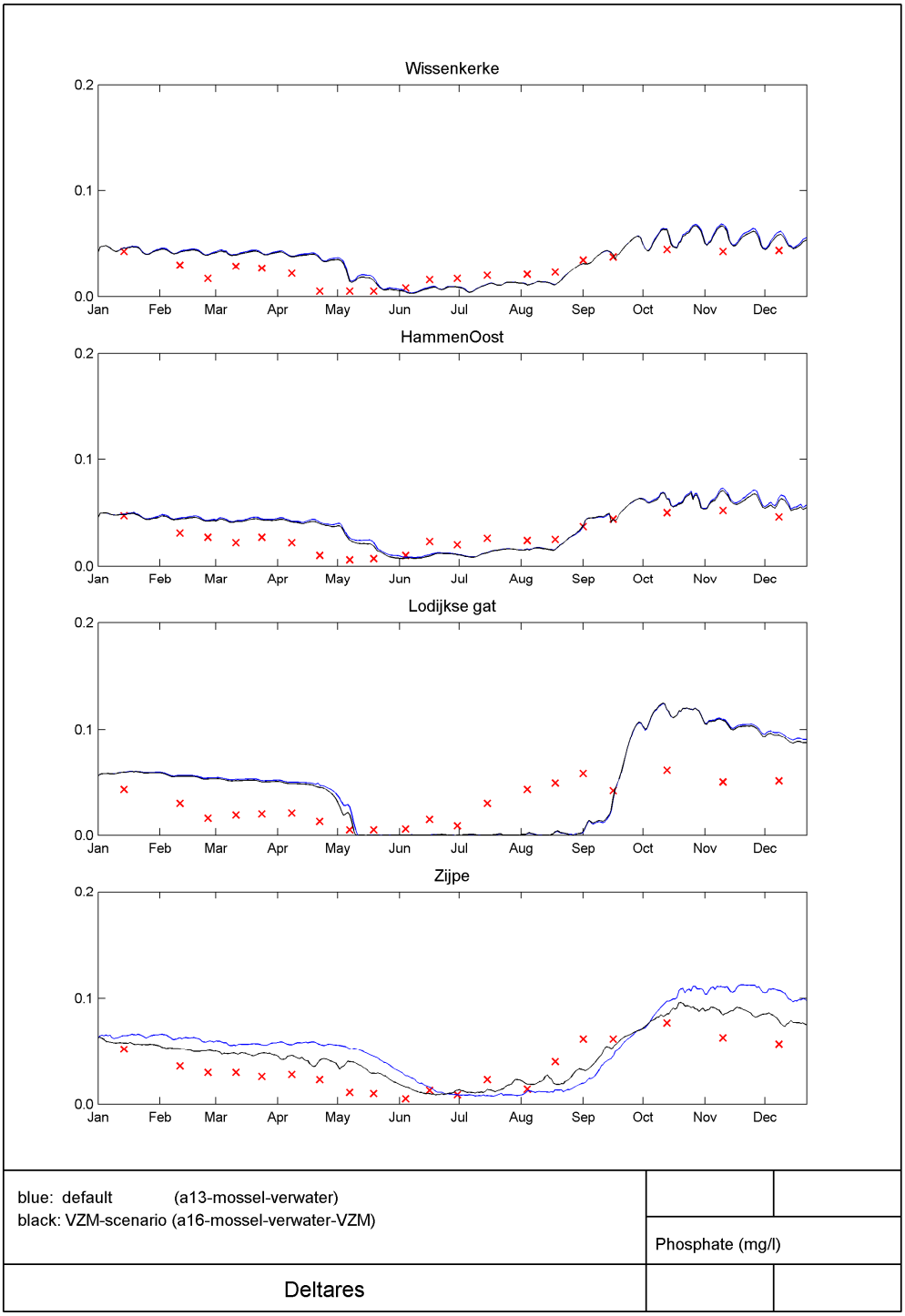
blue: default (a13-mossel-verwater) black: VZM-scenario (a16-mossel-verwater-VZM)		
	Chlorophyll (ug/l)	
Deltares		

A.16 Nitrate: simulated NO3 of the 2D GEM with grazing in the default scenario (blue curves), and in case of an increased discharge through the Krammer-sluices (black curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.

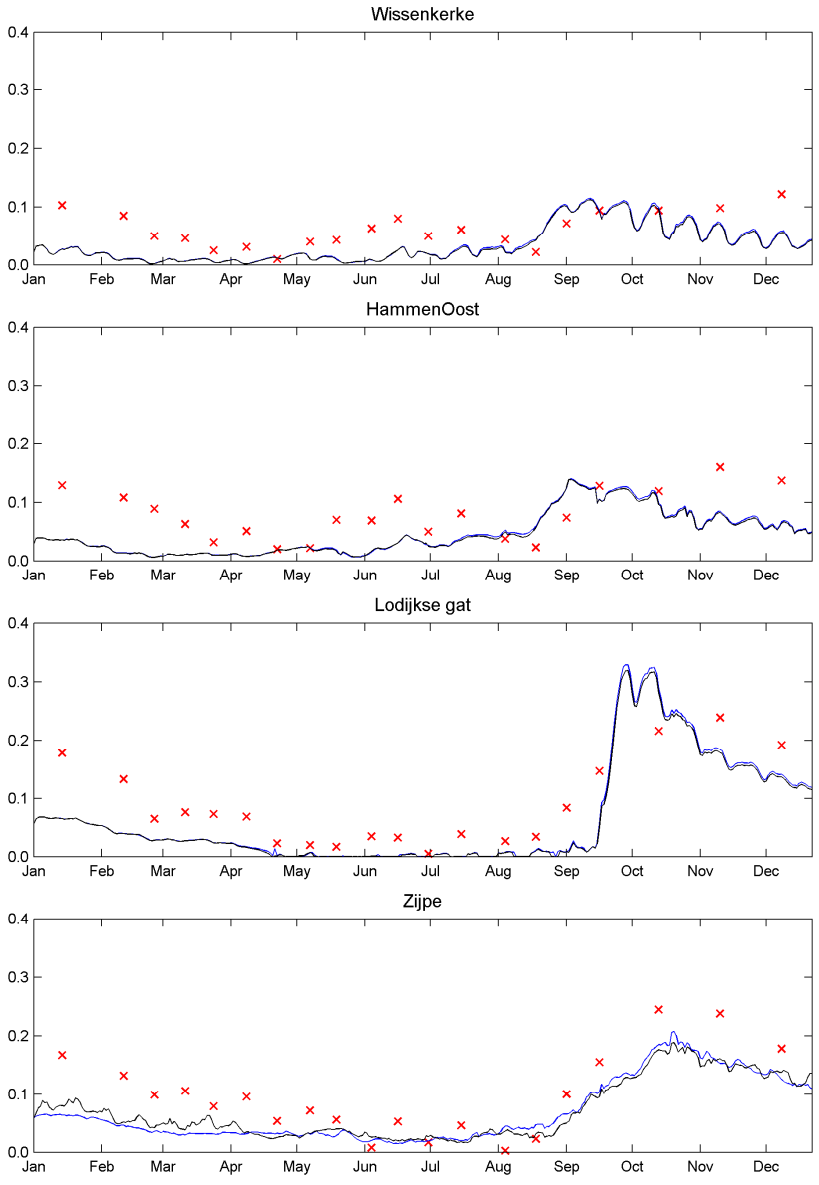


blue: default (a13-mossel-verwater) black: VZM-scenario (a16-mossel-verwater-VZM)		
	Nitrate (mg/l)	
Deltares		

A.17 Phosphate: simulated PO4 concentrations of the 2D GEM with grazing in the default scenario (blue curves), and in case of an increased discharge through the Krammer-sluices (black curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.



A.18 Ammonium: simulated NH4 concentrations of the 2D GEM with grazing in the default scenario (blue curves), and in case of an increased discharge through the Krammer-sluices (black curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.



blue: mussel (a13-mossel-verwater)
black: VZM-scenario (a16-mossel-verwater-VZM)

Ammonium (mg/l)

Deltares

B Model-code

B.1 DEB population growth for iso-morphs

```
      SUBROUTINE DEBISO  (PMSA , FL , IPOINT , INCREM , NOSEG , NOFLUX ,
     +                    IEXPNT, IKNMRK, IPODIM, NOQ1, NOQ2, NOQ3     )
C*******************************************************************************
C     +---------------------------------------+
C     |   D E L F T   H Y D R A U L I C S     |
C     |   WAter Resources and Environment     |
C     +---------------------------------------+
C
C*******************************************************************************
C
C     Project : Hogere trofische niveaus met DEB
C     Author  : Tineke Troost
C     Date    : 091012            Version : 0.01
C
C     History :
C
C     Date    Author         Description
C     ------  -------------  ----------------------------------
C     040116  Jeroen Wijsman  Created STORG
C     080820  Tineke Troost   Transformed the module into a DEB V1 morph
C     091012  Tineke Troost   Transformed the module into a DEB iso morph
C
C*******************************************************************************
C
C     Description of the module :
C
C        General water quality module for DELWAQ:
C        General routine for the dynamics of a standard organism. The
C        organism can consume various (pelagic and benthic) food types,
C        including dynamo and bloom algae and various detritus fractions
C        (DetX, POX and DetXS1). The consumer has a specific preference
C        for each food type.
