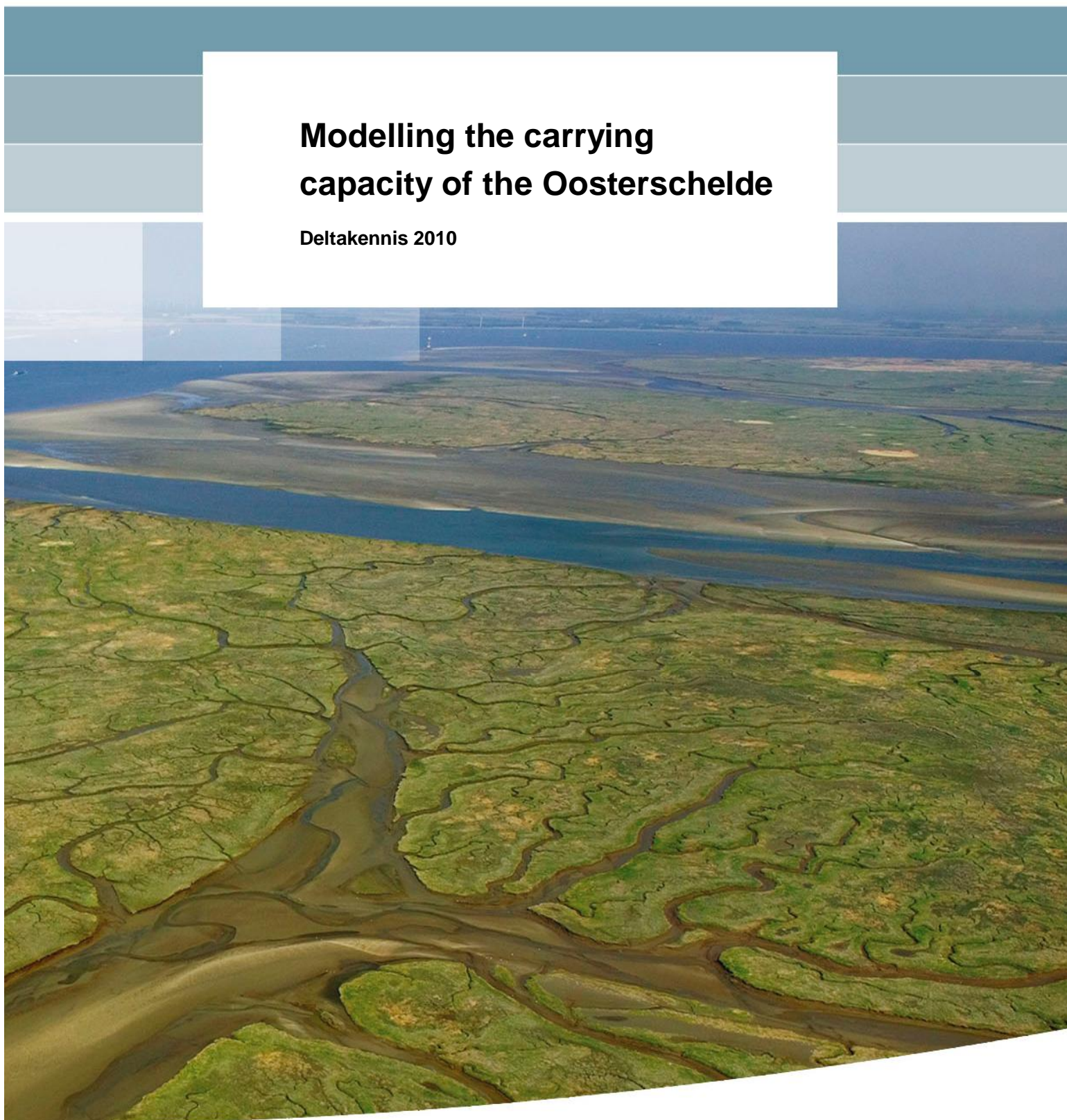**Deltares**

Enabling Delta Life

Modelling the carrying capacity of the Oosterschelde

Deltakennis 2010

Modelling the carrying capacity of the Oosterschelde

Deltakennis 2010

dr. T.A. Troost

1202193-000

Deltares

Title
Modelling the carrying capacity of the Oosterschelde

Client	Project	Reference	Pages
RWS Waterdienst	1202193-000	1202193-000-ZKS-0006	82

Keywords
Oosterschelde, carrying capacity, grazers, primary production, Deltakennis, erosion

Summary
The integrated ecosystem model for the Oosterschelde, which was set up in 2008 and 2009 within the 'Deltakennis'-framework, has been improved in several ways. As a result, modelled primary productions now show an acceptable fit with measurements. In comparison to previous results, however, the fit with measured nutrient and chlorophyll concentrations has somewhat deteriorated. Also, results suggest that the existing grazer pressure in the Oosterschelde may be larger than was predicted before.

In line with previous studies, including grazers in the model improves the model fit considerably. Hereby, it does not seem to make much difference whether grazers are modelled using one generalized grazer or three different species of grazers. The 3D-model provides a better fit with measurements than the 2D-model, but also requires a larger computational time.

Although the hydrodynamic part of the Oosterschelde GEM was set up as a 'proof-of-concept' model, the model was applied to study the effects of long-term mudflat erosion on the primary production of the system.

Version	Date	Author	Initials	Review	Initials	Approval	Initials
1.0	feb. 2011	dr. T.A. Troost		dr. W. Stolte		T. Schilperoort	

State
final

Contents

Appendices

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 dams are causing erosion of the intertidal flats. Also, various invasive species have appeared whose densities are still increasing (e.g. pacific oyster and American razor clam (*Ensis spp.*)). Furthermore, plans exist to take out some of the dams and make Lake Volkerak saline again. Another recent development is the use of specific devices to catch mussel seed (MZI's).

In view of these changes and 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, in 2008 Rijkswaterstaat has asked Deltares to develop a grazer module and include it into an ecosystem model of the Oosterschelde (Troost 2008). In a follow-up in 2009, this model was improved, and its results were validated against observed chlorophyll and nutrient concentrations, as well as carrying-capacity related parameters such as primary production and turnover time (Troost 2009).
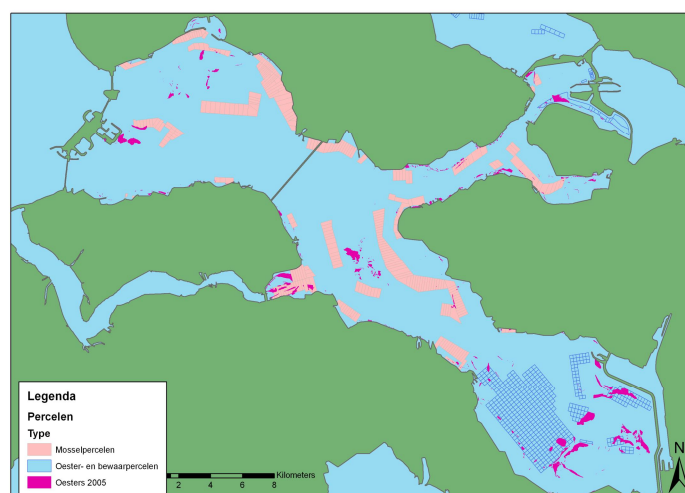
Figure 1.1. Mussel farms, oyster farms, and wild oyster populations in the Oosterschelde in 2005.

Objective for 2010

In 2010, Rijkswaterstaat has asked for further improvements of the model. These included setting up the model to simulate a more average year, recalibration of the model to improve the fit with measurements, formalization of the initial grazer values, and several technical improvements (see Section 1.2). Furthermore, the model was to be applied within the project framework 'Autonomous Negative Trends (ANT) Oosterschelde' to study the effects of erosion on primary production.

Readers' guide

Chapter 1 gives an introduction and describes the objective of this study. It includes an overview of past steps (Section 1.1) and present (Section 1.2) steps that were taken to set up and improve the Oosterschelde model and grazer module.

Chapter 2 provides a detailed description of the grazer module, starting with the underlying shellfish model for individual growth and its scaling up to a population model.

Chapter 3 describes the set-up of the ecosystem model for the Oosterschelde, its parameterisation and calibration. Results from various variants of the model are shown and validated.

In Chapter 4 the Oosterschelde model is applied to study the effects of erosion on primary production.

In Chapter 5 our conclusions on model improvements and results are summarized, and recommendations are given to improve the model performance in future.

1.1 History

The integrated ecosystem model for the Oosterschelde and the grazer module described in this report are elaborations of the Oosterschelde model and grazer module developed in the Deltakennis project of 2008 and 2009 (Troost 2008 & 2009). In these previous years, the following steps have been taken:

2008

1   Modelling 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 time series 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.

<u>2009</u>

5 Grazer Initialization and parameterization
Initial grazer densities were based on the actual (known) shellfish densities.
Furthermore, the model was parameterized and run for a generalized grazer species, for three different species, and for three different size classes.

6 Improvement code
The feedback fluxes were improved by removing an error from the code. Furthermore, steps were taken to facilitate the future implementation of pseudofaeces production in the model.

7 Calibration of salinity concentrations
The modelled salinity concentrations were improved by adjusting the north-western boundary concentrations.

8 Depth-related filtration threshold
A minimum-depth threshold for filtration was included such that, at low tide, grazers on dry-falling intertidal mudflats stop grazing.

1.2 This year's improvements and activities

• Average year
Originally, the model was set up to simulate 1998, which was a very wet year. Now, the model has been set up to simulate a more average and recent year, 2002. This included adjusting the discharges, loads, meteorology, and other forcing functions (Sections 3.2.1 and 3.3.4). Both the hydrodynamical and ecological model had to be rerun.

• Recalibration
The model has been recalibrated to result in a better fit with the measurements of nutrient concentrations and primary production. Adjustments involved the following parameters (Section 3.3.5):
 – light extinction
 – energy-content algae
 – phosphate adsorption and release from the sediment

• Grazers initialization and parameterization
The initialization and parameterization of grazers has been improved, standardized and documented (Section 3.3.5, Appendix C). Furthermore, the half-saturation constant for oysters and the algal energy content have been validated by data from flume experiments (Section 3.3.5, Appendix D).

• Technical improvements
The grazer module was made compatible with the newest version of the Delwaq-executable and thereby with the newest integration routine (Section 3.3.1).

- Case study on erosion

The model was applied to study the impact of large-scale mudflat erosion on the primary production of the system (Chapter 4).

- SMC-mussels

In a related project, the model was used to investigate the impact of Seed Mussel Capture (SMC) devices on the carrying capacity of the Oosterschelde system (Troost en van Duren, 2010).

2 The grazer module

2.1 DEB model for individual growth

2.1.1 DEB principles

The Dynamic Energy Budget theory is a modelling framework based on first principles and simple physiology-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 generic models of organism and population growth, and can be used for basically any species or life stage. Furthermore, DEB models can accommodate a variety of complexities ranging from a high level of physiological detail to simplified 'toy'-models.

The aspect that makes DEB framework unique and separates it from so-called "net production" models, are its energy 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 periods in-between meals. In net-production models, this requires implementation of various artificial switches in the model that are not based on physiological processes. The common DEB structure is given in Figure 2.1.
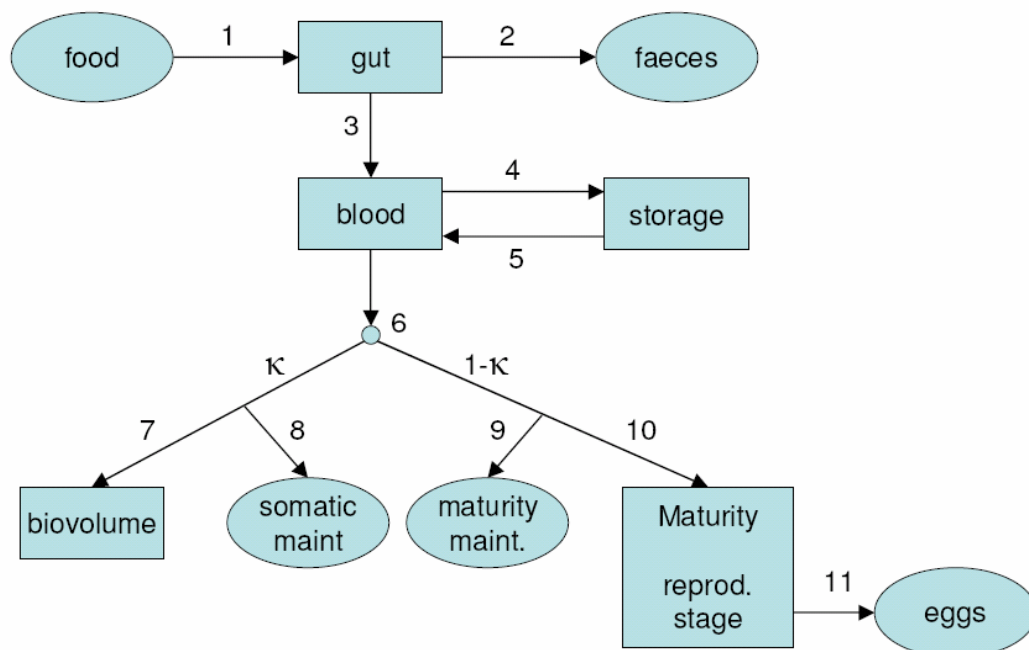
Figure 2.1. General structure of a DEB model with (1) ingestion, (2) defecation, (3) assimilation, (4 and 5) storage or reserve dynamics, (6) utilization, (7) growth, (8) maintenance, (9) maturation, (10) reproduction, and (11) spawning.

In addition to its generality, the DEB framework is also flexible and the models can be extended to include any species-specific characteristics that are necessary for a certain application. In case of modelling shellfish, certain adjustments are made to include filter feeding, spawning, and pseudofaeces production (Section 2.1.3).

2.1.2 General DEB equations

The general DEB equations described below apply to the growth of an individual organism. 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), energy reserves (E, Joule) and reproductive buffer (R, Joule).

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 (see Section 2.2).

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

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 (a_e) is independent of the feeding rate and the assimilation rate (p_A, J d^{-1}), which is calculated by

$$p_A = p_X \cdot a_e$$

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 rate and the mobilization rate (p_C, J d^{-1}), while the dynamics of the structural volume (growth) are based on the κ-fraction of the mobilization flux and somatic maintenance as follows:

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

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

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

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.

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. 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 energy flux going to reproduction is given by:

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

Derived variables

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 weight-converted 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}).

Temperature dependency
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.

2.1.3   Shellfish-specific DEB equations

The DEB models for shellfish are based on the standard DEB model described above. Some additions are made to the standard DEB model to incorporate shellfish-specific aspects. These additions are not new but have been included before in other shellfish modelling studies using DEB (Bacher & Gangnery 2006, Pouvreau *et al.* 2006, Rosland et al. 2009).

Functional response
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):

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

where FOOD is the available density of food (mgC/l), and Y is the concentration of inorganic matter (mg/l) 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, at equal food concentration, 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. The negative effect can be compensated by higher food concentration (competitive inhibition).

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 ecosystem model, however, FOOD may consist of both algae and detritus.

Spawning
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.

Selective feeding and pseudofaeces
Shellfish can (passively and actively) select their food. Active selection will lead to pseudofaeces production. Pseudofaeces production itself does not affect growth differently than faeces production, but the two products have different characteristics with respect to sedimentation and mineralization. At the moment, however, the exact differences between faeces and pseudofaeces with regard to decaying rates are not well known.

Though (active and passive) selection of food sources is technically implemented in the model, pseudofaeces production is not. A module for pseudofaeces production has already been developed (van de Wolfshaar, 2007), which may be used for this purpose.