C        The organism decreases in biomass by defaecation, respiration
C        and mortality
C
C Name      T   L I/O   Description                              Units
C ----      --- -  -    -------------------                      ----
C DELT        R*4 1 I    x timestep for processes                    (d)
C Volume      R*4 1 I    x volume of computational cell              (m3)
C Temp        R*4 1 I    x ambient water temperature                 (oC)
C Depth       R*4 1 I    x depth of segment                          (m)
C SWDetTyp    R*4 1 I    x use DetX (0) or POXi for GEM (1)          (-)
C Number      R*4 1 I    x Number of suspension feeders              (#/m2)
C V           R*4 1 I    x Volume of individual suspension feeder    (cm3)
C E           R*4 1 I    x Storage of individual suspension feeder   (J)
C R           R*4 1 I    x Reproductional storage of ind.            (J)
C Length      R*4 1 I    x actual length of individual               (cm)
C Lm          R*4 1 I    x maximum length of individual              (cm)
C Vp          R*4 1 I    x volume at start of reproductive stage     (cm3)
C shape       R*4 1 I    x shape coefficient                         (-)
C Em_L3       R*4 1 I    x Maximum storage density                   (J/cm3)
C Eg_L3       R*4 1 I    x Volume-specific costs for growth          (J/cm3)
C Pm_L3       R*4 1 I    x Volume-specific maintenance rate          (J/d)
C JXm_L2      R*4 1 I    x Maximum surface area-spec.ingestion rate (J/cm2/d)
C AE          R*4 1 I    x Assimilation efficiency                   (-)
C kappa       R*4 1 I    x fraction of util.energy spent on maint&growth  (-)
C Ta          R*4 1 I    x Arrhenius temperature                     (K)
C Tah         R*4 1 I    x Arr temp for rate of decrease at upper boundary(K)
C Tal         R*4 1 I    x Arr temp for rate of decrease at lower boundary(K)
C Th          R*4 1 I    x Upper boundary of tolerance range         (K)
C Tl          R*4 1 I    x Lower boundary of tolerance range         (K)
C GSI_upper   R*4 1 I    x Minimum GSI for spawning                  (-)
C rSpawn      R*4 1 I    x Spawning rate                             (-)
C MinSPTemp   R*4 1 I    x Minimum temperature for spawning          (oC)
```

```
C Xk_S        R*4 1 I    x Halfrate const pelagic food uptake Sup fdr (gC/m3)
C Xk_B        R*4 1 I    x Halfrate const benthic food uptake Sus fdr (gC/m2)
C rMor        R*4 1 I    x rate constant non-predation mortality Sus fdr (/d)
C fMor        R*4 1 I    x rate constant fishery mortality Sus feeders   (/d)
C c_J_gC      R*4 1 I    x conversion factor from J into gC           (gC/J)
C c_cm3_gC    R*4 1 I    x conversion factor from cm3 into gC        (gC/cm3)
C c_AFDW_gC   R*4 1 I    x conversion factor from gAFDW into gC    (gC/gAFDW)
C c_WW_gC     R*4 1 I    x conversion factor from gWW into gC        (gC/gWW)
C conv_cm3_J  R*4 1 I    x conversion factor from cm3 into J         (J/cm3)
C TC          R*4 1 I    x C:C ratio Suspens feeders                (gC/gC)
C TN          I*4 1 I    x N:C ratio Suspension feeders             (gN/gC)
C TP          R*4 1 I    x P:C ratio Suspension feeders             (gP/gC)
C TSi         R*4 1 I    x Si:C ratio Suspension feeders            (gSi/gC)
C BENTHS      R*4 1 I    x Use 0 [pelagic] or 1 [benthic] for Sus fdrs    (-)
C Pref(1)     R*4 1 I    x Preference of Sus feedrs for DetC or POC1      (-)
C Pref(2)     R*4 1 I    x Preference of Sus feeders for DetCS1           (-)
C Suspension  R*4 1 I    x Relative importance suspension feeding         (-)
C
C
C
C
C
C

C     Logical Units : -

C     Modules called : -

C     Name     Type   Library
C     ------   -----  ------------

      IMPLICIT  NONE

      REAL     :: PMSA  (*)  , FL  (*)
      INTEGER  :: IPOINT(*)  , INCREM(*), NOSEG , NOFLUX,
     +            IEXPNT(4,*), IKNMRK(*), IKMRK2, IPODIM, NOQ1,NOQ2, NOQ3

      INTEGER  :: NO_POINTER = 206 +11 ! number of input output variables in PMSA array
      INTEGER  :: IP(217)              ! index pointer in PMSA array updated for each segment

      INTEGER  :: I,IFOOD,IFLUX,ISEG,IKMRK1
      INTEGER  :: NFOOD = 25 +2        ! 25 food sources + DetCS1 + Detritus
      INTEGER  :: NTOTNUT = 4          ! Carbon, Nitrogen, Phosphorus and Silica
      INTEGER  :: BENTHS
C
C From PMSA array
      REAL     :: DELT  , Volume, temp , Depth, GEM , Number, Dens, V, E, R,
     +            rMor  , fMor , TC   , TN  , TP , TSi,
     +            Ta    , Suspension  , Tal,Tah, Th,Tl,
     +            Length  , Lm , Vp, shape  , Em_L3  , Eg_L3 , Pm_L3 ,
     +            JXm_L2  , PAm_L2  , kappa   , Xk_B, XK_S ,
     +            conv_J_cm3 , conv_J_gC , conv_cm3_gC,conv_gWW_gC,conv_gAFDW_gC

      REAL     ::  CFOOD   (27),  CCFOOD (27),  Pref   (27),
     +            NCFOOD  (27),  PCFOOD (27),  SICFOOD(27),
     +            dUpt    (27)
      INTEGER  ::  BENFOOD (27)

      REAL     ::  POM(4), DETRIT(4), DETS1(4), DETBIO(4)

      REAL     ::  Area, Preftot, Food, dDef, dNDef,  dPDef , dSiDef,
     +            dRes, dNRes  , dPRes  , StressResp,
     +            StressMort, dMor, dNMor, dPMor,
     +            kT, Q10fac, Uptake, Nuptake, Puptake, SiUptake,
     +            Nexcess, Pexcess,   Faeces, FoodPel, FoodBen,
     +            V_m2, E_m2, R_m2, dummy, Rdec,
     +            GSI, GSI_upper, MinSPtemp, DoSpawn, rSpawn, Spawning,
     +            Em_gC, Eg_gC , JXm_gC2,JXm_gC,
     +            Pa, Pc, Pj, Pr, Pg, Pm, Pd, Growth, kappa_R,
```

```
     +          f_B, f_S, AE, JXm_L3, Vd, Mv,convJC_L3, convJC_L2,
     +          Onethird, dDeftot,dNDeftot,dPDeftot

      REAL ::     Nin, Nuit, Pin, Puit, Siin, Siuit, Nbal, Pbal, Sibal,
     +          UptakeC, NuptakeC,PuptakeC,LimUptake,Biomass,BioAFDW,
     +          BioWW, SemiNetGr, GrossGr,Spawn, Harvest

      LOGICAL INIT
      SAVE INIT
      DATA INIT /.TRUE./

C     initialise pointers for PMSA and FL array
      IP = IPOINT(1:NO_POINTER)

      IFLUX = 0
      DO 9000 ISEG = 1 , NOSEG
      CALL DHKMRK(1,IKNMRK(ISEG),IKMRK1)
C     !if cell is active
        IF (IKMRK1.EQ.1) THEN
        CALL DHKMRK(2,IKNMRK(ISEG),IKMRK2)
C     !if cell has bottom
      IF ((IKMRK2.EQ.0).OR.(IKMRK2.EQ.3)) THEN

C Read input from first part of the PSMA
      DELT       =          PMSA( IP( 1))
      Volume     =          PMSA( IP( 2))
      Temp       =          PMSA( IP( 3))
      Depth      =          PMSA( IP( 4))
      GEM        =          PMSA( IP( 5))
      Number     =          PMSA( IP( 6))
        V        =          PMSA( IP( 7))
      E          =          PMSA( IP( 8))
      R          =          PMSA( IP( 9))
      Length     =          PMSA( IP(10))
      Lm         =          PMSA( IP(11))
      Vp         =          PMSA( IP(12))
      shape      =          PMSA( IP(13))
      Em_L3      =          PMSA( IP(14))
      Eg_L3      =          PMSA( IP(15))
      Pm_L3      =          PMSA( IP(16))
      JXm_L2     =          PMSA( IP(17))
      AE         =          PMSA( IP(18))
      kappa      =          PMSA( IP(19))
      Ta         =          PMSA( IP(20))
      Tah        =          PMSA( IP(21))
      Tal        =          PMSA( IP(22))
      Th         =          PMSA( IP(23))
      Tl         =          PMSA( IP(24))
      GSI_upper  =          PMSA( IP(25))
      rSpawn     =          PMSA( IP(26))
      MinSPTemp=            PMSA( IP(27))
      Xk_S       =          PMSA( IP(28))
      Xk_B       =          PMSA( IP(29))
      rMor       =          PMSA( IP(30))
      fMor       =          PMSA( IP(31))
      conv_J_gC=            PMSA( IP(32))
      conv_cm3_gC =         PMSA( IP(33))
      conv_gAFDW_gC=        PMSA( IP(34))
      conv_gWW_gC =         PMSA( IP(35))
      conv_J_cm3  =         PMSA( IP(36))
      TC         =          PMSA( IP(37))
      TN         =          PMSA( IP(38))
      TP         =          PMSA( IP(39))
      TSi        =          PMSA( IP(40))
      BENTHS     =          NINT(PMSA( IP(41)))
      Pref(1)    =          PMSA( IP(42))
      Pref(2)    =          PMSA( IP(43))
      Suspension  =         PMSA( IP(44))
```

```
      DO 51 I=1,NTOTNUT
        DETRIT(I)  = MAX(0.,PMSA(IP(44 + I             ))  )
        POM(I)     = MAX(0.,PMSA(IP(44 + I +     NTOTNUT))  )
        DETS1(I)   = MAX(0.,PMSA(IP(44 + I + 2 * NTOTNUT))  )

        DETBIO(I)  = MAX(0.,DETRIT(I)*(1.0-GEM) + POM(I)*GEM)
   51 CONTINUE

      DO 110 IFOOD=3,NFOOD
          CFOOD(IFOOD)  = MAX(0.,PMSA( IP(54 +             IFOOD)))
          CCFOOD(IFOOD) = 1.