Feeding at intertidal mudflats
Obviously, shellfish cannot feed during periods at which they fall dry. In the model, this is taken into account by setting the shellfish' functional response to zero when the depth of the grid cell in which it is located becomes smaller than 0.1m, which in practice is equal to falling dry.

2.2 Isomorphs and V1-morphs

Two different types of organisms can be modelled using the equations above. On the one hand, there are organisms that do not change shape during their life. A growing organism that does not change in shape during its life is called an **isomorph** (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. This makes it possible to model a complete growth trajectory without requiring size classes or (other) artificial switches in the model.

On the other hand, some organisms do change shape during their life. A specific class of shape-changing organisms are those that have a constant surface-to-volume ratio, such as

rod- or filament-shaped bacteria, and are called **V1-morphs** (Kooijman 2000). This approach is also suitable for small organisms such as unicellulars, since the change in size during their lives is relatively small.

The choice for isomorphs or V1-morphs can be made 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 in section 2.2.1. For isomorphs M_v is equal to 1. For individual V1-morphs, the reference length determines the surface area to volume ratio.

2.3   From individual based model to population model

The DEB model described in the section above applies to the growth and reproduction of an individual organism. 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 isomorphs, however, this would lead 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 age classes (cohorts) 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 age class thus follow the same growth trajectory. To go from individual isomorphs to such age classes of isomorphs, some additional information is required on initial values, larval settlement and mortality, which issues are discussed below.

2.3.1   Population of isomorphs

Density
An additional state variable is needed to keep track of the total number of individuals in the population or age class. This variable is affected only by mortality, since recruitment is included as a new age class, and thus does not lead to an increase in the number of individuals in the existing age classes. The number of individuals can be used to calculate density 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) per individual.

Initialization
To describe each age 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 age class of young/small individuals has to be included periodically in the simulation (e.g. once a year).

Mortality and starvation
Another important population process is mortality. The mortality rate constant was calculated assuming a first order decrease:

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

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.

Starvation occurs when the growth rate and/or reproduction rate become negative, which is 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 our isomorphic model, such negative growth rates will simply lead to a decrease in the structural volume. The corresponding interpretation is that in case the reserves are not sufficient, the organisms will shrink. Starvation will thus not lead to additional mortality. Would this however be desired, this can be changed in the code.

Equations and model code for a population (size-class) of isomorphs
The state of the isomorphic population is described by its state variables 'individual structural body volume' (V) in units of cm^3 and 'energy reserves' (E) expressed in units of J. The reserves can also be expressed as energy density (E/V) in units of J cm^{-3}.

The full model code of the isomorphic-population model as it is incorporated in the GEM is provided in Appendix B.1. Note that the units and interpretation of the used state variables slightly differ from those in the isomorphic population model (see Table 2.1), as is explained below (Section 2.3.2).

2.3.2 Population of V1-morphs

The model for isomorphs described above requires four state variables per age or 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 age classes is not available.

When one of the above mentioned problems cannot be overcome, or if detailed output on the various age 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 is to approximate the population of differently-sized and growing individuals by a population of equally sized organisms that do not change in size (V1-morphs). This assumption is valid as long as the population does not change its size-distribution too much over time. In this case, the reference length will characterize the population size composition.

Modelling a population of V1-morphs instead of isomorphs greatly simplifies the model structure, since one of the state variables (length, and thus the structural volume of an individual) becomes a constant. In other words, 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. As a result, a whole population can be simulated by three state variables (the total structural volume of the population, the energy density and reproductional density). Note that the population can still be split up into various size-classes.

Still, some additional information on reproduction and mortality is required. These 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. Therefore, it is not needed to simulate 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 2.1).

Larval settlement

When simulating the population by V1-morphs, settlement of larvae can simply be implemented by an increase of the total population size.

Mortality and starvation

For mortality of V1-morphs, the same rules apply as for the isomorphs (see Section 2.3.1). 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. While the corresponding interpretation for isomorphs was that the organisms become smaller (shrinking), in the V1-model this is equivalent to a fraction of the organisms dying (starvation mortality).

Maturity 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. In the isomorphic population model, the transition of juvenile to adult occurs at fixed size (V_p). In the population model of V1-morphs, 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:

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

Derived variables
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):

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

Model equations and code for V1-Morphs
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 expressed as energy density (E/V) in units of J cm^{-3}.

The full model code of the V1-population model as it is incorporated in the GEM is provided in Appendix B.2. Note that the units and interpretation of the used state variables slightly differ from those in the isomorphic population model, where they still correspond to the individual state (see Table 2.1).

Table 2.1 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}**

2.4 Parameter values

2.4.1 Physiological parameter values

Values for physiological parameters are obtained from Van der Veer et al. (2006). This 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 2.2).

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 in-between the values of oysters and cockles.

The following conversion factors were used:1 gWW = 1cm^3 = 0.12g AFDW and 1 gAFDW = 0.4 gC = 23kJ.

Table 2.2. 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

2.4.2 System specific parameters

The species-specific parameter values discussed above are constant across systems. However, various species-specific parameter values are also system-specific. The species- and system-specific parameters that are needed to describe the population dynamics of the grazers include:

- initial values,

- mortality rates,

- reference lengths,

- half-saturation constants, and

- algal energy content

Setting and calibration of these parameters is discussed in section 3.3.5.

3 Integrated ecosystem model – Oosterschelde

The integrated ecosystem model for the Oosterschelde is set up using Delft3D-software. It consists of a hydrodynamic model (Delft3D-FLOW) and an ecosystem model (Delft3D-BLOOM/GEM) both of which are briefly discussed below.

Originally, the model was set up for the year 1998 (Troost 2008 & 2009). Below we describe the model set-up and results for the year 2002, since this is a more recent year with more average weather conditions.

3.1 Oosterschelde bathymetry and GRID

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 in the horizontal plane (Figure 3.1), and 5 layers in the vertical in case of the 3D model.
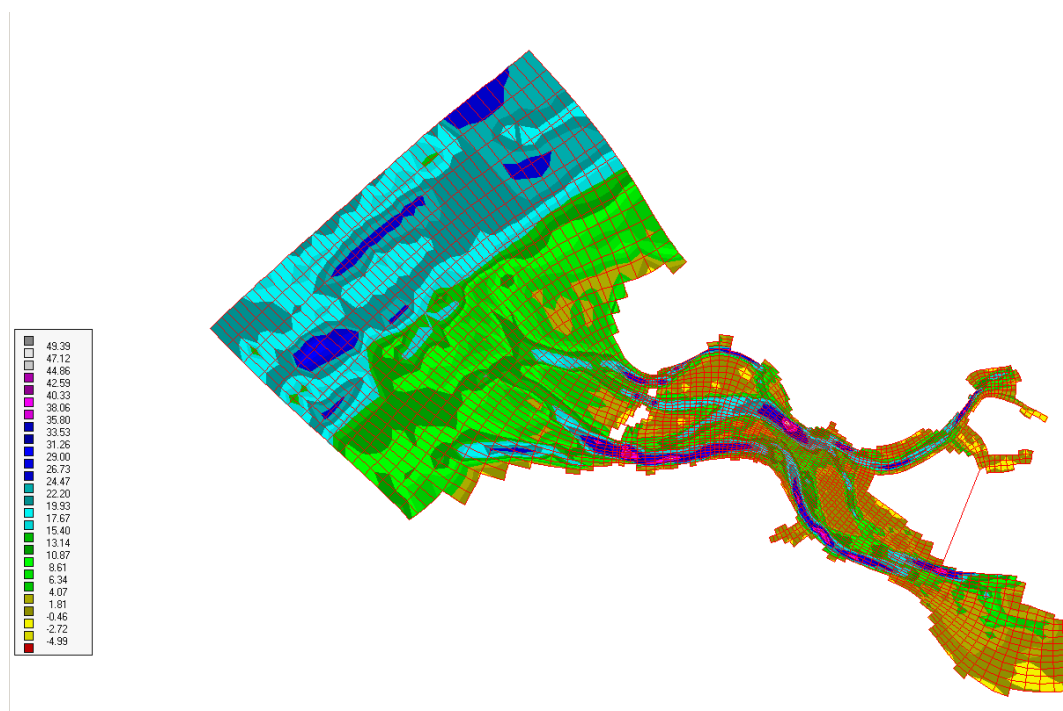
Figure 3.1. Grid and bathymetry of the Oosterschelde

3.2 Oosterschelde FLOW set-up

Delft3D-FLOW software is used to calculate the hydrodynamics of the Oosterschelde. Delft3D-FLOW is a multi-dimensional (2D or 3D) hydrodynamic (and transport) simulation program which calculates non-steady flow and transport phenomena that result from tidal and

meteorological forcing on a rectilinear or a curvilinear boundary fitted grid. Process details and set-up of this program are described in (WL | Delft Hydraulics 2006).

The boundary water levels were obtained from the ZUNO-model using the nesting procedure provided through the Delft3D interface. The number and thickness of the layers is set equal to the ZUNO-model (N=10) to enable the nesting procedure. The model includes precipitation and evaporation. The FLOW model was run with a time step of 1 minute. The FLOW-output was provided for a 2D-GEM model (existing of one vertical layer), and for a 3D-GEM model (existing of five layers).

3.2.1 Forcing funtions

Meteorological data
Meteorological data includes wind speed, precipitation and evaporation data from location Wilhelminadorp, which were obtained from KNMI.

Discharge data
Discharge quantities were obtained from the 1D-Deltamodel (Meijers et al 2007). Since no data were available for the discharges from the Krammersluices in 2002, initially the average discharges for the period 2000-2005 were used. However, this resulted in a poor fit with measured salinities and nitrogen concentrations in the Northern compartment. Therefore, for the Krammersluices, discharge data for 2000 were used instead. These were smaller than the average discharges and resulted in a better fit.

3.2.2 Parameterisation and validation

Physical parameter values en settings
Physical parameter values and settings were based on those as used in the model of lake Grevelingen (Nolte et al, 2008).

Validation by water levels
The hydrodynamic model has only been roughly validated on basis of water levels and salinity concentrations.

3.2.3 Model limitations

The hydrodynamic model has only been roughly validated on basis of water levels. Water balance and transports have not been checked. Therefore, the hydrodynamic model should be considered a 'proof of concept' model, and its results should thus be interpreted with caution.

3.3 Oosterschelde GEM set-up

Standard Delft3D-DELWAQ 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

These processes are discussed in detail in (WL | Delft Hydraulics 2002 & 2003).

3.3.1 Coupling D3D-FLOW and GEM, time steps and integration routines

The coupling of the FLOW and the GEM models was done on the fly ('online'), providing output at a time step of 1 hour. To avoid the GEM-model from crashing, too large flow velocities were removed by running the customized software "flow-check" which artificially increases the problematic cell-volumes. As a compromise between large time steps (small computational time) and large volume adjustments (changing the system behaviour), the GEM-time step was set to 2 minutes. With an integration routine causing numerical dispersion (upwind scheme, no. 1) and horizontal dispersion coefficients of 1 this leads to good results. Using an integration routine that does not involve numerical dispersion (flux correct transport method, no. 5), combined with horizontal dispersion coefficients of 50, leads to similar results.

An alternative to the volume adjustments discussed above has recently become available in the form of a new integration routine (no. 22). This routine can avoid crashing by switching between explicit and implicit integration. To be able to make use of this routine, the grazer-module first had to be made compatible with the newest Delwaq version in which the routine is available. This included some adjustments of the module-code, recompilation with the new fortran compiler and building a new 'dll' for the grazer module. Also, the process definition file had to be extended with the process definitions of the grazer module. The integration routine was tested and was shown to lead to the same results as the integration routine used before.

Although the new integration routine enables larger GEM time steps without crashing, larger time steps (10 and 20 minutes) led to unacceptable changes in the system behaviour. Therefore, the time step was left at 2 minutes like before.

3.3.2 Set-up 2D and 3D-GEM

The FLOW-output was provided for a 2D-GEM model (existing of one vertical layer), and for a 3D-GEM model (existing of five layers). Apart from the hydrodynamics, also the monitoring points and groups had to be redefined as to include the third dimension. Furthermore, the forcing functions had to be adjusted, and various processes related to the vertical dispersion had to be activated. Also, in the 3D-model, the horizontal dispersion coefficient was set to 1.

3.3.3 Set-up grazers

For the model runs in this report, only V1-morphs were used. Runs were performed with one type/species of generalized grazer, and with three species of grazers.

3.3.4 Forcing functions

Meteorological conditions
Meteorological conditions for 2002 were obtained from KNMI, station Wilhelminadorp.

Nutrient loads
The polder loads and locations were obtained from the 1D Deltamodel (Meijers et al 2007). Boundary concentrations were derived from the GEM-model for the Southern North Sea (Los et al, 2008) using a nesting procedure. The chosen boundary-settings led to a too large inflow of fresh and nutrient-rich water from the Rhine into the Oosterschelde. This has been corrected by adjusting the incoming salinity- and nutrient concentrations through the north-western boundary of the model.

At location Steenbergen, no measured silica-concentrations were available. Therefore, the incoming silica-concentrations at the Krammersluices were based on silica-concentrations at other discharge locations.

Inorganic matter
The function for inorganic matter was based on measured concentrations, and describes a gradient from higher concentrations in the west to smaller concentrations in the east (Figure 3.2).
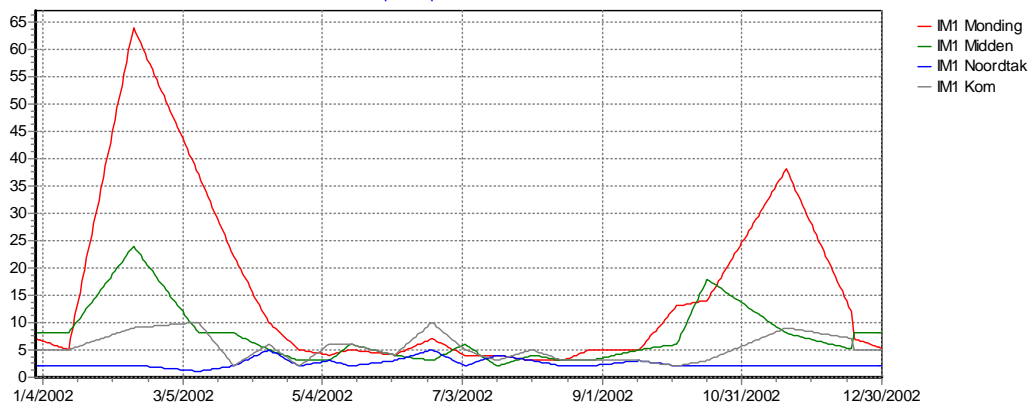
Figure 3.2 Inorganic matter as forced in each of the compartments for 2002.

Phosphate adsorption and release flux from the sediment
The adsorption and release flux of phosphate was based on the function as used the 1D Deltamodel (Meijers et al 2007), but calibrated to better fit the measured PO4-concentrations (see Figure 3.3). This flux is equally applied to the whole bottom area.
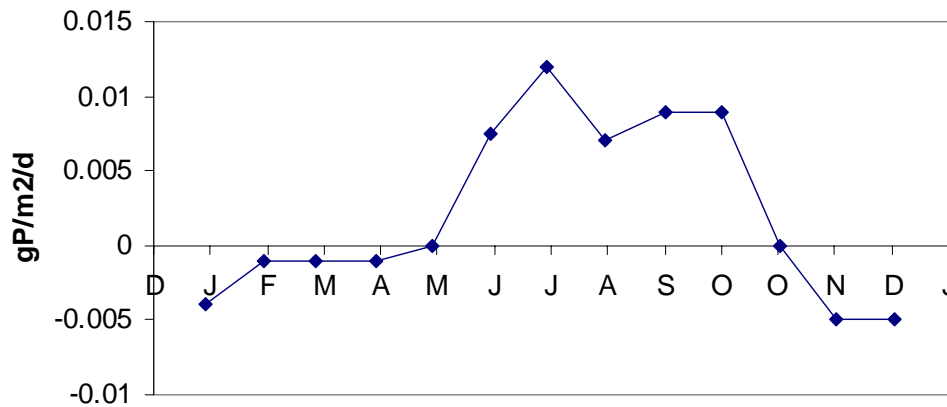
Fig 3.3. Phosphate adsorption and release from the sediment

3.3.5 Parameterisation and calibration

Chemical parameter values and settings
Chemical parameter values and settings in the Oosterschelde model are based on those as used in the model of lake Grevelingen (Nolte et al, 2008).