          NCFOOD(IFOOD) =       PMSA( IP(54 +   (NFOOD-2) + IFOOD))
          PCFOOD(IFOOD) =       PMSA( IP(54 + 2*(NFOOD-2) + IFOOD))
          SiCFOOD(IFOOD)=       PMSA( IP(54 + 3*(NFOOD-2) + IFOOD))
          Pref(IFOOD)   =       PMSA( IP(54 + 4*(NFOOD-2) + IFOOD))
          BenFood(IFOOD)=  NINT (PMSA( IP(54 + 5*(NFOOD-2) + IFOOD)))
  110 CONTINUE

C Add Detbio and DetS1 to the food array's
C DetBIO is pelagic detritus
      CFOOD   (1) = DETBIO(1)
      CCFOOD  (1) = 1.
      BenFood (1) = NINT(0.)

      if (DETBIO(1).gt.1e-010) then
        NCFOOD(1)   = DETBIO(2) / DETBIO(1)
        PCFOOD(1)   = DETBIO(3) / DETBIO(1)
        SiCFOOD(1)  = DETBIO(4) / DETBIO(1)
      else
        NCFOOD(1)   = 0.
        PCFOOD(1)   = 0.
        SiCFOOD(1)  = 0.
      endif

C DetS1 is a benthic detritus
      CFOOD   (2) = DETS1(1)
      CCFOOD  (2) = 1.
      BenFood (2) = NINT(1.)

      if (DETS1(1).gt.0.) then
        NCFOOD(2)   = DETS1(2) / DETS1(1)
        PCFOOD(2)   = DETS1(3) / DETS1(1)
        SiCFOOD(2)  = DETS1(4) / DETS1(1)
      else
        NCFOOD(2)   = 0.
        PCFOOD(2)   = 0.
        SiCFOOD(2)  = 0.
      endif

C Statements

c Convert benthic components to units /m2
      Area = VOLUME / DEPTH

      if (BENTHS.eq.1) then
         Dens = Number/Area                    !(#/m2)
             E_m2 = E * Dens           !(J/m2)
        V_m2 = V * Dens           !(cm3/m2)
        R_m2 = R * Dens           !(J/m2)
c     else
c       V = V * Depth
      endif

       if (V.lt.1.000e-010) then
         V = 1.000e-010
      endif
      if (E.lt.1.000e-010) then
         E = 1.000e-010
      endif
```

```
c convert benthic food components to units gC m-2, do not convert pelagic components: unit stays gC m-3
      do 210 IFOOD = 1,NFOOD
         if (Benfood(ifood).eq.1) then
          CFOOD(IFOOD)=CFOOD(IFOOD) / AREA
         endif
210   CONTINUE



       Onethird = 1./3.


C Convert isomorphics to V1-morphics
c      Vd     = (shape*Length)**3.           !Vd is reference volume (cm) (niet nodig voor isomorph)
c      Mv     = (V_m2 / Vd)**(Onethird)      !Mv = Shape correction function (niet nodig voor isomorph)
C      Lv     = (V/shape**3)**(1/3)          !Lv = volumetric length

       Length = (V**(Onethird))/shape        !V is individual volume (cm)
       Mv = 1                                !Shape correction function


C Temperature dependent rates
C      Q10fac = Q10 ** ((temp-20.)/10.)
C      T=15
c      kT= exp((Ta/293.)-( Ta/(Temp +273.)))
       kT= exp(Ta/(20.+273.)- Ta/(Temp +273.))
     +      /(1.+ exp(Tal/(Temp+273.)-Tal/Tl)
     +         +  exp(Tah/Th-Tah/(Temp+273.)))


c effective food concentrations (gC/m3)
      FoodPel = 0.
      FoodBen = 0.
      do IFOOD = 1,NFOOD
         if (Benfood(ifood).eq.1) then
         CFood(IFOOD) = Pref(IFOOD) * CFood(IFOOD)
C             * (Pref(IFOOD) * CFood(IFOOD) /(Pref(IFOOD) * CFood(IFOOD) + LupBen))
         FoodBen = FoodBen + CFood(IFOOD)
         else
         CFood(IFOOD) = Pref(IFOOD) * CFood(IFOOD)
C        * (Pref(IFOOD) * CFood(IFOOD) /(Pref(IFOOD) * CFood(IFOOD) + LupPel))
         FoodPel = FoodPel + CFood(IFOOD)
         endif
      end do



c      OrganismX = 0.00


**********************************************************************
C UPTAKE: FILTRATION, INGESTION and ASSIMILATION
**********************************************************************
C Calculate filtration rate (m3 gC-1 d-1)
c      FiltRate = Filtmax * (KuptP) /
c     +          (FoodPel +  KuptP) * Q10Fac


C Calculate scaled functional respons FoodPel (-)

c     No assimilation and uptake when depth < 10 cm (to prevent uptake at dryfalling mudflats)

      IF(((IKMRK2.EQ.0).AND.(Depth.lt.0.05)).OR.
     & ((IKMRK2.EQ.3).AND.(Depth.lt.0.005))) THEN
```

```
c      if (Depth.lt. 0.05) then
         f_S = 0.
         f_B = 0.
       else
        f_S = (FoodPel / (FoodPel +  Xk_S))
        f_B = (FoodBen / (FoodBen +  Xk_B))
       endif

c to prevent division by zero in uptake rates
        if (FoodPel.eq.0.) then
           FoodPel=1.0e-10
        endif
        if (FoodBen.eq.0.) then
           FoodBen=1.0e-10
        endif

c Calculate uptake rates (J m-2 d-1)

       Uptake   = 0.
       Nuptake  = 0.
       PUptake  = 0.
       SiUptake = 0.

       do IFOOD=1,NFOOD
          if (Benfood(ifood).eq.1) then   ! Deposit feeding
            dUpt(IFOOD) = (1.-Suspension)*(Pref(IFOOD) * CFood(IFOOD) / FoodBen)
     +          * f_B * kT * Mv * (V**(2.*Onethird)) * JXm_L2
          else                            ! Suspension feeding
            dUpt(IFOOD) = Suspension * (Pref(IFOOD) * CFood(IFOOD) / FoodPel)
     +          * f_S * kT * Mv * (V**(2.*Onethird)) * JXm_L2
          endif

          Uptake   = Uptake   + dUpt(IFOOD) * CCFOOD(IFOOD)      !(J m-2 d-1)
          NUptake  = NUptake  + dUpt(IFOOD) * NCFOOD(IFOOD)      !(J m-2 d-1)
          PUptake  = PUptake  + dUpt(IFOOD) * PCFOOD(IFOOD)      !(J m-2 d-1)
          SiUptake = SiUptake + dUpt(IFOOD) * SiCFOOD(IFOOD)     !(J m-2 d-1)
       end do


*************************************************************************
C DEFEACATION
*************************************************************************
       dDef = 0.

C Part of Uptake, Nuptake and/or Puptake is released directly in order to correct
C N/C and P/C ratio of the ingested food to the N/C and P/C ratio of the Consumer
C Excess C,N and/or P is released directly as Faeces.
       if (uptake.gt.0.) then

          UptakeC   = Uptake * conv_J_gC                  ! Cuptake in (gC m-2 d-1)
          NuptakeC  = Nuptake * conv_J_gC / TN    ! Nuptake in carbon equivalents (gC m-2 d-1)
          PuptakeC  = Puptake * conv_J_gC / TP    ! Puptake in carbon equivalents (gC m-2 d-1)
          LimUptake = min(NuptakeC, PuptakeC, UptakeC)

          dDef  = (UptakeC  - LimUptake)          !(gC m-2 d-1)
          dNDef = (NuptakeC - LimUptake) * TN     !(gC m-2 d-1)
          dPDef = (PuptakeC - LimUptake) * TP     !(gC m-2 d-1)
C All uptake of silicate is lost by defecation
          dSiDef  = SiUptake * conv_J_gC          ! Si loss by def in carbon equivalents (gC m-2 d-1)
       else ! no food uptake, so no stoichiometric losses
          dDef = 0.
          dNDef = 0.
          dPDef = 0.
          dSiDef = 0.
       endif

C Additionally a part (qFec) of the ingested food is released as faeces
       PAm_L2 = JXm_L2 * Mv * AE * kT                 !(J m-2 d-1)
       Pa     = AE * (LimUptake)/conv_J_gC            !(J m-2 d-1)
```

```
      Faeces = (1. - AE) * (LimUptake)              !(gC m-2 d-1)
      dDeftot  = (dDef  + Faeces)                   !(gC m-2 d-1)
      dNDeftot = dNDef + Faeces * TN                !(gC m-2 d-1)
      dPDeftot = dPDef + Faeces * TP                !(gC m-2 d-1)

*************************************************************************
C ENERGY RESERVE DYNAMICS
*************************************************************************
C utilization rate
c       isomorph
        Pc = ((Eg_L3 / Em_L3) * PAm_L2 * V**(-1/3) + Pm_L3 *kT) /
     &      (kappa/V + Eg_L3/E)

C       V1 morph
c       Pc = ((Eg_L3 / Em_L3) * PAm_L2 * (V_m2**(-1.*Onethird))
c     &     + Pm_L3 * kT) /(kappa/V_m2 + Eg_L3/E_m2)      !(J m-2 d-1)

C       versimplificerende aanname, leidt wel tot massabalans-fout!
C       dE = (f * Em) - E
C       dE = (f- E / Em) *

*************************************************************************
C MAINTENANCE
*************************************************************************
C Respiration is only due to Basal respiration, not to activity or stress.
C Respiration of nutrients is related to the carbon respiration with ratios TN and TP
      Pm = Pm_L3 * V * kT             !(J m-2 d-1)


*************************************************************************
C GROWTH
*************************************************************************
C     Growth =(Assimilat/Organism)/(E+Eg/kappa)*(E/Em-length/maxlength)*Organism
      Pg = kappa * Pc - Pm                      !(J m-2 d-1)

      Growth = Pg / Eg_L3                       !(cm3 m-2 d-1)

*************************************************************************
C MATURITY and REPRODUCTION
*************************************************************************
c     dRecr  = 0.01   ! recruitment

      GSI = R_m2 / (V * conv_J_cm3 );
      kappa_R = 0.95
c     Vp = V

      if (V .LT. Vp) then
      Pj = ((1-kappa)/kappa) * Pm_L3 * V * kT       !(J m-2 d-1)
        Pd = ((1.-kappa)/kappa) * Eg_L3 * Growth !Energy costs development J d-1
      Pr = 0.
      else
      Pj = ((1-kappa)/kappa) * Pm_L3 * Vp * kT      !(J m-2 d-1)
      Pr = max((1-kappa) * Pc -Pj               ,0.)
        Pd = 0.
      endif

c     if too little energy or mat dev and maint, costs are paid by R
c Now, R can go through zero!!
      if (((1.-kappa)*Pc).LT.(Pj+Pd)) then
         Rdec= -1* (min(((1.-kappa) * Pc -Pj -Pd),0.))
        else
         Rdec=0.
        endif


c     Pr = max(0., ((1-kappa) * Pc -Pj))
      if (GSI.GT.GSI_upper) then
          if (Temp .GT. MinSPtemp) then
```

```
          DoSpawn  = 1.
        else
          DoSpawn  = 0.
        endif
      else
        DoSpawn = 0.
      endif

      if (DoSpawn.EQ.1.) then
        Spawning = rSpawn * R + kappa_R*Pr
      else
        Spawning = 0.
      endif


*******************************************************************
C RESPIRATION
*******************************************************************
       StressResp = 0.              ! Respiration due to stress not implemented
c     dRes =(Pm+Pj+Pd+Pg*(1-1/Eg_L3)+(1-kappa_R)*Pr + StressResp)*conv_J_gC                    !(gC m-2 d-1)
      dRes =(Pm+Pj+Pd+(1-kappa_R)*Pr + StressResp)*conv_J_gC                  !(gC m-2 d-1)
      dNRes = dRes * TN
      dPRes = dRes * TP

*******************************************************************
C MORTALITY
*******************************************************************
c Non-predation mortality rate (natural mortality and mortality due to fishery
c      StressMort = 0.                 ! Mortality due tot stress not implemented
       !rMor = 0.
      dMor  = rMor *(V*conv_cm3_gC+(E+R)*conv_J_gC)                  !(gC m-2 d-1)
      dNMor = dMor * TN
      dPMor = dMor * TP

*******************************************************************
C End of Statements
*******************************************************************

c Fluxes all in units of gX m-2 convert to g/m3/d for WAQ
c and to gC for benthic suspension feeders
c
c
      FL ( 1 + IFLUX  ) = dMor*Dens  /DEPTH
      FL ( 2 + IFLUX  ) = dNMor*Dens /DEPTH
      FL ( 3 + IFLUX  ) = dPMor*Dens /DEPTH
      FL ( 4 + IFLUX  ) = dRes*Dens  /DEPTH
      FL ( 5 + IFLUX  ) = dNRes*Dens /DEPTH
      FL ( 6 + IFLUX  ) = dPRes*Dens /DEPTH
      FL ( 7 + IFLUX  ) = dDeftot*Dens  /DEPTH
      FL ( 8 + IFLUX  ) = dNDeftot*Dens /DEPTH
      FL ( 9 + IFLUX  ) = dPDeftot*Dens /DEPTH
      FL (10 + IFLUX  ) = dSiDef*Dens  /DEPTH
      FL (11 + IFLUX  ) = Growth / VOLUME
      FL (12 + IFLUX  ) = (Pa - Pc) / VOLUME
      FL (13 + IFLUX  ) = (kappa_R*Pr - Rdec) / VOLUME
      FL (14 + IFLUX  ) = (rMor * Dens) /DEPTH
       FL (15 + IFLUX  ) = (fMor * Dens) /DEPTH
      FL (16 + IFLUX  ) = Spawning / VOLUME
      FL (17 + IFLUX  ) = (0.+ GEM)*dUpt(1)* conv_J_gC *Dens /DEPTH
      FL (18 + IFLUX  ) = (1.- GEM)*dUpt(1)* conv_J_gC *Dens /DEPTH
      FL (19 + IFLUX  ) =          dUpt(2)* conv_J_gC *Dens /DEPTH
      FL (20 + IFLUX  ) = (0.+ GEM)*dUpt(1)* NCFOOD(1) *conv_J_gC*Dens /DEPTH
      FL (21 + IFLUX  ) = (1.- GEM)*dUpt(1)* NCFOOD(1) *conv_J_gC*Dens /DEPTH
      FL (22 + IFLUX  ) =          dUpt(2)* NCFOOD(2) *conv_J_gC*Dens /DEPTH
      FL (23 + IFLUX  ) = (0.+ GEM)*dUpt(1)* PCFOOD(1) *conv_J_gC*Dens /DEPTH
      FL (24 + IFLUX  ) = (1.- GEM)*dUpt(1)* PCFOOD(1) *conv_J_gC*Dens /DEPTH
      FL (25 + IFLUX  ) =          dUpt(2)* PCFOOD(2) *conv_J_gC*Dens /DEPTH
      FL (26 + IFLUX  ) = (0.+ GEM)*dUpt(1)* SiCFOOD(1)*conv_J_gC*Dens /DEPTH
      FL (27 + IFLUX  ) = (1.- GEM)*dUpt(1)* SiCFOOD(1)*conv_J_gC*Dens /DEPTH
```

```
      FL (28 + IFLUX  ) =          dUpt(2)* SiCFOOD(2)*conv_J_gC*Dens /DEPTH

      do IFOOD=3,NFOOD
           FL(26 + IFOOD + IFLUX) = dUpt(IFOOD)*conv_J_gC*Dens /DEPTH
      enddo

c Check on budgets: Nbal, Pbal and Sibal should be zero.