Background extinction
The background extinction was decreased from 0.2 to 0.08. This leads to an overall increase in primary production. However, an additional effect is that light limitation will come to play a smaller role in shallow locations, and other (volume-related) factors such as nutrients will become more important. As a result, primary production will become more closely related to depth: deeper areas will have more primary production than shallow areas. This leads to an improved fit with measured primary productions.

Physiological parameter values for grazers
Physiological parameter values for the various grazer species are derived from (Van der Veer et al., 2006), see Table 2.2.

Initial biomasses, distribution, and mortality rates
Initial grazer biomasses were based on the estimated shellfish stocks in literature (Geurts van Kessel et al., 2003) and are updated by personal communication (Wijsman, 2010). These include biomasses of mussels, cockles, sublittoral and littoral 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'). An extended summary of all initial values can be found in appendix C.

Cockles were restricted to areas on or close to mudflats (areas with a depth < 2m), see Figure 3.4A. Mussels were located on dedicated mussel farms, which were appointed in the schematization on basis of Figure 1.1 (see Figure 3.4B). Sublittoral and littoral oysters were restricted to areas with a depth<1, and littoral oysters to areas with a 15<depth<1 (see Figure 3.4C).

Wet-weights were converted into AFD-weights by multiplying them with the (species-specific) afdw/ww-ratio. The AFD-weights (kg) were converted into grams by multiplying them by a factor 1000, 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. Initial, intermediate and resulting values can be found in appendix C.

The densities were calculated per species by dividing their total biomass per compartment by the total surface area at which they occur (see Figure 3.4). The total initial density for a generalized grazer was derived by summing the densities of the separate species, resulting in an initial distribution as shown in Figure 3.5.

Half-saturation constants

Half saturation constants for cockles and mussels in the Oosterschelde were derived from field measurements on individual growth curves and environmental conditions (Troost et al, 2010). They were converted in units of mgC/l by multiplication with the C/Chl-ratio 0.078 mgC/ugChl.

For oysters in the Oosterschelde, no suitable field measurements are available. Therefore, the half-saturation constant was calibrated such that the oyster population in the model remained more or less stable (i.e. that after a year, the structural volume returns to its initial value) . The resulting value (Xk= 5 mgC/l) is about twice as high as the values for cockles and mussels, and lies within in the range as reported in literature for oysters in other systems (Pouvreau et al. 2006, Bourles et al. 2009, Bacher et al. 2006). Also, the value is supported by data from flume-experiments (Van Duren & Troost, 2010), see the tentative derivation in Appendix D.

Algal energy content

The algal energy content couples the grazer module to its environment. It determines how many algae are needed by the shellfish to fulfil their energy requirements. Its value is not shellfish-species specific, but algal-species specific. For the time being, however, we use a single value for all algal species. It was calibrated such that the growth of the grazer populations is in the right order of magnitude. The resulting value (10^{-5} J/mgC) is around 10 times higher than its original setting to increase the grazing pressure and as such compensate for the decrease in background extinction and the related higher primary production (see above). The new value lies well within the range of values as derived from data from flume-experiments (Appendix D).

Reference lengths

After the first rough calibration of the grazer growth with the algal energy content (see above), the reference lengths of the grazers are adjusted such that the populations become stable. 'Stable' means that after a year the structural biomasses (V) return to their initial values.

Initial energy reserves (E) and gonads (R)

Initial values for energy reserves and gonads were chosen such that they are more or less stable, i.e. have returned to their initial values at the end of the year.
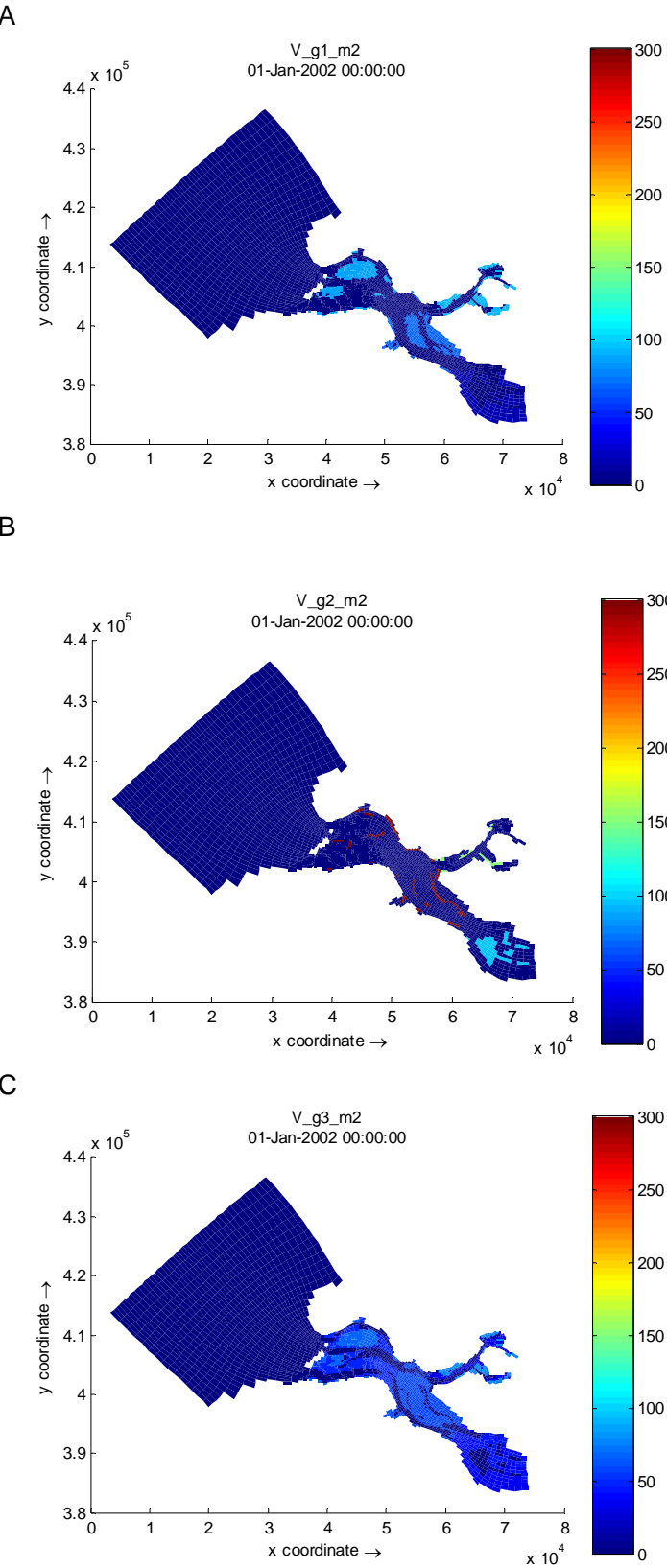
Figure 3.4. Initial densities (gWW/m2) of cockles (A), mussels (B), and oysters (C).
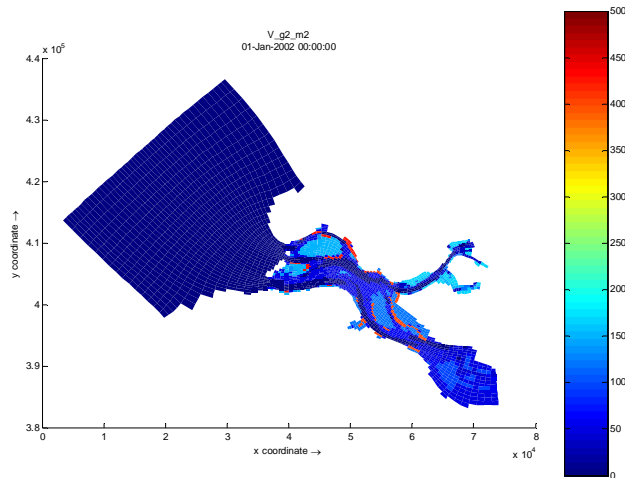
Figure 3.5. Initial distribution (structural volume gWW/m2) for a generalized grazer.

3.3.6 Validation method and measurements

Validation was carried out by visual comparison of model results with field measurements. A goodness of fit measure was not used, since in this stage a visual comparison was generally sufficient to determine which model variant performed best. Measurements used for validation are nutrient and chlorophyll concentrations, and primary production.

Nutrient and chlorophyll concentrations
For nutrient and chlorophyll concentrations, measurements are available for the locations Wissenkerke, HammenOost, Lodijkse gat, and Zijpe (Dutch monitoring program, MWTL).

Primary production
In some years, primary production of the phytoplankton has been measured by the NIOO-CEME at a bi-monthly interval at five locations in the Oosterschelde: Lodijkse Gat (east), Zandkreek (central), Zeelandbrug (central), Roompot (west), and the "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). Unfortunately, no measurements are available for 2002. Instead, measurements of 2000 were used to validate the GEM. This is acceptable since the measured primary productions show only relatively small differences between adjacent years, which are clearly smaller than the differences between compartments (Geurts van Kessel 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.

3.3.7 Model assumptions and limitations

Only suitable to simulate short periods

The Oosterschelde GEM is not yet suitable to simulate long time series (several sequential years). This is mainly due to the fact that the relation between spawning and recruitment of shellfish success is unknown. One way to overcome this problem would be to restart the simulations each year from measured shellfish biomasses.

Stability grazer populations

Exact developments of the existing grazer populations are unknown. Therefore, they are assumed to be stable. This means that after a year their structural biomasses (V) more or less return to their initial values.

Structure grazer populations

In the model runs described in this report, the grazers are modelled using a V1-morph approach. This is only valid if the population composition (size and/ or age distributions) remains more or less stable. Also, the modelled grazers are not size- or age structured. However, these choices do not seem to have much effect on the system behaviour (Troost 2009). Still, isomorphic or structured models may be preferred when population structure or individual growth needs to be studied in more detail.

Timing of sowing and harvesting

In the model runs so far, it is assumed that throughout the whole year a fixed fraction of the populations is being harvested. Also, the sowing of the mussels takes place continuously and with a fixed fraction. As such, the model was kept simple in order to facilitate calibration and analysis. In reality, however, sowing takes place in spring, and harvesting in fall. In future these refinements may be included in the model.

Other grazers

So far, the grazer module has been parameterised only for oysters, mussels and cockles. Parameters for other grazers, such as zooplankton species and ensis, which are also present in the Oosterschelde, are not fully known, and these grazers are at best only implicitly included in the current model. With 'implicit' it is meant that the model is calibrated to measurements that are unavoidably affected by such grazers. Through its parameterization, the model thus implicitly takes these grazers into account.

Selective grazing and pseudofaeces production

The model does not yet include selective grazing. Selective grazing may lead to the dominance of certain algal species or sizes, and may thus affect food availability, primary production and overall system behaviour. Furthermore, selective grazing may cause additional pseudofaeces production, which also may affect the system's nutrient recycling. Like selective grazing, pseudofaeces production is not yet explicitly included in the model.

3.4 Oosterschelde GEM results and validation

First, a generalized grazer is included into the integrated ecosystem model of the Oosterschelde. The generalized grazer is constant in size (V1-morph), and its parameter values are based on those of mussels (Section 2.4).

3.4.1   Results for a generalized grazer

Results are compared to the results of the integrated ecosystem model without grazers. Figure 3.6 shows the total population size (gC) throughout the year in the various compartments.
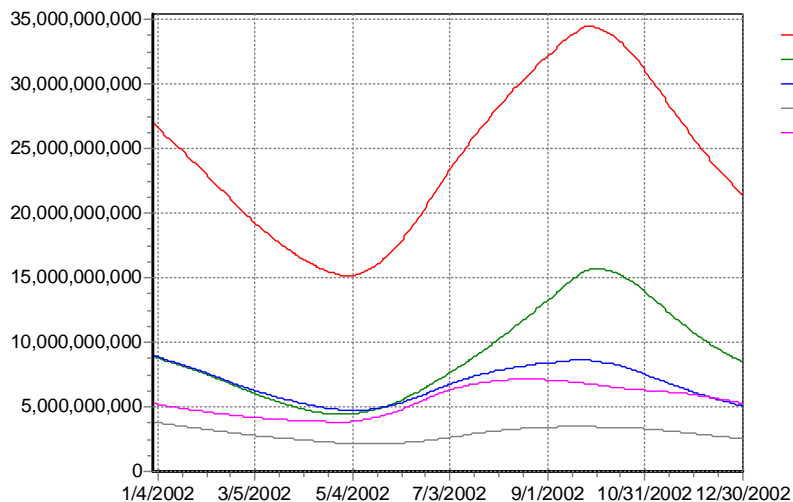
Fig 3.6. Total shellfish population sizes (gC) in each of the compartments: Western (green), Central (blue), Northern (grey), Eastern (pink), and in the Oosterschelde as a whole (red).

Validation by nutrient concentrations
In appendices A.2 to A.5, the nutrient concentrations from the GEM including grazers are compared to those resulting from the GEM without grazers, and to observed nutrient concentrations.

Visual inspection of these figures shows that the overall model fit improves considerably due to the grazers, especially at the eastern (Lodijkse gat) and northern locations (Zijpe). Chlorophyll concentrations in western (Wissenkerke) and central locations (Hammen Oost), however, change only slightly, and their fit with observations slightly deteriorates when including grazers. Nutrient concentrations do not depend much on whether gazing is included or not.

In comparison to the results for 1998 (Troost 2009), however, the algal bloom now starts too late, especially in the western and central compartments. This is possibly due to the (forced) concentrations of inorganic matter, which remain high until May. Including variation in this forcing function may induce earlier blooms, which may also lead to an improved fit of nutrient concentrations during this period.

On the other hand, the fit of phosphate concentrations has improved considerably due to the included adsorption and release flux of phosphate.

Validation by primary production
Figure 3.7 shows measured and simulated annual primary productions. In almost all compartments, the simulated primary productions are too large when grazing is not taken into account (Figures 5.7 and 5.8). Especially in the northern and eastern compartments primary

productions are overestimated. When including grazing, primary production substantially decreases, and the overall fit improves considerably.

When comparing results for 2002 to those of 1998 (Troost 2009), some differences become apparent. Thanks to a recalibration of parameter settings, the fits of the 2002-model (including grazing) have improved considerably when compared to the 1998-model. This is especially the case in the northern and eastern compartment, for which the primary productions for the 1998-model were still various orders of magnitudes off. A side-effect, however, is that while the 1998-model predicted an overall increase in primary production due to grazing, the 2002-model predicts an overall decrease. This suggests that the existing grazer pressure in the Oosterschelde may be larger than was predicted before.

Figure 3.7. Primary production in each of the four compartments of the Oosterschelde, measured (blue) and simulated by the model with (purple) and without grazing (yellow).