      Nin   = NUptake * conv_J_gC * Dens/DEPTH
      Nuit  = FL(2+IFlux)+ FL(5+IFLUX)+FL(8+IFLUX)
      Pin   = PUptake * conv_J_gC * Dens/DEPTH
      Puit  = FL(3+IFlux)+ FL(6+IFLUX)+FL(9+IFLUX)
      Siin  = SiUptake * conv_J_gC/DEPTH
      Siuit = FL(10+IFLUX)
      Nbal  = Nin - Nuit - FL(11+IFLUX)*conv_cm3_gC*TN*Dens/Depth
     & -(FL(12+IFLUX)+FL(13+IFLUX))*conv_J_gC*TN*Dens/Depth
      Pbal  = Pin - Puit - FL(11+IFLUX)*conv_cm3_gC*TP*Dens/Depth
     & -(FL(12+IFLUX)+FL(13+IFLUX))*conv_J_gC*TP*Dens/Depth
      Sibal = Siin - Siuit
      Food  = FoodPel+FoodBen
c     GrossGr = (kappa * Pc /Eg_L3) * conv_cm3_gC / DEPTH
c     SemiNetGr = Growth * conv_cm3_gC / DEPTH
      Harvest=fMor/conv_gWW_gC*(V_m2*conv_cm3_gC+(E_m2+R_m2)*conv_J_gC)/DEPTH
       !(gC m-2 d-1)
       Spawn = (Spawning*conv_J_gC*Dens/ conv_gWW_gC) /DEPTH
      Biomass= V_m2 * conv_cm3_gC + (E_m2+R_m2) * conv_J_gC
      BioAFDW= Biomass / conv_gAFDW_gC
      BioWW=   Biomass / conv_gWW_gC


        PMSA(IP(207)) = Food
        PMSA(IP(208)) = Nbal
        PMSA(IP(209)) = Pbal
        PMSA(IP(210)) = SiBal
        PMSA(IP(211)) = V_m2
        PMSA(IP(212)) = E_m2
        PMSA(IP(213)) = R_m2
        PMSA(IP(214)) = Harvest
        PMSA(IP(215)) = Spawn
        PMSA(IP(216)) = BioWW
        PMSA(IP(217)) = Length



      ENDIF   ! (IKMRK1.EQ.1)
      ENDIF   ! (IKMRK2.EQ.0 OR 3)
C       update pointering in PMSA and FL array
        IFLUX = IFLUX + NOFLUX
        IP    = IP    + INCREM(1:NO_POINTER)

c
 9000 CONTINUE
c

      IF (INIT) INIT = .FALSE.

      RETURN
C
      END
```

B.2 DEB population growth for V1-morphs

```
      SUBROUTINE DEBV1  (PMSA , FL , IPOINT , INCREM , NOSEG , NOFLUX ,
     +                   IEXPNT, IKNMRK, IPODIM, NOQ1, NOQ2, NOQ3     )
C*******************************************************************************
C     +---------------------------------------+
C     |   D E L F T   H Y D R A U L I C S     |
C     |   WAter Resources and Environment     |
C     +---------------------------------------+
C
C*******************************************************************************
C
C     Project : Hogere trofische niveaus met DEB
C     Author  : Tineke Troost
C     Date    : 040116            Version : 0.01
C
C     History :
C
C     Date    Author          Description
C     ------  --------------  ---------------------------------
C     040116  Jeroen Wijsman  Created STORG
C     080820  Tineke Troost   Transformed the module into a DEB structure for V1 morphs
C
C*******************************************************************************
C
C     Description of the module :
C
C        General water quality module for DELWAQ:
C        General routine for the dynamics of a standard organism. The
C        organism can consume various (pelagic and benthic) food types,
C        including dynamo and bloom algae and various detritus fractions
C        (DetX, POX and DetXS1). The consumer has a specific preference
C        for each food type.
C        The organism decreases in biomass by defaecation, respiration
C        and mortality
C
C Name      T   L I/O   Description                                  Units
C ----      --- - -   --------------------                         ----
C DELT       R*4 1 I    x timestep for processes                    (d)
C Volume     R*4 1 I    x volume of computational cell              (m3)
C Temp       R*4 1 I    x ambient water temperature                 (oC)
C Depth      R*4 1 I    x depth of segment                          (m)
C SWDetTyp   R*4 1 I    x use DetX (0) or POXi for GEM (1)          (-)
C V          R*4 1 I    x tot. struct. volume of grazers in segment (cm3)
C E          R*4 1 I    x storage                                   (J)
C R          R*4 1 I    x reproductional storage                    (J)
C Lref       R*4 1 I    x actual length of individual               (cm)
C Lm         R*4 1 I    x maximum length of individual              (cm)
C Vp         R*4 1 I    x volume at start of reproductive stage     (cm3)
C shape      R*4 1 I    x shape coefficient                         (-)
C Em_L3      R*4 1 I    x maximum storage density                   (J/cm3)
C Eg_L3      R*4 1 I    x volume-specific costs for growth          (J/cm3)
C Pm_L3      R*4 1 I    x volume-specific maintenance costs         (J/d)
C JXm_L2     R*4 1 I    x maximum surface area-spec.ingestion rate (J/cm2/d)
C AE         R*4 1 I    x assimilation efficiency                   (-)
C kappa      R*4 1 I    x fraction of util.energy spent on maint&growth  (-)
C Ta         R*4 1 I    x arrhenius temperature                     (K)
C Tah        R*4 1 I    x arr temp for rate of decrease at upper boundary(K)
C Tal        R*4 1 I    x arr temp for rate of decrease at lower boundary(K)
C Th         R*4 1 I    x upper boundary of tolerance range         (K)
C Tl         R*4 1 I    x lower boundary of tolerance range         (K)
C GSI_upper  R*4 1 I    x minimum GSI for spawning                  (-)
C rSpawn     R*4 1 I    x spawning rate                             (-)
C MinSPTemp  R*4 1 I    x minimum temperature for spawning          (oC)
C Xk_S       R*4 1 I    x halfrate const pelagic food uptake Sup fdr (gC/m3)
```

```
C Xk_B        R*4 1 I    x halfrate const benthic food uptake Sus fdr (gC/m2)
C rMor        R*4 1 I    x rate constant non-predation mortality Sus fdr (/d)
C fMor        R*4 1 I    x rate constant fishery mortality Sus feeders   (/d)
C c_J_gC      R*4 1 I    x conversion factor from J into gC          (gC/J)
C c_cm3_gC    R*4 1 I    x conversion factor from cm3 into gC        (gC/cm3)
C c_AFDW_gC   R*4 1 I    x conversion factor from gAFDW into gC    (gC/gAFDW)
C c_WW_gC     R*4 1 I    x conversion factor from gWW into gC        (gC/gWW)
C conv_cm3_J  R*4 1 I    x conversion factor from cm3 into J         (J/cm3)
C TC          R*4 1 I    x C:C ratio Suspens feeders                (gC/gC)
C TN          I*4 1 I    x N:C ratio Suspension feeders             (gN/gC)
C TP          R*4 1 I    x P:C ratio Suspension feeders             (gP/gC)
C TSi         R*4 1 I    x Si:C ratio Suspension feeders            (gSi/gC)
C BENTHS      R*4 1 I    x use 0 [pelagic] or 1 [benthic] for Sus fdrs    (-)
C Pref(1)     R*4 1 I    x preference of Sus feedrs for DetC or POC1      (-)
C Pref(2)     R*4 1 I    x preference of Sus feeders for DetCS1           (-)