3.4.2 Results for three different grazer species

Figure 3.8 shows the total population size of each of the three species when these are simulated simultaneously. The figure shows that, with the current settings, the populations are not entirely stable: the cockle population shows a net decrease over the year, while the mussel population shows a net increase. When simulating several sequential years, this will probably lead to extinction of one or two of the simulated species, which is a known phenomenon in deterministic modelling called competitive exclusion. If desirable, the population stabilities could be improved by changing the reference lengths of these two species.
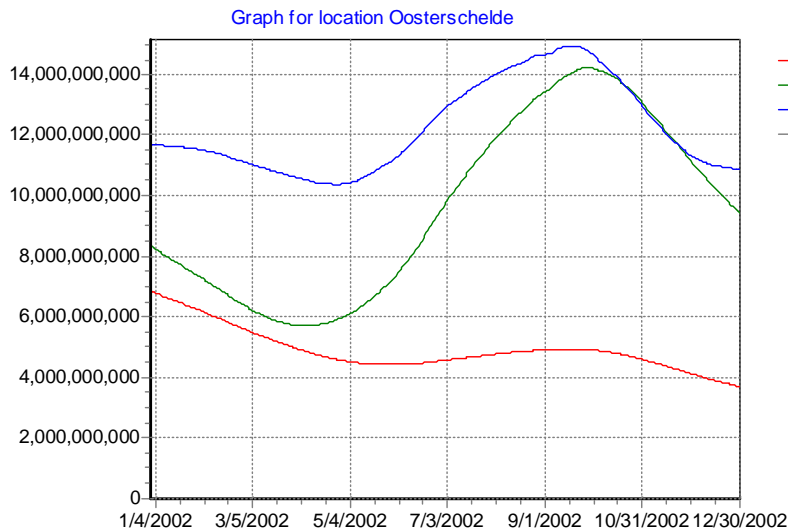
Fig 3.8 Total population sizes of oysters (blue curve), mussels (green curve), and cockles (red curve) in the Oosterschelde.

Validation by nutrient concentrations

In appendices A.2 to A.5, the nutrient concentrations from the GEM including three different grazers are compared to those resulting from the GEM with a generalized grazer.

These figures show that the choice for a generalized grazer or three different species of grazers hardly affects the resulting nutrient concentrations. Only in the eastern compartment, grazing pressure is slightly decreased because oysters have a larger half-saturation constant than generalized grazers. This is in line with previous results (Troost, 2009).

Validation by primary productions

In Figure 3.9, the primary productions from the GEM including three different grazers are compared to measurements, and to results from the GEM with a generalized grazer. The figure shows that predicted primary productions hardly deviate between the two approaches.

The choice for including either a generalized grazer of for three different species of grazers will thus mainly depend on the level of detail required to answer a specific question.

Deltares
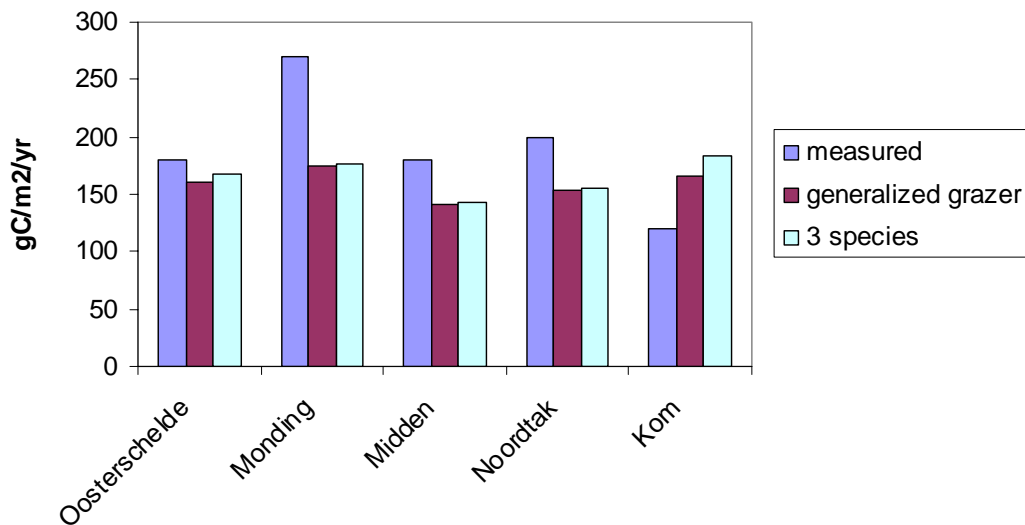
Fig 3.9. Primary production in each of the four compartments of the Oosterschelde, measured (blue) and simulated using a generalized grazer (purple) and three species of grazers (light blue).

3.4.3 Comparison 2D and 3D

Appendices A.6 to A.9 shows the nutrient concentrations resulting from the 3D-GEM model as compared to the 2D-model.

Resulting nutrient concentrations suggest that the differences between the 2D-model and the 3D-model are not very large. Yet, the nutrient and chlorophyll concentrations (especially the latter) show more variation and lead to some improvement of the fits, especially in location Zijpe. Probably, this is related to the vertical gradient, showing higher concentrations at the surface where field measurements are taken. The 3D-model is capable of reproducing this vertical gradient, and has monitoring locations in the surface layer, while the results from the 2D-model merely represent a vertically averaged value for the whole water column. This is supported by the fact that especially in location Zijpe (where the improvement is most apparent), the vertical gradient is large due to the inflow of freshwater through the Krammersluices.

Figure 3.10 gives an impression of the vertical gradient occurring in chlorophyll concentrations. It shows the chlorophyll concentrations over the water column along a (west to east) cross-section of the Oosterschelde. Vertical gradients in chlorophyll concentration are most apparent in the eastern part, which has higher shellfish densities. The vertical gradient may be caused by physical stratification and/or light extinction, but may be enhanced due to the spatial segregation of shellfish (in the bottom layer) and algae (in the surface layers), which may favour algal growth in the surface layers.
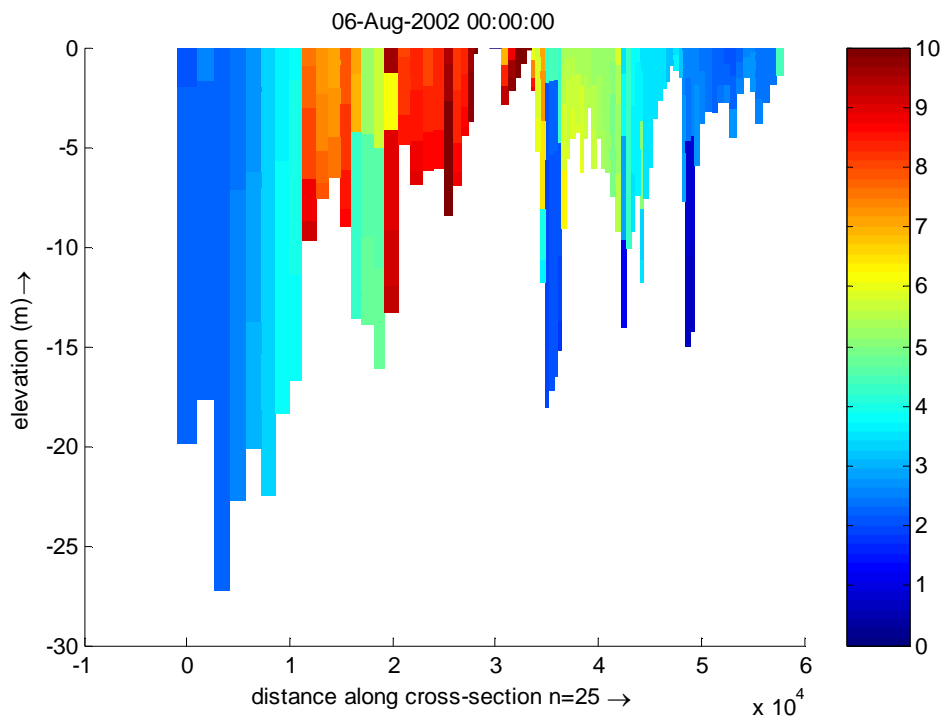
Figure 3.10. Chlorophyll concentrations (ug/l) over depth along a cross-section of the Oosterschelde system running from west (left) to east (right).

Figure 3.11 shows the annual primary productions as measured and as predicted by the 2D and 3D-models. The 3D-model predicts a higher primary production in almost all compartments, leading to a better fit with the measurements than the 2D-model. Apart from this quantitative improvement, the 3D-model also leads to a qualitative improvement, since it correctly reproduces productions to be larger in the northern compartment than in the eastern compartment. This is the other way around in primary productions predicted by the 2D-model. The better performance of the 3D-model can be explained by the fact that 3D-models are knowingly more capable of simulating water transports especially in shallow areas.

These results may suggest that the 3D-model is to be preferred over the 2D-model. However, the larger computation time of the 3D-model should also be taken into account. This is 5 times as large as the computation time of the 2D-model (because of the 5 vertical layers), which comes down to a total computation time of around 4 days. The choice for 2D- or 3D-modelling will thus also depend on the level of detail that is required to answer the question at hand.
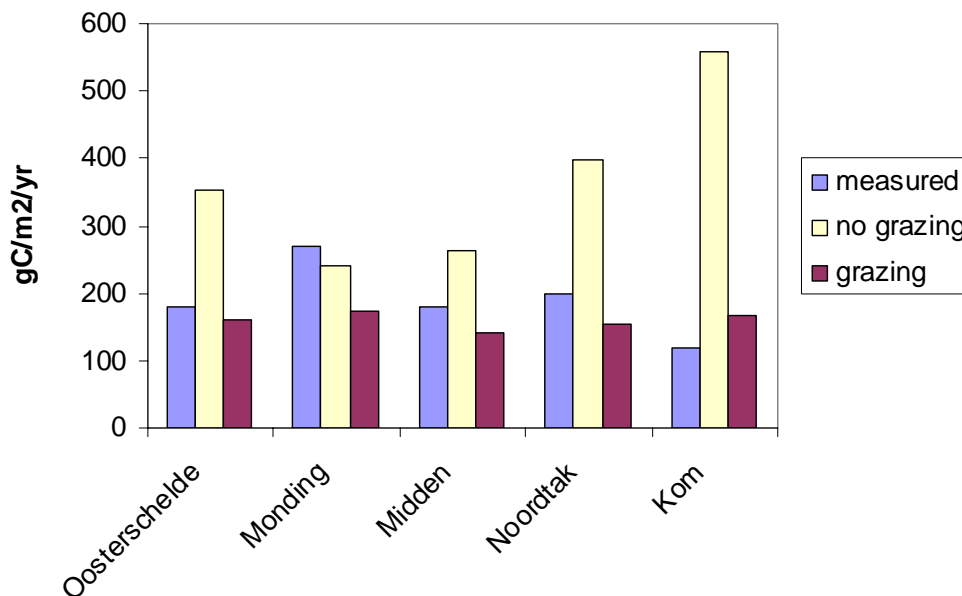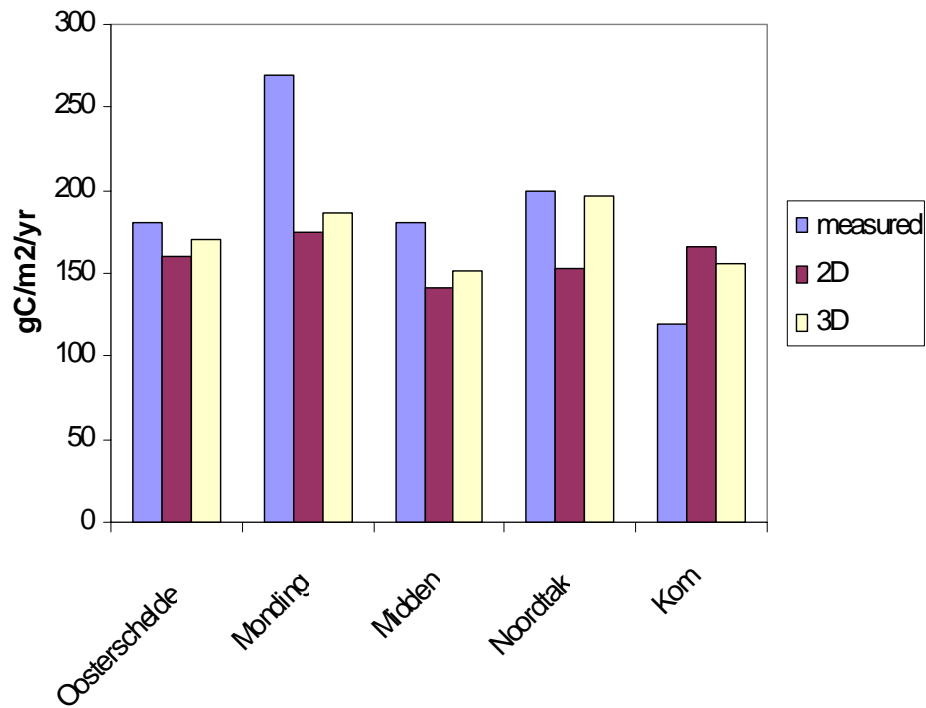
Fig 3.11 Primary production in each of the four compartments of the Oosterschelde, measured (blue) and simulated using the 2D-model (purple) and the 3D-model (yellow).

4 Case study

4.1 Objective and scope

Various autonomous negative trends are taking place in the Oosterschelde system. These trends are being studied within the project framework called 'ANT-Oosterschelde'. One of these trends is the ongoing erosion of intertidal flats, often referred to as 'sand hunger'. It is caused by the Deltaworks, which have changed the morphological steady state of the Oosterschelde. Below, we apply the Oosterschelde-GEM described above to study the effects of the changing bathymetry on the system's primary production.

Note that only *direct* effects of the changing bathymetry on primary production were investigated; *indirect* effects such as a corresponding change in (initial) shellfish distributions were not taken into account.

Furthermore, the reader is reminded that the hydrodynamic model underlying the GEM has only been roughly validated, and should be considered a 'proof of concept' model. Since this case study focuses on the effects of small bathymetric (and thus hydrodynamic) changes, its results should be interpreted with caution.

4.2 Approach

To study the effects of mudflat erosion on primary production, the model bathymetry was adjusted. Resulting primary productions were compared to those of a reference run with the default bathymetry.

Since the effects of mud flat erosion on primary production were expected to be rather small, it was chosen to induce a relative extreme change (-1 m.) in the intertidal depths, which lies in the order of magnitude that is thought to occur some 100 years from now.

More precisely, all grid cells of less than 1 meter depth were lowered by 1 meter. Cells deeper than 3 meters were left unaffected. Locations between 1 and 3 meters depth were lowered by an intermediate amount (0 - 1 m.), inversely proportional to their depth, such that the resulting gradients remained smooth. Default and case study bathymetries, as well as their differences, are shown in Figure 4.1.

4.3 Methods

For this case-study, the Oosterschelde GEM was run in 3D-mode to optimally simulate transport and fate of substrates in shallow areas. Also, the artificial volume-adjustments that were required to deal with too large flow-velocities (Section 3.3.1) were avoided, since these adjustments are often specifically applied to shallow areas such as the mudflats and lead to unreliable results in those areas.