C Suspension  R*4 1 I    x relative importance suspension feeding         (-)
C
C
C
C
C
C

C     Logical Units : -

C     Modules called : -

C     Name     Type   Library
C     ------   -----  ------------

      IMPLICIT  NONE

      REAL     :: PMSA  (*)  , FL  (*)
      INTEGER  :: IPOINT(*)  , INCREM(*), NOSEG , NOFLUX,
     +            IEXPNT(4,*), IKNMRK(*), IKMRK2, IPODIM, NOQ1,NOQ2, NOQ3

      INTEGER  :: NO_POINTER = 205 +10 ! number of input output variables in PMSA array
      INTEGER  :: IP(215)              ! index pointer in PMSA array updated for each segment

      INTEGER  :: I,IFOOD,IFLUX,ISEG,IKMRK1
      INTEGER  :: NFOOD = 25 +2        ! 25 food sources + DetCS1 + Detritus
      INTEGER  :: NTOTNUT = 4          ! Carbon, Nitrogen, Phosphorus and Silica
      INTEGER  :: BENTHS
C
C From PMSA array
      REAL     :: DELT  , Volume, temp , Depth, GEM , V, E, R,
     +            rMor  , fMor , TC   , TN  , TP , TSi,
     +            Ta    , Suspension  , Tal,Tah, Th,Tl,
     +            Lref  , Lm , Vp, shape  , Em_L3  , Eg_L3 , Pm_L3 ,
     +            JXm_L2  , PAm_L2  , kappa   , Xk_B, XK_S ,
     +            conv_cm3_J,conv_J_gC, conv_cm3_gC , conv_gWW_gC, conv_gAFDW_gC

      REAL     ::  CFOOD   (27),  CCFOOD (27),  Pref   (27),
     +            NCFOOD  (27),  PCFOOD (27),  SICFOOD(27),
     +            dUpt    (27)
      INTEGER  ::  BENFOOD (27)

      REAL     ::  POM(4), DETRIT(4), DETS1(4), DETBIO(4)

      REAL     ::  Area, Preftot, Food, dDef, dNDef,  dPDef , dSiDef,
     +            dRes, dNRes  , dPRes  , StressResp,
     +            StressMort, dMor, dNMor, dPMor,
     +            kT, Q10fac, Uptake, Nuptake, Puptake, SiUptake,
     +            Nexcess, Pexcess,   Faeces, FoodPel, FoodBen,
     +            V_m2, E_m2, R_m2, dummy, Vd_m2,
     +            GSI, GSI_upper, MinSPtemp, DoSpawn, rSpawn, Spawning,
     +            Em_gC, Eg_gC , JXm_gC2,JXm_gC,
     +            Pa, Pc, Pj, Pr, Pg, Pm, Pd, Growth, kappa_R,
```

```
     +          f_B, f_S, AE, JXm_L3, Vd, Mv,convJC_L3, convJC_L2,
     +          Onethird, dDeftot,dNDeftot,dPDeftot , Rdec

      REAL ::     Nin, Nuit, Pin, Puit, Siin, Siuit, Nbal, Pbal, Sibal,
     +          UptakeC, NuptakeC,PuptakeC,LimUptake,Biomass,BioAFDW,
     +          BioWW, SemiNetGr, GrossGr, Spawn, Harvest

      LOGICAL INIT
      SAVE INIT
      DATA INIT /.TRUE./

C     initialise pointers for PMSA and FL array
      IP = IPOINT(1:NO_POINTER)

      IFLUX = 0
      DO 9000 ISEG = 1 , NOSEG
      CALL DHKMRK(1,IKNMRK(ISEG),IKMRK1)
C     !if cell is active
        IF (IKMRK1.EQ.1) THEN
        CALL DHKMRK(2,IKNMRK(ISEG),IKMRK2)
C     !if cell has bottom (this allows only benthic grazers)
      IF ((IKMRK2.EQ.0).OR.(IKMRK2.EQ.3)) THEN

C Read input from first part of the PSMA
      DELT        =              PMSA( IP( 1))
      Volume      =              PMSA( IP( 2))
      Temp        =              PMSA( IP( 3))
      Depth       =              PMSA( IP( 4))
      GEM         =              PMSA( IP( 5))
      V           =              PMSA( IP( 6))
      E           =              PMSA( IP( 7))
      R           =              PMSA( IP( 8))
      Lref        =              PMSA( IP( 9))
      Lm          =              PMSA( IP(10))
      Vp          =              PMSA( IP(11))
      shape       =              PMSA( IP(12))
      Em_L3       =              PMSA( IP(13))
      Eg_L3       =              PMSA( IP(14))
      Pm_L3       =              PMSA( IP(15))
      JXm_L2      =              PMSA( IP(16))
      AE          =              PMSA( IP(17))
      kappa       =              PMSA( IP(18))
      Ta          =              PMSA( IP(19))
      Tah         =              PMSA( IP(20))
      Tal         =              PMSA( IP(21))
      Th          =              PMSA( IP(22))
      Tl          =              PMSA( IP(23))
      GSI_upper   =              PMSA( IP(24))
      rSpawn      =              PMSA( IP(25))
      MinSPTemp   =              PMSA( IP(26))
      Xk_S        =              PMSA( IP(27))
      Xk_B        =              PMSA( IP(28))
      rMor        =              PMSA( IP(29))
      fMor        =              PMSA( IP(30))
      conv_J_gC   =              PMSA( IP(31))
      conv_cm3_gC =              PMSA( IP(32))
      conv_gAFDW_gC=             PMSA( IP(33))
      conv_gWW_gC =              PMSA( IP(34))
      conv_cm3_J  =              PMSA( IP(35))
      TC          =              PMSA( IP(36))
      TN          =              PMSA( IP(37))
      TP          =              PMSA( IP(38))
      TSi         =              PMSA( IP(39))
      BENTHS      =              NINT(PMSA( IP(40)))
      Pref(1)     =              PMSA( IP(41))
      Pref(2)     =              PMSA( IP(42))
      Suspension =               PMSA( IP(43))
      DO 51 I=1,NTOTNUT
        DETRIT(I)   = MAX(0.,PMSA(IP(43 + I          )) )
```

```
      POM(I)       = MAX(0.,PMSA(IP(43 + I +     NTOTNUT))  )
      DETS1(I)    = MAX(0.,PMSA(IP(43 + I + 2 * NTOTNUT))  )

      DETBIO(I)   = MAX(0.,DETRIT(I)*(1.0-GEM) + POM(I)*GEM)
   51 CONTINUE

      DO 110 IFOOD=3,NFOOD
         CFOOD(IFOOD)  =         MAX(0.,PMSA( IP(53 +              IFOOD)))
         CCFOOD(IFOOD) =         1.
         NCFOOD(IFOOD) =         PMSA( IP(53 +   (NFOOD-2) + IFOOD))
         PCFOOD(IFOOD) =         PMSA( IP(53 + 2*(NFOOD-2) + IFOOD))
         SiCFOOD(IFOOD)=         PMSA( IP(53 + 3*(NFOOD-2) + IFOOD))
         Pref(IFOOD)   =         PMSA( IP(53 + 4*(NFOOD-2) + IFOOD))
         BenFood(IFOOD)=  NINT (PMSA( IP(53 + 5*(NFOOD-2) + IFOOD)))
  110 CONTINUE

C Add Detbio and DetS1 to the food array's
C DetBIO is pelagic detritus
      CFOOD   (1) = DETBIO(1)
      CCFOOD  (1) = 1.
      BenFood (1) = NINT(0.)

      if (DETBIO(1).gt.0.) then
        NCFOOD(1)   = DETBIO(2) / DETBIO(1)
        PCFOOD(1)   = DETBIO(3) / DETBIO(1)
        SiCFOOD(1)  = DETBIO(4) / DETBIO(1)
      else
        NCFOOD(1)   = 0.
        PCFOOD(1)   = 0.
        SiCFOOD(1)  = 0.
      endif

C DetS1 is a benthic detritus
      CFOOD   (2) = DETS1(1)
      CCFOOD  (2) = 1.
      BenFood (2) = NINT(1.)

      if (DETS1(1).gt.0.) then
        NCFOOD(2)   = DETS1(2) / DETS1(1)
        PCFOOD(2)   = DETS1(3) / DETS1(1)
        SiCFOOD(2)  = DETS1(4) / DETS1(1)
      else
        NCFOOD(2)   = 0.
        PCFOOD(2)   = 0.