Avoiding artificial volume-adjustments was done by using integration routine 22, which can deal with large flow-velocities by switching from explicit integration methods to implicit ones. To use this new integration routine, the grazer-module had to be updated and made compatible with the latest Delwaq version (Section 3.3.1).
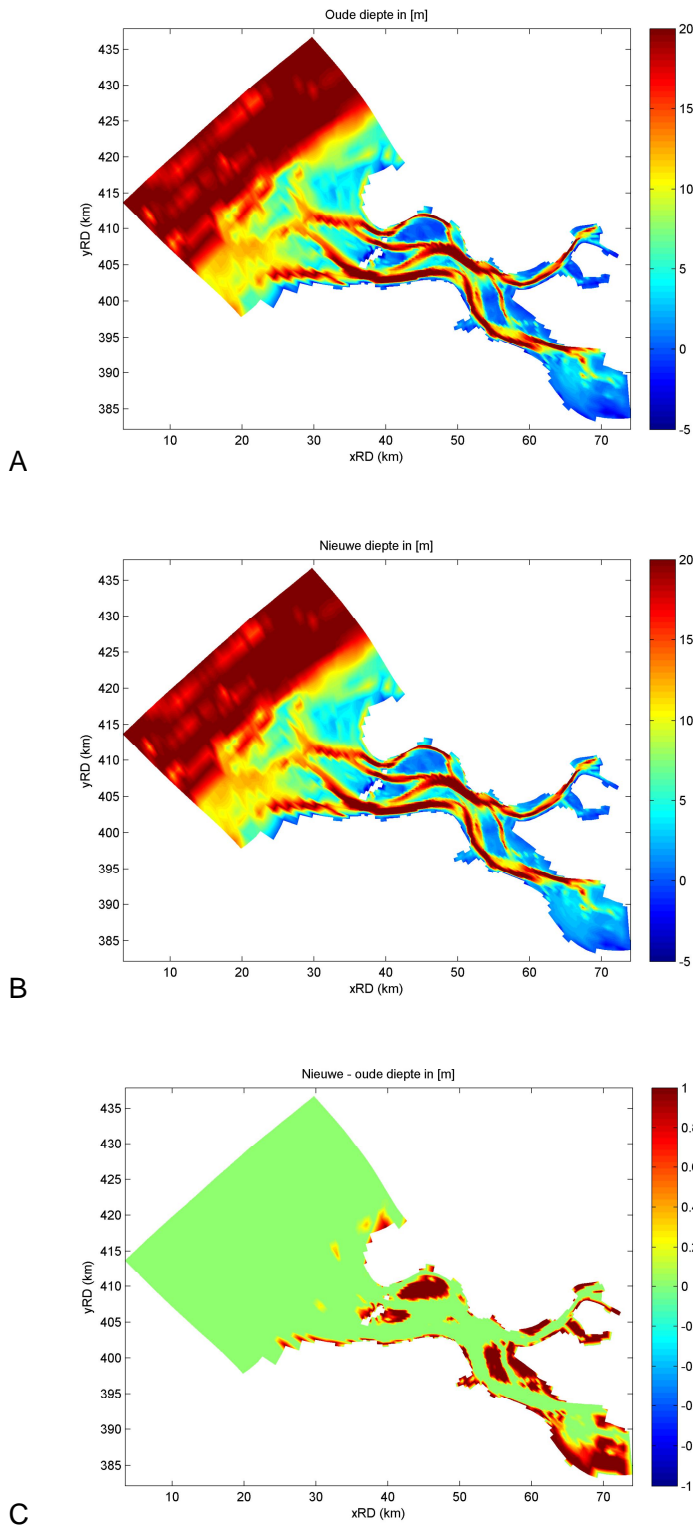
Figure 4.1. The default Oosterschelde bathymetry in the base model (A) and the bathymetry in the scenario study (B) with eroded mudflats, as well as the depth-differences between the two (C).

4.4 Results

Appendix A.10-13 shows the nutrient and chlorophyll concentrations resulting from the 3D-GEM base case as compared to those from the 3D-GEM case study. The figures show that the differences between the base case and the case study are negligible. This suggests that changes in bathymetry do not very much affect these concentrations.

Figure 4.2 shows the primary production in the base case and in the case study. Apparently, an increased depth of the intertidal mudflats leads to a small increase (~9%) in primary production in all compartments. However, the increase in primary productions is quite small, and falls within the range of differences between measured and modelled values (modelling error).

The increase in primary production may be explained by the fact that, in shallow areas, primary production is not light-limited. Instead, it may be limited by nutrients or other 'volume-dependent' factors. If so, any increase in depth in shallow areas will lead to an increase in primary production.

Results, however, show that the relative increase in primary production (~9%) is larger than the relative increase in depth or volume (~2.5%). This suggests that also other (feedback) processes may be involved, and/or that the nonlinearity of the processes may play a role.
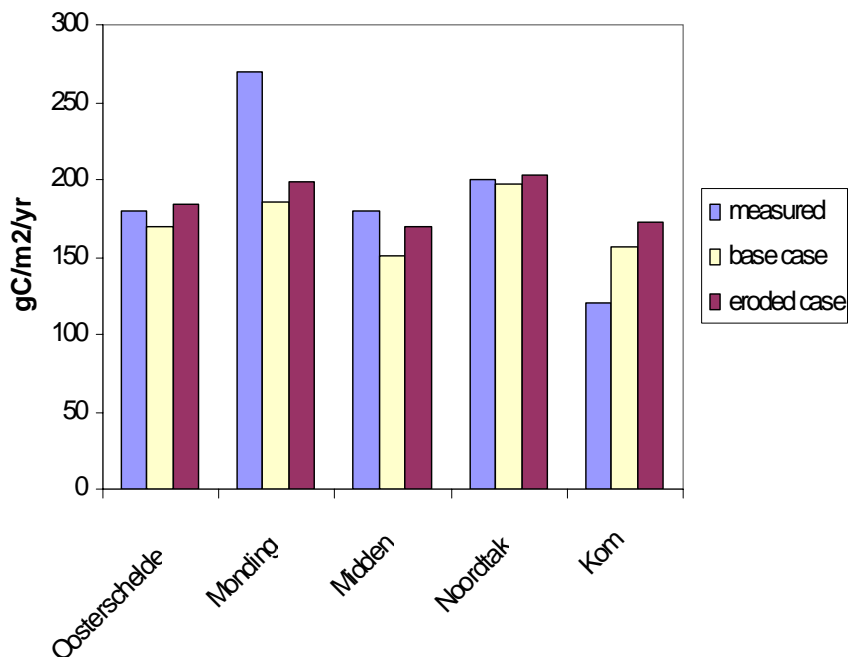
Figure 4.2. Primary production in each of the four compartments of the Oosterschelde, measured (blue) and simulated with the default bathymetry (yellow) and the eroded bathymetry (purple).

5 Conclusions

The integrated ecosystem model for the Oosterschelde was improved in several ways. First, it was adjusted to simulate a more average and recent year (2002). Then, various parameters were recalibrated and a forcing function for the phosphate adsorption was included. Initial grazer settings were updated, standardized and documented. Also, some of the system system-specific parameter values were validated on basis of data from flume-experiments. Finally, the model was made compatible with newest Delwaq version.

Modelled primary productions now show an acceptable fit with measurements. In comparison to previous results, however, the fit with measured nutrient and chlorophyll concentrations has somewhat deteriorated. Also, the predicted grazer pressure in the Oosterschelde is larger than was calculated before.

In line with previous results, including grazing into the model greatly improves the model performance. Hereby, it does not seem to affect the system behaviour much whether one generalized grazer is included in the model or three separate species of grazers. The choice for either one of these approaches will thus depend on the amount of detail required to answer a specific question.

The 3D-model seems to perform better than the 2D-model, but also requires a larger computational time. The choice for 2D- or 3D-modelling will thus also depend on the level of detail that is required to answer the question at hand.

Although the hydrodynamic part of the Oosterschelde GEM was set up as a 'proof-of-concept' model, the GEM was applied to study the effects of long term mudflat erosion on the primary production of the system. Results suggest that the nutrient and chlorophyll concentrations are hardly affected by the disappearance of the mudflats, but that the primary production in the system may slightly increase. The analysis however only includes direct effects of erosion, and the results fall within the range of modelling errors.

5.1 Recommendations

The integrated ecosystem model for the Oosterschelde and/or the grazer module could be improved in many ways. Recommended improvements are the following:

Hydrodynamic model
The hydrodynamic (FLOW) model should be better validated and, if necessary, improved.

Ecosystem model
The ecosystem model could be improved with respect to the timing of the bloom.

Selective grazing and pseudofaeces production
The model could be extended with (active and passive) selective grazing and pseudofaeces production. Selective grazing may lead to the dominance of certain algal species or sizes, and may thus affect food availability, primary production and overall system behaviour. Furthermore, selective grazing may cause additional pseudofaeces production, which also may affect the system's nutrient recycling.

Long term simulations
As yet, the model is suitable for short-term simulations (up to one year) or for comparative case studies. To be able to simulate a longer time-series (several sequential years), input

data for all simulated years are required. Furthermore, a relation between spawning and recruitment and other (environmental) factors would have to be included into the grazer-module. Since this relation is largely unknown, this would require a substantial modelling effort in which the relation and/or underlying processes have to be calibrated by long-term data on various population stocks.

6 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.

Bourlès, Y, Alunno-Bruscia M., Pouvreau, S., Tollu, G., Leguay, D., Arnaud, C., Goulletquer, P., Kooijman, S.A.L.M., 2009. Modelling growth and reproduction of the Pacific oyster Crassostrea gigas: Advances in the oyster-DEB model through application to a coastal pond. Journal of Sea Research 62: 62-71.

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 applicability of different biogeochemical model versions. Journal of Marine Systems AMEMR 2008 Special Issue.

Los F. J., M.T Villars, and M.W.M. Van der Tol. 2008. A 3-dimensional primary production model (BLOOM/GEM) and its applications to the (southern) North Sea (coupled physical–chemical–ecological model). Journal of Marine Systems 74 (2008) 259–294.

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.

Troost, T.A. 2008. Modelling carrying capacity. Report Deltakennis z4574.30, Deltares.

Troost, T.A. 2009. Modelling carrying capacity. Report Deltakennis 1200314.001, Deltares.

Troost, T.A., Wijsman, J.W.M., Saraiva, S., Freitas, V. 2010. Modelling shellfish growth with dynamic energy budget models: an application for cockles and mussels in the Oosterschelde (southwest Netherlands). Phil. Trans. R. Soc. B. 365:3567-3577.

Troost T.A., and van Duren, L.A. 2010. Draagkracht voor MZI's in de Oosterschelde. Deltares rapport 1203038.

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

Duren, L.A. and K. Troost. 2010. Feedback processes on ecosystem functioning by native and invasive shellfish species. Submitted.

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.

WL | Delft Hydraulics 2002. GEM documentation and user manual, WL | Delft Hydraulics report Z3197, Delft, The Netherlands.

WL | Delft Hydraulics 2003. Delft3D-WAQ users manual., WL | Delft Hydraulics, Delft, The Netherlands.

WL | Delft Hydraulics 2005. Delft3D-WAQ technical reference manual. Detailed description of processes, WL | Delft Hydraulics, Delft, The Netherlands.

WL | Delft Hydraulics 2006. Delft3D-WAQ users manual. Simulation of multi-dimensional hydrodynamic flows and transport phenomena, including sediments. WL | Delft Hydraulics, Delft, The Netherlands.

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) at four locations in the Oosterschelde in the surface layer of the 2D GEM model (blue curve) and the 3D GEM model (green curve), compared to measurements (red crosses).
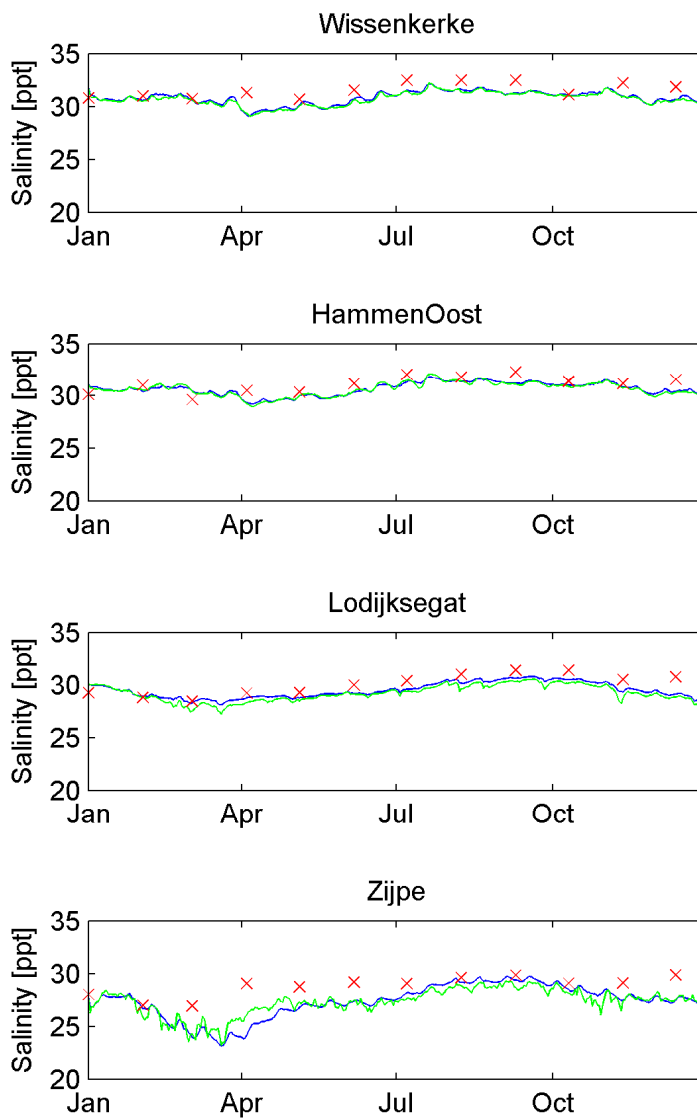
A.2   Chlorophyll: simulated Chl-concentrations of the 2D GEM without grazers (green curve) , with a generalized grazer (blue curve) , and with three species of grazers (red curve) at four locations in the Oosterschelde. Red crosses are measured concentrations.
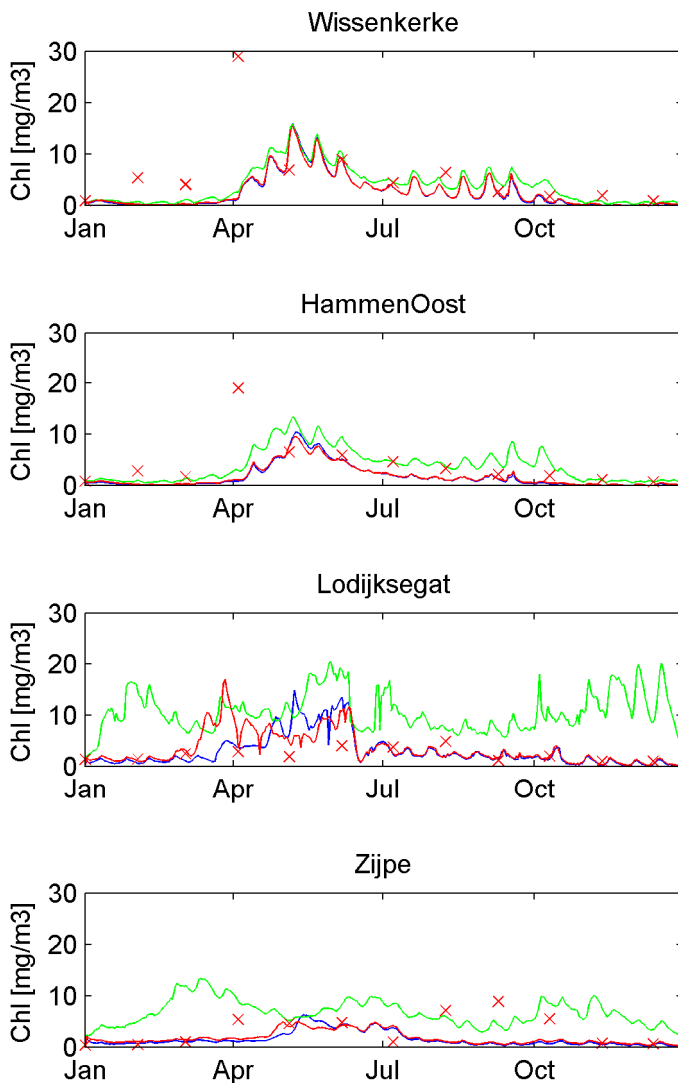
A.3 **Nitrate: simulated NO3 concentrations of the 2D GEM without grazers (green curve) , with a generalized grazer (blue curve) , and with three species of grazers (red curve) at four locations in the Oosterschelde. Red crosses are measured concentrations.**

Deltares
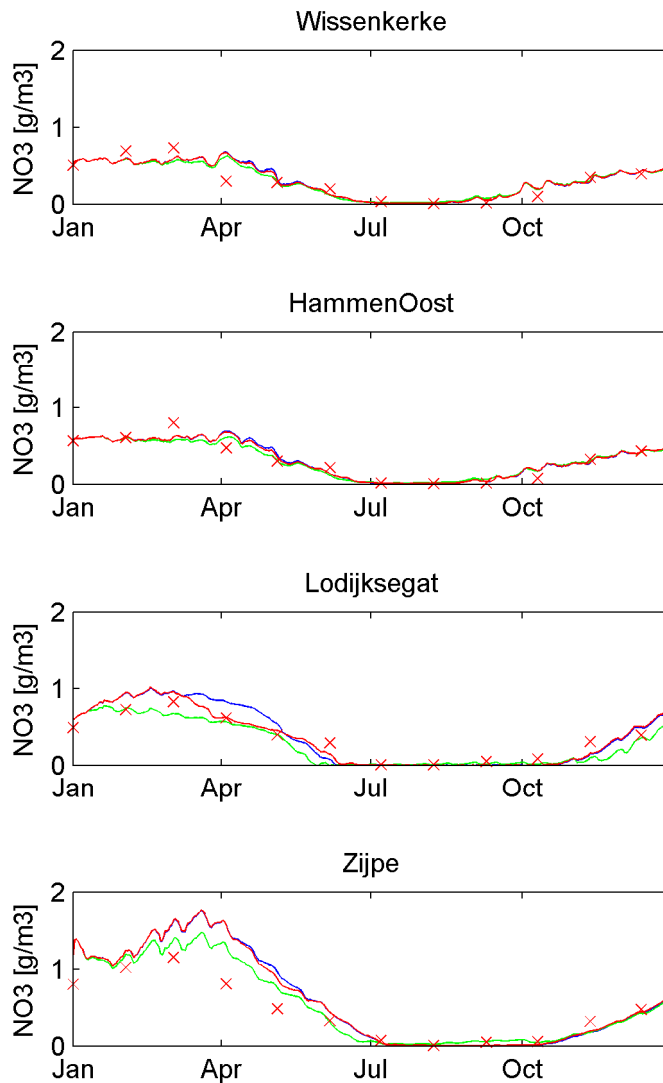
A.4 Phosphate: simulated PO4 concentrations concentrations of the 2D GEM without grazers (green curve) , with a generalized grazer (blue curve) , and with three species of grazers (red curve) at four locations in the Oosterschelde. Red crosses are measured concentrations.

Wissenkerke

HammenOost

Lodijksegat

Zijpe

Modelling the carrying capacity of the Oosterschelde
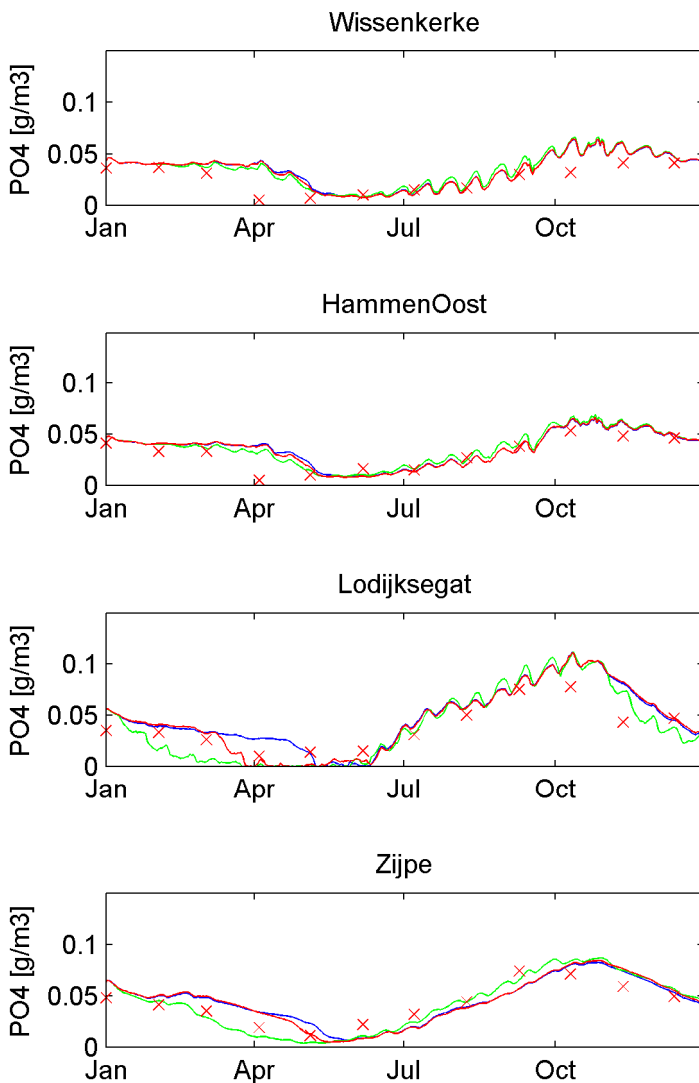
A.5 Silicate: simulated Si concentrations of the 2D GEM without grazers (green curve), with a generalized grazer (blue curve), and with three species of grazers (red curve) at four locations in the Oosterschelde. Red crosses are measured concentrations.
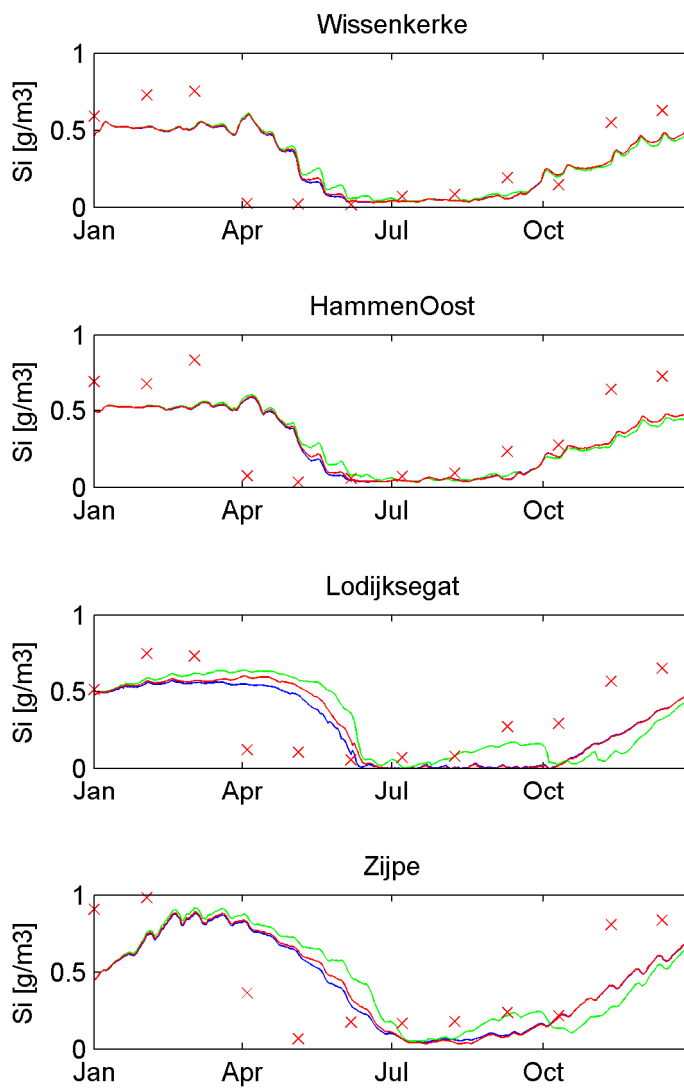
A.6 **Chlorophyll: simulated chlorophyll concentrations of the 2D GEM (blue curves) and of the 3D GEM (green curves) both with a generalized grazer, at four locations in the Oosterschelde. Red crosses are measured concentrations.**
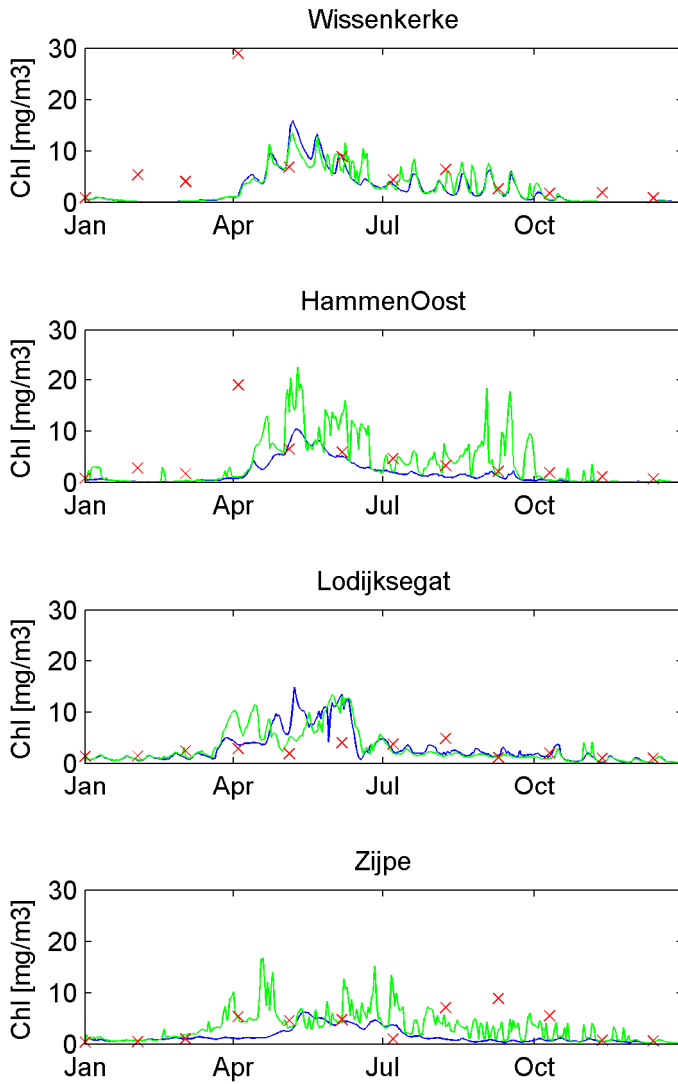
A.7 **Nitrate: simulated NO3 concentrations of the 2D GEM (blue curves) and of the 3D GEM (green curves) both with a generalized grazer, at four locations in the Oosterschelde. Red crosses are measured concentrations.**

A.8   Phosphate: simulated PO4 concentrations of the 2D GEM (blue curves) and of the 3D GEM (green curves) both with a generalized grazer, at four locations in the Oosterschelde. Red crosses are measured concentrations.