        SiCFOOD(2)  = 0.
      endif

C Statements

c Convert benthic components to units /m2
      Area = VOLUME / DEPTH

      if (BENTHS.eq.1) then
         E_m2 = E /Area          !(J/m2)
         V_m2 = V /Area          !(cm3/m2)
         R_m2 = R /Area          !(J/m2)
c     else
c        V = V * Depth
      endif

c setting minimum amount allows growth everywhere
       if (V_m2.lt.1.000e-010) then
         V_m2 = 1.000e-010
      endif
      if (E_m2.lt.1.000e-010) then
         E_m2 = 1.000e-010
      endif
```

```
c convert benthic food components to units gC m-2, do not convert pelagic components: unit stays gC m-3
      do 210 IFOOD = 1,NFOOD
         if (Benfood(ifood).eq.1) then
          CFOOD(IFOOD)=CFOOD(IFOOD) / AREA
         endif
210   CONTINUE



        Onethird = 1./3.


C Convert isomorphics to V1-morphics
        Vd     = (shape*Lref)**3.              !Vd is reference volume (cm3)
        Mv     = (V_m2 / Vd)**(Onethird)      !Mv is shape correction function
c       (NB Mv is hier niet dimensieloos maar klopt wel: Mv *V_m2^(2/3) = [cm2/m2])
C       Lv     = (V/shape**3)**(1/3)          !Lv is volumetric length (cm)


C Temperature dependent rates
C     Q10fac = Q10 ** ((temp-20.)/10.)
C     T=15
c       kT= exp((Ta/293.)-( Ta/(Temp +273.)))
        kT= exp(Ta/(20.+273.)- Ta/(Temp +273.))
     +      /(1.+ exp(Tal/(Temp+273.)-Tal/Tl)
     +         +  exp(Tah/Th-Tah/(Temp+273.)))


c effective food concentrations (gC/m3)
      FoodPel = 0.
      FoodBen = 0.
      do IFOOD = 1,NFOOD
         if (Benfood(ifood).eq.1) then
         CFood(IFOOD) = Pref(IFOOD) * CFood(IFOOD)
C             * (Pref(IFOOD) * CFood(IFOOD) /(Pref(IFOOD) * CFood(IFOOD) + LupBen))
         FoodBen = FoodBen + CFood(IFOOD)
         else
         CFood(IFOOD) = Pref(IFOOD) * CFood(IFOOD)
C        * (Pref(IFOOD) * CFood(IFOOD) /(Pref(IFOOD) * CFood(IFOOD) + LupPel))
         FoodPel = FoodPel + CFood(IFOOD)
         endif
      end do



c      OrganismX = 0.00


*************************************************************************
C UPTAKE: FILTRATION, INGESTION and ASSIMILATION
*************************************************************************
C Calculate filtration rate (m3 gC-1 d-1)
c      FiltRate = Filtmax * (KuptP) /
c     +          (FoodPel +  KuptP) * Q10Fac


C Calculate scaled functional respons FoodPel (-)

c     No assimilation and uptake when depth < 5 cm (to prevent uptake at dryfalling mudflats)
c      when schematisation is 3D, the minimum depth should be smaller (e.g. for 10 layers 0.5 cm)


       IF(((IKMRK2.EQ.0).AND.(Depth.lt.0.05)).OR.
     & ((IKMRK2.EQ.3).AND.(Depth.lt.0.005))) THEN
c      if (Depth.lt. 0.05) then
         f_S = 0.
         f_B = 0.
```

```
      else
       f_S = (FoodPel / (FoodPel +  Xk_S))
       f_B = (FoodBen / (FoodBen +  Xk_B))
      endif

c to prevent division by zero in uptake rates
       if (FoodPel.eq.0.) then
          FoodPel=1.0e-10
        endif
        if (FoodBen.eq.0.) then
          FoodBen=1.0e-10
        endif

c Calculate uptake rates (J m-2 d-1)

      Uptake   = 0.
      Nuptake  = 0.
      PUptake  = 0.
      SiUptake = 0.

      do IFOOD=1,NFOOD
         if (Benfood(ifood).eq.1) then   ! Deposit feeding
           dUpt(IFOOD) = (1.-Suspension)*(Pref(IFOOD) * CFood(IFOOD) / FoodBen)
     +          * f_B * kT * Mv * (V_m2**(2.*Onethird)) * JXm_L2
         else                          ! Suspension feeding
           dUpt(IFOOD) = Suspension * (Pref(IFOOD) * CFood(IFOOD) / FoodPel)
     +          * f_S * kT * Mv * (V_m2**(2.*Onethird)) * JXm_L2
         endif

         Uptake   = Uptake   + dUpt(IFOOD) * CCFOOD(IFOOD)      !(J m-2 d-1)
         NUptake  = NUptake  + dUpt(IFOOD) * NCFOOD(IFOOD)      !(J m-2 d-1)
         PUptake  = PUptake  + dUpt(IFOOD) * PCFOOD(IFOOD)      !(J m-2 d-1)
         SiUptake = SiUptake + dUpt(IFOOD) * SiCFOOD(IFOOD)     !(J m-2 d-1)
      end do


************************************************************************
C DEFEACATION
************************************************************************
      dDef = 0.

C Part of Uptake, Nuptake and/or Puptake is released directly in order to correct
C N/C and P/C ratio of the ingested food to the N/C and P/C ratio of the Consumer
C Excess C,N and/or P is released directly as Faeces.
      if (Uptake.gt.0.) then

         UptakeC   = Uptake * conv_J_gC                  ! Cuptake in (gC m-2 d-1)
         NuptakeC  = Nuptake * conv_J_gC / TN    ! Nuptake in carbon equivalents (gC m-2 d-1)
         PuptakeC  = Puptake * conv_J_gC / TP    ! Puptake in carbon equivalents (gC m-2 d-1)
         LimUptake = min(NuptakeC, PuptakeC, UptakeC)

         dDef  = (UptakeC  - LimUptake)          !(gC m-2 d-1)
         dNDef = (NuptakeC - LimUptake) * TN     !(gC m-2 d-1)
         dPDef = (PuptakeC - LimUptake) * TP     !(gC m-2 d-1)
C All uptake of silicate is lost by defecation
         dSiDef  = SiUptake * conv_J_gC          ! Si loss by def in carbon equivalents (gC m-2 d-1)
      else ! no food uptake, so no stoichiometric losses
         dDef = 0.
         dNDef = 0.
         dPDef = 0.
         dSiDef = 0.
      endif

C Additionally a part (qFec) of the ingested food is released as faeces
      PAm_L2 = JXm_L2 * Mv * AE * kT                !(J m-2 d-1)
      Pa     = AE * (LimUptake)/conv_J_gC           !(J m-2 d-1)
      Faeces = (1. - AE) * (LimUptake)              !(gC m-2 d-1)
      dDeftot  = (dDef  + Faeces)                   !(gC m-2 d-1)
```

```
      dNDeftot = dNDef + Faeces * TN                !(gC m-2 d-1)
      dPDeftot = dPDef + Faeces * TP                !(gC m-2 d-1)

*************************************************************************
C ENERGY RESERVE DYNAMICS
*************************************************************************
C utilization rate
c       isomorph
C       Pc = kT * ((Eg_L3 / Em_L3) * PAm_L2 * V**(-1/3) + Pm_L3 ) /
C      &       (kappa/V + Eg_L3/E)

C       V1 morph
        Pc = ((Eg_L3 / Em_L3) * PAm_L2 * (V_m2**(-1.*Onethird))
     &     + Pm_L3 * kT) /(kappa/V_m2 + Eg_L3/E_m2)      !(J m-2 d-1)

C       versimplificerende aanname, leidt wel tot massabalans-fout!
C       dE = (f * Em) - E
C       dE = (f- E / Em) *

*************************************************************************
C MAINTENANCE
*************************************************************************
C Respiration is only due to Basal respiration, not to activity or stress.