Wissenkerke

HammenOost

Lodijksegat

Zijpe

Modelling the carrying capacity of the Oosterschelde
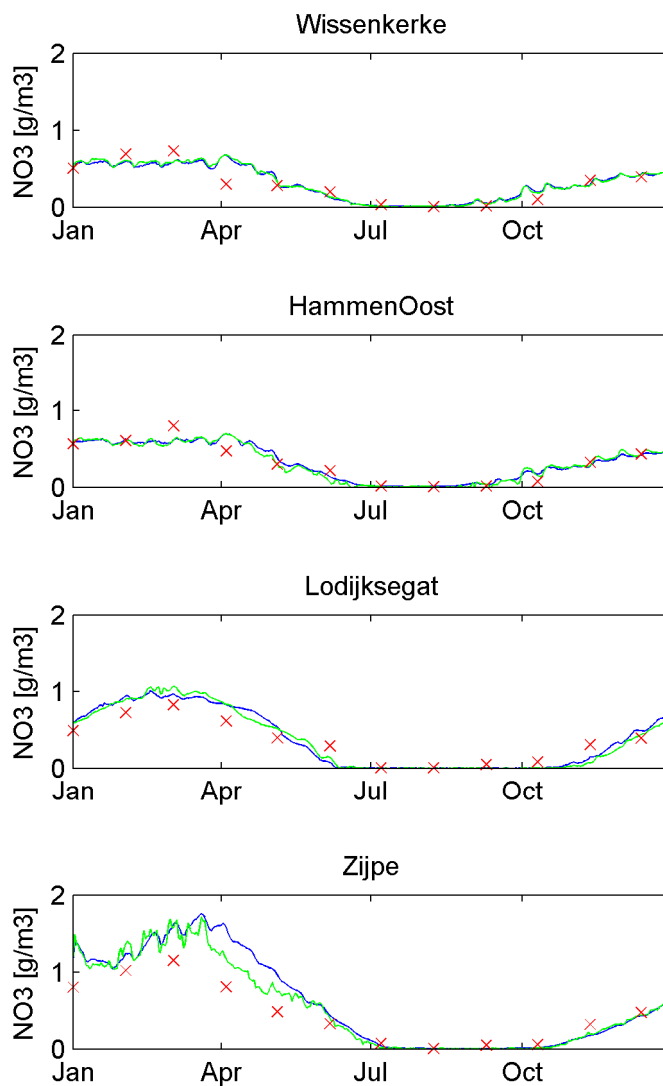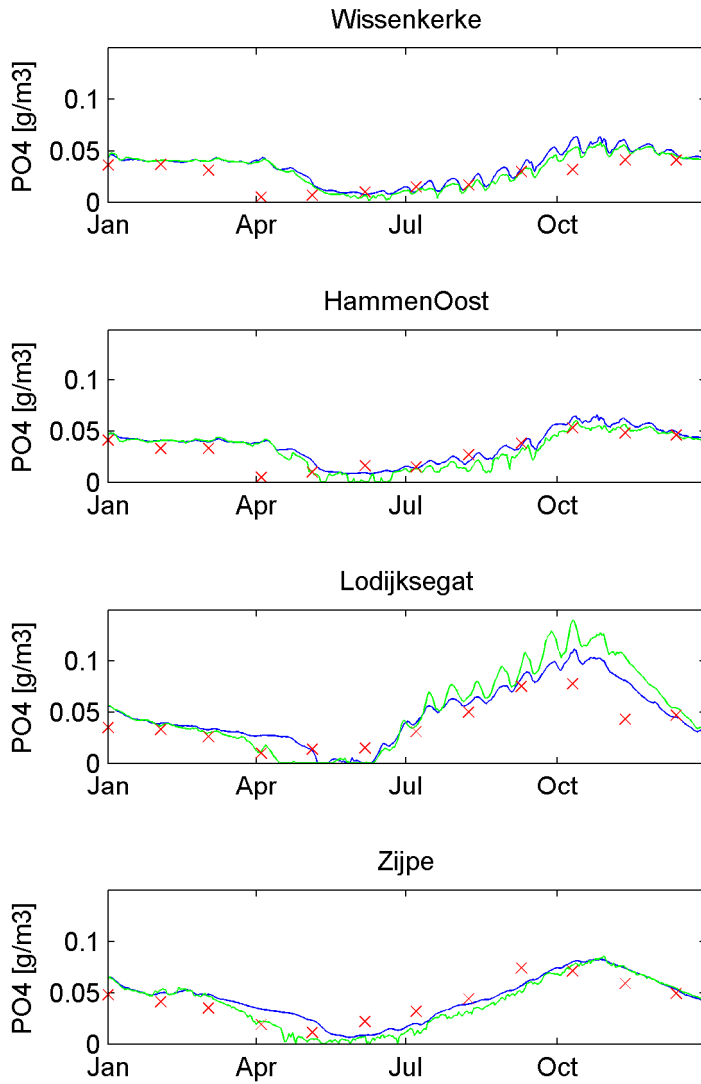
A.9 Silicate: simulated Si concentrations of the 2D GEM (blue curves) and of the 3D GEM (green curves) both with a generalized grazer, at four locations in the Oosterschelde. Red crosses are measured concentrations.
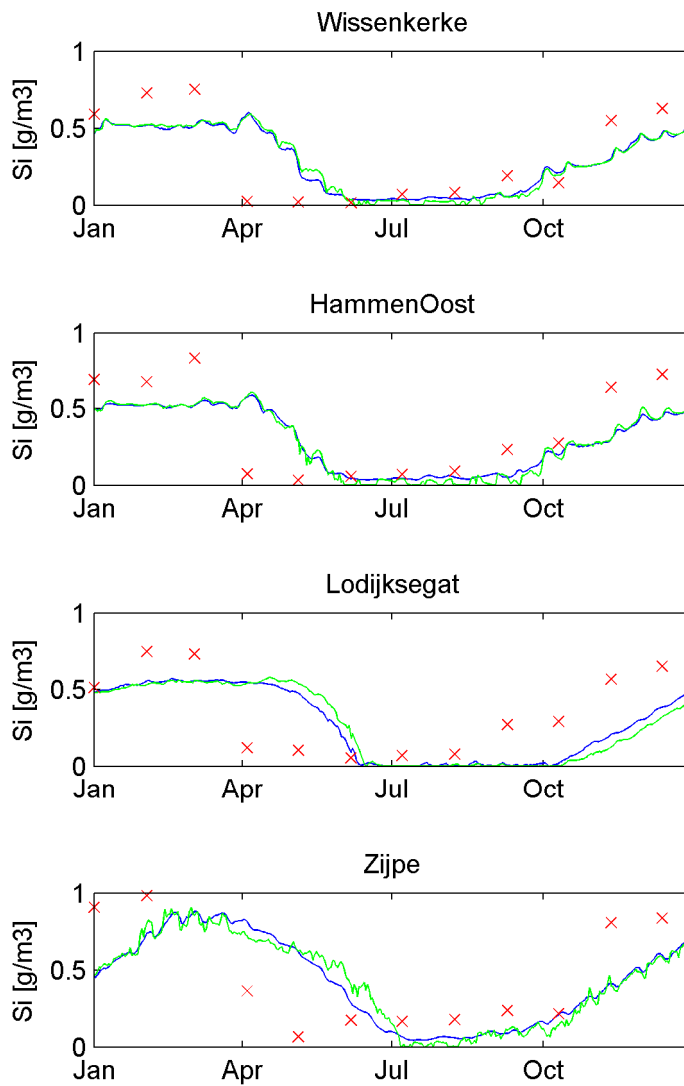
A.10   Chlorophyll: simulated chlorophyll concentrations of the 3D GEM base case (blue curves) and for the 3D GEM case study on erosion (green curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.
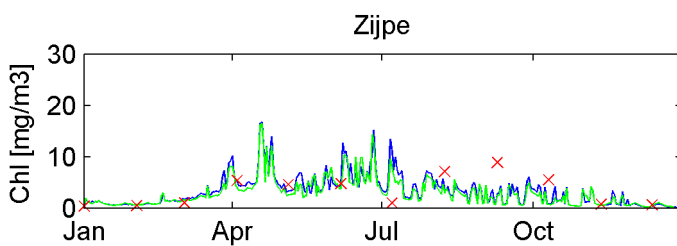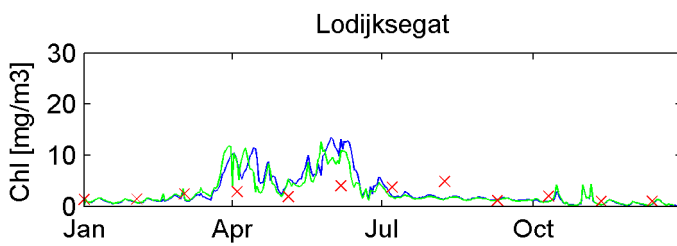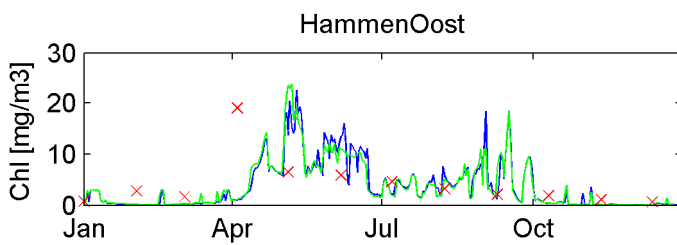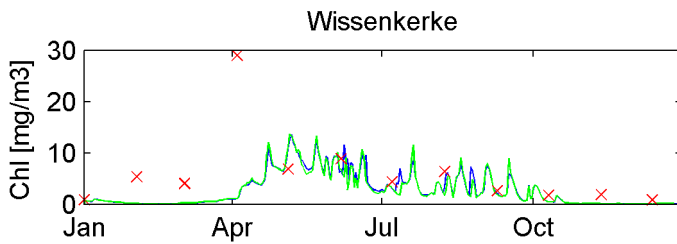
A.11 **Nitrate: simulated NO3 concentrations of the 3D GEM base case (blue curves) and for the 3D GEM case study on erosion (green curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.**
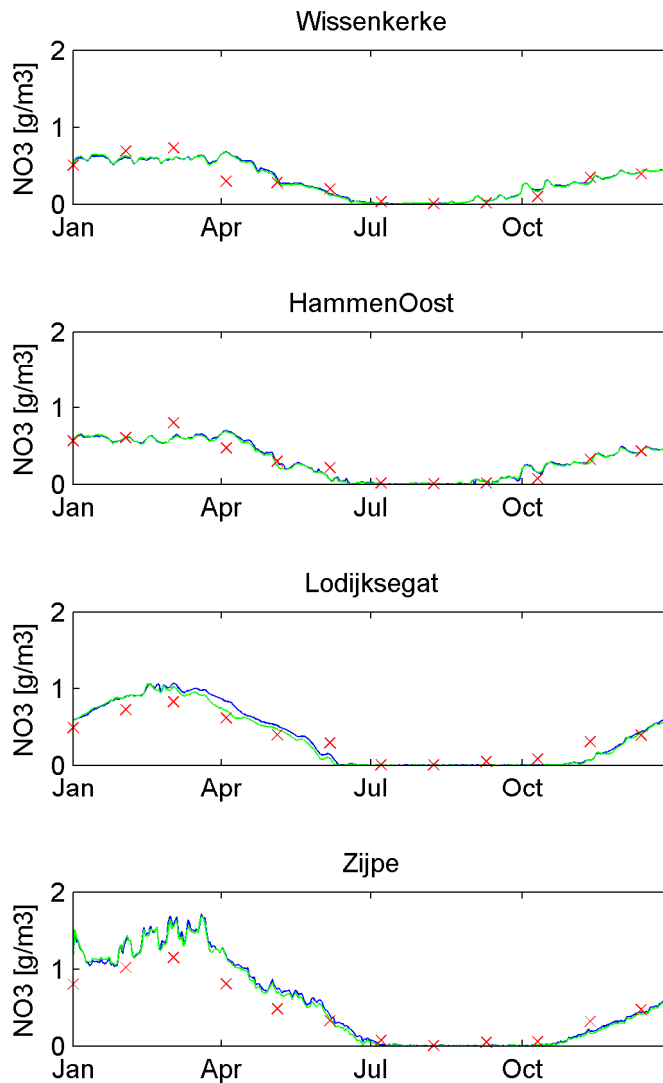
A.12   Phosphate: simulated PO4 concentrations of the 3D GEM base case (blue curves) and for the 3D GEM case study on erosion (green curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.
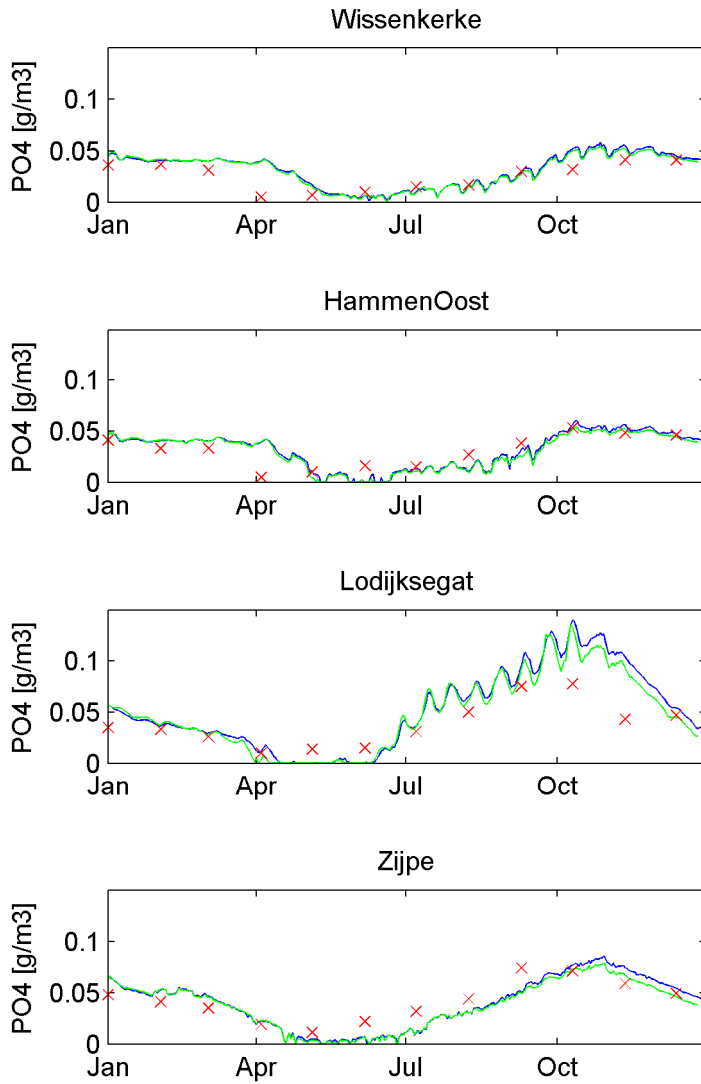
A.13 **Silicate: simulated Si concentrations of the 3D GEM base case (blue curves) and for the 3D GEM case study on erosion (green curves) at four locations in the Oosterschelde. Red crosses are measured concentrations.**
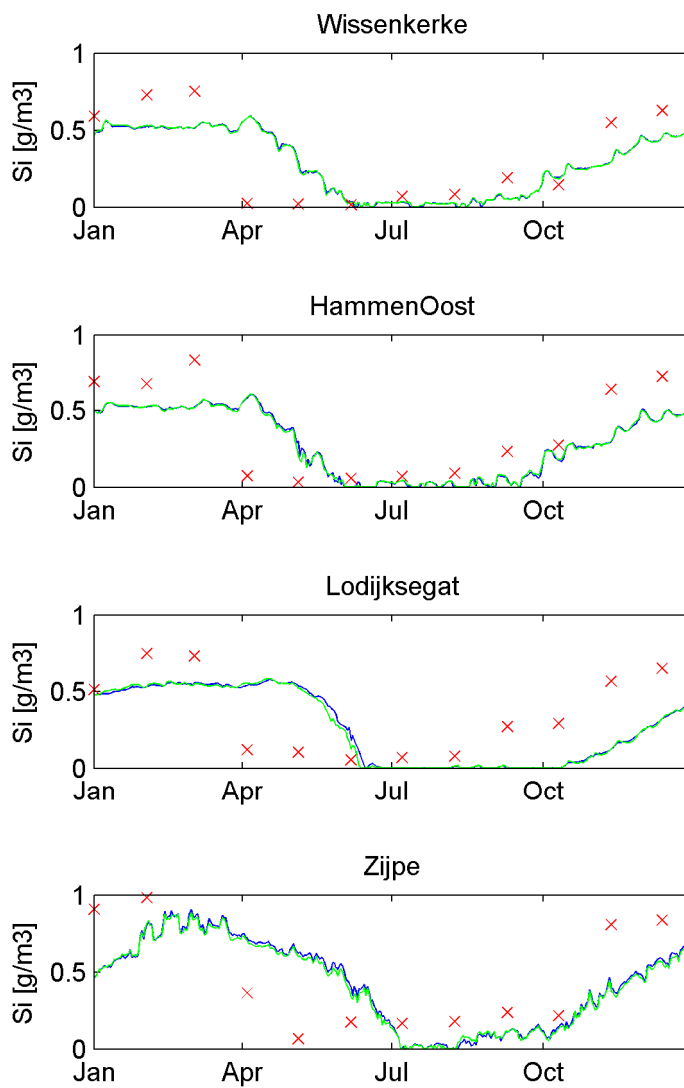
B Model code