C Respiration of nutrients is related to the carbon respiration with ratios TN and TP
      Pm = Pm_L3 * V_m2 * kT            !(J m-2 d-1)


*************************************************************************
C GROWTH
*************************************************************************
C     Growth =(Assimilat/Organism)/(E+Eg/kappa)*(E/Em-length/maxlength)*Organism
      Pg = kappa * Pc - Pm                      !(J m-2 d-1)

      Growth = Pg / Eg_L3                       !(cm3 m-2 d-1)

*************************************************************************
C MATURITY and REPRODUCTION
*************************************************************************
c       dRecr  = 0.01   ! recruitment

      GSI = (R_m2) / ((1-(Vp/(Vp+Vd)))* V_m2 *conv_cm3_J )
      kappa_R = 0.95 ! fraction remaining after paying overhead costs, or egg survival
c       Vp = V

c       if (Vd .LT. Vp) then
c       Pj = ((1.-kappa)/kappa) * Pm_L3 * V_m2 * kT                 !(J m-2 d-1)
c       Pr = 0
c       else
      Pd = ((1.-kappa)/kappa) * Eg_L3 * Vp/(Vp+Vd) * Growth                 !maturity development
        Pj = ((1.-kappa)/kappa) * Pm_L3 *
     & (Vp/(Vp+Vd)+(1-Vp/(Vp+Vd))*Vp/Vd)* V_m2 * kT      !maturity maintenance (J m-2 d-1)

        Pr = max(((1.-kappa) * Pc -Pj -Pd),0.)
c       if too little energy or mat dev and maint, costs are paid by R
c Now, R can go through zero!!
      if (((1.-kappa)*Pc).LT.(Pj+Pd)) then
         Rdec= -1* (min(((1.-kappa) * Pc -Pj -Pd),0.))
        else
         Rdec=0.
        endif

c       endif


c       Pr = max(0., ((1-kappa) * Pc -Pj))
      if (GSI.GT.GSI_upper) then
         if (Temp .GT. MinSPtemp) then
            DoSpawn  = 1.
         else
```

```
          DoSpawn  = 0.
        endif
      else
        DoSpawn = 0.
      endif

      if (DoSpawn.EQ.1.) then
        Spawning = rSpawn * R_m2 + kappa_R*Pr
      else
        Spawning = 0.
      endif

*******************************************************************
C RESPIRATION
*******************************************************************
       StressResp = 0.              ! Respiration due to stress not implemented
c      dRes =(Pm+Pj+Pd+Pg*(1-1/Eg_L3)+(1-kappa_R)*Pr + StressResp)*conv_J_gC                    !(gC m-2 d-1)
      dRes =(Pm+Pj+Pd+(1-kappa_R)*Pr + StressResp)*conv_J_gC                  !(gC m-2 d-1)
       dNRes = dRes * TN
      dPRes = dRes * TP

*******************************************************************
C MORTALITY
*******************************************************************
c Non-predation mortality rate (natural mortality and mortality due to fishery
c      StressMort = 0.                ! Mortality due tot stress not implemented
       !rMor = 0.
      dMor  = rMor * (V_m2*conv_cm3_gC+(E_m2+R_m2)*conv_J_gC)              !(gC m-2 d-1)
      dNMor = dMor * TN
      dPMor = dMor * TP

*******************************************************************
C End of Statements
*******************************************************************

c Fluxes all in units of gX m-2 convert to g/m3/d for WAQ
c and to gC for benthic suspension feeders
c
c
      FL ( 1 + IFLUX  ) = dMor  /DEPTH
      FL ( 2 + IFLUX  ) = dNMor /DEPTH
      FL ( 3 + IFLUX  ) = dPMor /DEPTH
      FL ( 4 + IFLUX  ) = dRes  /DEPTH
      FL ( 5 + IFLUX  ) = dNRes /DEPTH
      FL ( 6 + IFLUX  ) = dPRes /DEPTH
      FL ( 7 + IFLUX  ) = dDeftot  /DEPTH
      FL ( 8 + IFLUX  ) = dNDeftot /DEPTH
      FL ( 9 + IFLUX  ) = dPDeftot /DEPTH
      FL (10 + IFLUX  ) = dSiDef  /DEPTH
      FL (11 + IFLUX  ) = (Growth - rMor*V_m2)  /DEPTH
      FL (12 + IFLUX  ) = (Pa - Pc - rMor*E_m2) /DEPTH
      FL (13 + IFLUX  ) = (kappa_R*Pr - rMor*R_m2 - Rdec) /DEPTH
      FL (14 + IFLUX  ) = (fMor*V_m2)/DEPTH
      FL (15 + IFLUX  ) = (fMor*E_m2)/DEPTH
      FL (16 + IFLUX  ) = (fMor*R_m2)/DEPTH
      FL (17 + IFLUX  ) = Spawning /DEPTH
      FL (18 + IFLUX  ) = (0.+ GEM)*dUpt(1) * conv_J_gC /DEPTH
      FL (19 + IFLUX  ) = (1.- GEM)*dUpt(1) * conv_J_gC /DEPTH
      FL (20 + IFLUX  ) =          dUpt(2) * conv_J_gC /DEPTH
      FL (21 + IFLUX  ) = (0.+ GEM)*dUpt(1) * NCFOOD(1)  * conv_J_gC /DEPTH
      FL (22 + IFLUX  ) = (1.- GEM)*dUpt(1) * NCFOOD(1)  * conv_J_gC /DEPTH
      FL (23 + IFLUX  ) =          dUpt(2) * NCFOOD(2)  * conv_J_gC /DEPTH
      FL (24 + IFLUX  ) = (0.+ GEM)*dUpt(1) * PCFOOD(1)  * conv_J_gC /DEPTH
      FL (25 + IFLUX  ) = (1.- GEM)*dUpt(1) * PCFOOD(1)  * conv_J_gC /DEPTH
      FL (26 + IFLUX  ) =          dUpt(2) * PCFOOD(2)  * conv_J_gC /DEPTH
      FL (27 + IFLUX  ) = (0.+ GEM)*dUpt(1) * SiCFOOD(1) * conv_J_gC /DEPTH
      FL (28 + IFLUX  ) = (1.- GEM)*dUpt(1) * SiCFOOD(1) * conv_J_gC /DEPTH
      FL (29 + IFLUX  ) =          dUpt(2) * SiCFOOD(2) * conv_J_gC /DEPTH
```

```
      do IFOOD=3,NFOOD
           FL(27 + IFOOD + IFLUX) = dUpt(IFOOD)*conv_J_gC /DEPTH
      enddo

c Check on budgets: Nbal, Pbal and Sibal should be zero.
c note that fmort and spawning are not included in the balances
c since these are subtracted later (outside this module) from the state vars to make  them show up in the prn file
      Nin   = NUptake * conv_J_gC/DEPTH
      Nuit  = FL(2+IFLUX)+FL(5+IFLUX)+FL(8+IFLUX)
c      + FL(14+IFLUX)*conv_cm3_gC*TN +
c     &       (FL(15+IFLUX)+FL(16+IFLUX)+FL(17+IFLUX))*conv_J_gC*TN
      Pin   = PUptake * conv_J_gC/DEPTH
      Puit  = FL(3+IFLUX)+FL(6+IFLUX)+FL(9+IFLUX)
c      + FL(14+IFLUX)*conv_cm3_gC*TP +
c     &       (FL(15+IFLUX)+FL(16+IFLUX)+FL(17+IFLUX))*conv_J_gC*TP
      Siin  = SiUptake * conv_J_gC/DEPTH
      Siuit = FL(10+IFLUX)
      Nbal  = Nin - Nuit - FL(11+IFLUX)*conv_cm3_gC*TN -
     & (FL(12+IFLUX)+FL(13+IFLUX))*conv_J_gC * TN
      Pbal  = Pin - Puit - FL(11+IFLUX)*conv_cm3_gC*TP -
     & (FL(12+IFLUX)+FL(13+IFLUX))*conv_J_gC * TP
      Sibal = Siin - Siuit
      Food  = FoodPel+FoodBen
        Harvest=fMor/ conv_gWW_gC* (V_m2*conv_cm3_gC+(E_m2+R_m2)*conv_J_gC)/DEPTH
        !(gC m-2 d-1)
        Spawn = (Spawning*conv_J_gC/ conv_gWW_gC) /DEPTH
c      GrossGr = (kappa * Pc /Eg_L3) * conv_cm3_gC / DEPTH
c      SemiNetGr = Growth * conv_cm3_gC / DEPTH
      Biomass= V_m2 * conv_cm3_gC + (E_m2+R_m2) * conv_J_gC
      BioAFDW= Biomass / conv_gAFDW_gC
      BioWW=   Biomass / conv_gWW_gC


        PMSA(IP(206)) = Food
        PMSA(IP(207)) = Nbal
        PMSA(IP(208)) = Pbal
        PMSA(IP(209)) = SiBal
        PMSA(IP(210)) = V_m2
        PMSA(IP(211)) = E_m2
        PMSA(IP(212)) = R_m2
        PMSA(IP(213)) = Harvest
        PMSA(IP(214)) = Spawn
        PMSA(IP(215)) = BioWW




      ENDIF   ! (IKMRK1.EQ.1)
      ENDIF   ! (IKMRK2.EQ.0 OR 3)
C       update pointering in PMSA and FL array
        IFLUX = IFLUX + NOFLUX
        IP    = IP    + INCREM(1:NO_POINTER)

c
 9000 CONTINUE
c

      IF (INIT) INIT = .FALSE.

      RETURN
C
      END
```