B.1 DEB population growth for isomorphs

```
    SUBROUTINE DEBISO  (PMSA , FL , IPOINT , INCREM , NOSEG , NOFLUX ,
   +           IEXPNT, IKNMRK, IPODIM, NOQ1, NOQ2, NOQ3    )
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS: 'DEBISO' :: DEBISO
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 k1         R*4 1 I   x rate at 20degC                   (-)
```

```
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 +14 ! number of input output variables in PMSA array
      INTEGER :: IP(220)           ! 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,k1,
     +        Length , Lm , Vp, shape , Em_L3 , Eg_L3 , Pm_L3 ,
     +        JXm_L2 , PAm_L2 , kappa , Xk , Xk_B, XK_S ,
     +        conv_J_gC_Alg , 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, Length_act

     REAL ::    Nin, Nuit, Pin, Puit, Siin, Siuit, Nbal, Pbal, Sibal,
      +        UptakeC, NuptakeC,PuptakeC,LimUptake,Biomass,BioAFDW,
      +        BioWW,BioWW_m2,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))
    k1        =          PMSA( IP(25))
    GSI_upper =          PMSA( IP(26))
    rSpawn    =          PMSA( IP(27))
    MinSPTemp =          PMSA( IP(28))
    Xk        =          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_gC_Alg=       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
```

```fortran
      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_act = (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.)))
c        kT= exp(Ta/(20.+273.)- Ta/(Temp +273.))
c    +     /(1.+ exp(Tal/(Temp+273.)-Tal/Tl)
c    +        + exp(Tah/Th-Tah/(Temp+273.)))
         kT=   exp(Ta/(20.+273.)- Ta/(Temp +273.))*
     +    (1.+ exp(Tal/(20.+273.)-Tal/Tl)+exp(Tah/Th-Tah/(20.+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)

      Xk_S = Xk
      Xk_B = Xk

   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_Alg                    ! Cuptake in (gC m-2 d-1)
        NuptakeC  = Nuptake * conv_J_gC_Alg / TN    ! Nuptake in carbon equivalents (gC m-2 d-1)
        PuptakeC  = Puptake * conv_J_gC_Alg / 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_Alg          ! 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_Alg        !(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.*Onethird) + 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

c     GSI = R / (V * conv_J_cm3 );
          GSI = GSI_upper+1
     kappa_R = 0.80
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_Alg *Dens /DEPTH
     FL (18 + IFLUX  ) = (1.- GEM)*dUpt(1)* conv_J_gC_Alg *Dens /DEPTH
     FL (19 + IFLUX  ) =          dUpt(2)* conv_J_gC_Alg *Dens /DEPTH
     FL (20 + IFLUX  ) = (0.+ GEM)*dUpt(1)* NCFOOD(1) *conv_J_gC_Alg*Dens/DEPTH
     FL (21 + IFLUX  ) = (1.- GEM)*dUpt(1)* NCFOOD(1) *conv_J_gC_Alg*Dens/DEPTH
     FL (22 + IFLUX  ) =          dUpt(2)* NCFOOD(2) *conv_J_gC_Alg*Dens/DEPTH
     FL (23 + IFLUX  ) = (0.+ GEM)*dUpt(1)* PCFOOD(1) *conv_J_gC_Alg*Dens/DEPTH
     FL (24 + IFLUX  ) = (1.- GEM)*dUpt(1)* PCFOOD(1) *conv_J_gC_Alg*Dens/DEPTH
     FL (25 + IFLUX  ) =          dUpt(2)* PCFOOD(2) *conv_J_gC_Alg*Dens/DEPTH
     FL (26 + IFLUX  ) = (0.+ GEM)*dUpt(1)* SiCFOOD(1)*conv_J_gC_Alg*Dens/DEPTH
     FL (27 + IFLUX  ) = (1.- GEM)*dUpt(1)* SiCFOOD(1)*conv_J_gC_Alg*Dens/DEPTH
     FL (28 + IFLUX  ) =          dUpt(2)* SiCFOOD(2)*conv_J_gC_Alg*Dens/DEPTH

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

c Check on budgets: Nbal, Pbal and Sibal should be zero.
     Nin  = NUptake * conv_J_gC_Alg * Dens/DEPTH
     Nuit = FL(2+IFlux)+ FL(5+IFLUX)+FL(8+IFLUX)
     Pin  = PUptake * conv_J_gC_Alg * Dens /DEPTH
     Puit = FL(3+IFlux)+ FL(6+IFLUX)+FL(9+IFLUX)
```

```
  Siin  = SiUptake * conv_J_gC_Alg/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
  Harvest=fMor/conv_gWW_gC*(V_m2*conv_cm3_gC+(E_m2+R_m2)*conv_J_gC)*Area        !(gWW d-1)
       Spawn = Spawning * (conv_J_gC/ conv_gWW_gC) * Dens * Area / Delt
  Biomass= (V_m2 * conv_cm3_gC + (E_m2+R_m2) * conv_J_gC) * Area
  BioAFDW= Biomass / conv_gAFDW_gC
  BioWW=  Biomass / conv_gWW_gC
  BioWW_m2=  Biomass / conv_gWW_gC / Area
  GrossGr= Pa * (conv_J_gC) * Dens
       SemiNetGr= ((Growth * conv_cm3_gC)+ (Pr * conv_J_gC)) * Dens


   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)) = BioWW_m2
   PMSA(IP(218)) = Length_act
   PMSA(IP(219)) = GrossGr
   PMSA(IP(220)) = SemiNetGr


  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    )
!DEC$ ATTRIBUTES DLLEXPORT, ALIAS: 'DEBV1' :: DEBV1
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 k1        R*4 1 I   x rate at 20degC                                           (-)
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        R*4 1 I   x halfrate const food uptake Sup fdr   (gC/m3)
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 +13 ! number of input output variables in PMSA array
     INTEGER :: IP(218)         ! 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 , Xk_B, XK_S ,
    +        conv_J_gC_Alg, 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, k1

     REAL ::   Nin, Nuit, Pin, Puit, Siin, Siuit, Nbal, Pbal, Sibal,
```

```
      +         UptakeC, NuptakeC,PuptakeC,LimUptake,Biomass,BioAFDW,
      +         BioWW,BioWW_m2,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))
    TI        =          PMSA( IP(23))
    k1        =          PMSA( IP(24))
    GSI_upper =          PMSA( IP(25))
    rSpawn    =          PMSA( IP(26))
    MinSPTemp =          PMSA( IP(27))
    Xk        =          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_J_gC_Alg=       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.)))
c        kT= exp(Ta/(20.+273.)- Ta/(Temp +273.))
c    +    /(1.+ exp(Tal/(Temp+273.)-Tal/Tl)
c    +      + exp(Tah/Th-Tah/(Temp+273.)))
        kT=  exp(Ta/(20.+273.)- Ta/(Temp +273.))*
   +    (1.+ exp(Tal/(20.+273.)-Tal/Tl)+exp(Tah/Th-Tah/(20.+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)

        Xk_S = Xk
        Xk_B = Xk

        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_Alg                 ! Cuptake in (gC m-2 d-1)
      NuptakeC = Nuptake * conv_J_gC_Alg / TN    ! Nuptake in carbon equivalents (gC m-2 d-1)
      PuptakeC = Puptake * conv_J_gC_Alg / 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_Alg          ! 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_Alg        !(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
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          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

c     GSI = (R_m2) / ((1-(Vp/(Vp+Vd)))* V_m2 *conv_cm3_J )
      GSI = GSI_upper+1.

      kappa_R = 0.80 ! 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/(Vp+Vd)))* V_m2 * kT     !maturity maintenance (J m-2 d-1)

          Pr = max((((1.-kappa) * Pc -Pj -Pd)*(1-Vp/(Vp+Vd))),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
                       if (R_m2.GT.0) then
                               DoSpawn  = 1.
                       endif
        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_Alg /DEPTH
             FL (19 + IFLUX  ) = (1.- GEM)*dUpt(1) * conv_J_gC_Alg /DEPTH
             FL (20 + IFLUX  ) =          dUpt(2) * conv_J_gC_Alg /DEPTH
             FL (21 + IFLUX  ) = (0.+ GEM)*dUpt(1) * NCFOOD(1)  * conv_J_gC_Alg /DEPTH
             FL (22 + IFLUX  ) = (1.- GEM)*dUpt(1) * NCFOOD(1)  * conv_J_gC_Alg /DEPTH
             FL (23 + IFLUX  ) =          dUpt(2) * NCFOOD(2)  * conv_J_gC_Alg /DEPTH
             FL (24 + IFLUX  ) = (0.+ GEM)*dUpt(1) * PCFOOD(1)  * conv_J_gC_Alg /DEPTH
             FL (25 + IFLUX  ) = (1.- GEM)*dUpt(1) * PCFOOD(1)  * conv_J_gC_Alg /DEPTH
             FL (26 + IFLUX  ) =          dUpt(2) * PCFOOD(2)  * conv_J_gC_Alg /DEPTH
             FL (27 + IFLUX  ) = (0.+ GEM)*dUpt(1) * SiCFOOD(1) * conv_J_gC_Alg /DEPTH
             FL (28 + IFLUX  ) = (1.- GEM)*dUpt(1) * SiCFOOD(1) * conv_J_gC_Alg /DEPTH
             FL (29 + IFLUX  ) =          dUpt(2) * SiCFOOD(2) * conv_J_gC_Alg /DEPTH

             do IFOOD=3,NFOOD
                 FL(27 + IFOOD + IFLUX) = dUpt(IFOOD)*conv_J_gC_Alg /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_Alg/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_Alg/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_Alg/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*(V_m2*conv_cm3_gC+(E_m2+R_m2)*conv_J_gC)/conv_gWW_gC*Area          !(gWW  d-
1)
         Spawn = Spawning * (conv_J_gC/ conv_gWW_gC) * Area
    Biomass= (V_m2 * conv_cm3_gC + (E_m2+R_m2) * conv_J_gC) * Area
    BioAFDW= Biomass / conv_gAFDW_gC
    BioWW=  Biomass / conv_gWW_gC
    BioWW_m2=  Biomass / conv_gWW_gC / Area
         GrossGr= Pa * (conv_J_gC)
         SemiNetGr= (Growth * conv_cm3_gC) + (Pr * conv_J_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
     PMSA(IP(216)) = BioWW_m2
     PMSA(IP(217)) = GrossGr
     PMSA(IP(218)) = SemiNetGr



    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
```

C Initial settings and conversions

Parameter settings and initial values for grazers in the integrated ecosystem model for the Oosterschelde. For underlying formulas, see basisdata_v3.xls

C.1 Initial settings and conversions for the three species

Algemeen		waarde	eenheid	dimensie	ref
Carbon/Chl	carbon:chl ratio	0.078	mgC/ugChl		afgeleid uit modelresultaten
cm3/gAFDW		8.3			

Grazers opgedeeld per soort		waarde	eenheid	dimensie	ref
afdw/ww	mossel	0.056	-	g/g	afgeleid uit EVAII rapport
	oester	0.00825	-	g/g	email Jeroen
	kokkel	0.043	-	g/g	afgeleid uit EVAII rapport
	MZI-mossel	0.039	-	g/g	schatting
Standing stock	mosselen	3.00E+07	kg	WW (nettogewicht)	email Jeroen (2005-2009)
	oesters	3.55E+08	kg	WW (nettogewicht)	email Jeroen (2005-2009)
	kokkels	4.00E+07	kg	WW (nettogewicht)	email Jeroen (2005-2009)
Standing stock	mosselen	1.68E+06	kg	AFDW	calc obv afdw/ww ratio bij mosselen (EVAII rapport)
	oesters	2.93E+06	kg	AFDW	calc obv afdw/ww ratio bij oesters (EVAII rapport)
	kokkels	1.72E+06	kg	AFDW	calc obv afd/ww ratio bij kokkels (EVAII rapport)
Initial structural volume (population)	mosselen	6.69E+09	cm3	structural volume	calc obv 60% structural volume (tov tot massa incl energy reserves) en 80% van zomergewicht (afgeleid uit mosseldata in keyzones)
	oesters	1.17E+10	cm3	structural volume	calc obv 60% structural volume (tov tot massa incl energy reserves) en 80% van zomergewicht (afgeleid uit mosseldata in keyzones)
	kokkels	6.85E+09	cm3	structural volume	calc obv 60% structural volume (tov tot massa incl energy reserves) en 80% van zomergewicht (afgeleid uit mosseldata in keyzones)
Initial energy reserves (population)	mosselen	8.E+12	J	Energy reserve	Gekalibreerd zodat deze variabele over het jaar genomen stabiel gedrag vertoond
	oesters	6.E+12	J	Energy reserve	Gekalibreerd zodat deze variabele over het jaar genomen stabiel gedrag vertoond
	kokkels	3.E+12	J	Energy reserve	Gekalibreerd zodat deze variabele over het jaar genomen stabiel gedrag vertoond
Initial gonadal buffer (population)	mosselen	7.E+12	J	Gonads	Gekalibreerd zodat deze variabele over het jaar genomen stabiel gedrag vertoond
	oesters	2.E+12	J	Gonads	Gekalibreerd zodat deze variabele over het jaar genomen stabiel gedrag vertoond
	kokkels	2.E+12	J	Gonads	Gekalibreerd zodat deze variabele over het jaar genomen stabiel gedrag vertoond
Harvest	mosselen	2.25E+07	kg/yr	WW	WL rapport keyzones
	oesters	1.00E+06	kg/yr	WW	WL rapport keyzones
	kokkels	0.00E+00	kg/yr	WW	WL rapport keyzones
Harvest	mosselen	1.26E+06	kg/yr	AFDW	calc obv afdw/ww ratio bij mosselen (EVAII rapport)
	oesters	8.25E+03	kg/yr	AFDW	calc obv afdw/ww ratio bij oesters (EVAII rapport)
	kokkels	0.00E+00	kg/yr	AFDW	calc obv afdw/ww ratio bij kokkels (EVAII rapport)
Harvesting rate	mosselen	3.80E-03	1/d		Calc
	oesters	7.73E-06	1/d		Calc
	kokkels	0.00E+00	1/d		Calc
Background mortality rate	mosselen	6.11E-04	1/d		Calc obv 20% mortality per year
	oesters	2.89E-04	1/d		calc obv 10% mortality per year
	kokkels	2.67E-03	1/d		calc obv 28% mortality in 123 days (Kamermans 2003)
Reference length	mosselen	5.30	cm		medium size mosselen (WL rapport keyzones)

	oesters	10.00	cm		deze waarde is zo gekozen dat er een stabiele populatie ontstaat
	kokkels	2.30	cm		deze waarde is zo gekozen dat er een stabiele populatie ontstaat
saturatieconstante Xk	mosselen	0.17394	mgC/l	mgFood/l	=2.23 ugChl/l * C/Chl ratio (PTRS Troost et al 2010) Deze waarde correspondeert met alleen plankton, geen detc, en dus moet prefdetc=0
	oesters	0.5	mgC/l	mgFood/l	deze waarde is zo gekozen dat er een stabiele populatie ontstaat
	kokkels	0.21372	mgC/l	mgFood/l	=2.74 ugChl/l * C/Chl ratio (PTRS, Troost et al 2010) Deze waarde correspondeert met alleen plankton, geen detc, en dus moet prefdetc=0
Fysiologische parameters	mosselen, oesters, kokkels				Van der Veer et al 2006, zie tabel 2.2

C.2 Initial settings and conversions for generalized grazer

Grazers ongestructureerd		waarde	eenheid	Dimensie	ref
Totale schelpdierstock	sum mosselen, oesters, kokkels	4.25E+08	kg	WW	sum
Standing stock	sum mosselen, oesters, kokkels	6.33E+06	kg	AFDW	sum
Initial structural volume	sum mosselen, oesters, kokkels	2.52E+10	cm3	structural volume	sum
Initial energy reserves (population)	sum mosselen, oesters, kokkels	25.2E+12	J	Energy reserve	sum
Initial gonadal reserves (population)	sum mosselen, oesters, kokkels	11.E+12	J	Gonads	sum
Totale oogst schelpdieren	sum mosselen, oesters, kokkels	2.35E+07	kg/yr	WW	sum
Totale oogst schelpdieren	sum mosselen, oesters, kokkels	1.27E+06	kg/yr	AFDW	sum
Harvesting rate	avg mosselen, oesters, kokkels	6.13E-04	1/d		calc
Background mortality rate	avg mosselen, oesters, kokkels	5.29E-04	1/d		calc average obv stocks en survival rates per soort
Reference length	mosselen	5.30E+00	cm		medium size (WL rapport keyzones)
saturatieconstante Xk	mosselen	0.17394	mgC/l	mgFood/l	=2.23 ugChl/l * C/Chl ratio Deze waarde correspondeert met alleen plankton, geen detc, en dus moet prefdetc=0
Fysiologische parameters	mosselen				Van der Veer et al 2006, zie tabel 2.2

C.3 Initial settings and conversions for the three size-classes of grazers

Grazers opgedeeld in grootteklasses (klein, medium, en groot)		waarde	eenheid	dimensie	ref
Standing stock	small	8.50E+07	kg	WW	calc obv 20% totale stock (mosselen en kokkel AFDW/2 tov totale stock)
	medium	8.50E+07	kg	WW	calc obv 20% totale stock (mosselen en kokkel AFDW/2 tov totale stock)
	large	2.55E+08	kg	WW	calc obv 60% totale stock (oester AFDW tov totale stock)
Standing stock	small	4.76E+06	kg	AFDW	calc obv vleesgehalte mossel
	medium	7.01E+05	kg	AFDW	calc obv vleesgehalte mossel
	large	1.10E+07	kg	AFDW	calc obv vleesgehalte oester
Initial structural volume (population)	small	1.90E+10	cm3	structural volume	calc obv 60% structural volume (tov tot massa incl energy reserves) en 80% van zomergewicht
	medium	2.79E+09	cm3	structural volume	calc obv 60% structural volume (tov tot massa incl energy reserves) en 80% van zomergewicht
	large	4.37E+10	cm3	structural volume	calc obv 60% structural volume (tov tot massa incl energy reserves) en 80% van zomergewicht
Harvest	small	0.00E+00	kg/yr	WW	geen oogst van kleine ind
	medium	2.25E+07	kg/yr	WW	mosseloogst valt grotendeels in deze grootteklasse
	large	1.00E+06	kg/yr	WW	oesteroogst valt in deze grootteklasse
Harvesting rate	small	0.00E+00	1/d		calc
	medium	8.42E-04	1/d		calc
	large	1.08E-05	1/d		calc
Background mortality rate	small	1.90E-03	1/d		calc obv 50% mortality per year
	medium	6.11E-04	1/d		calc obv 20% mortality per year
	large	2.89E-04	1/d		calc obv 10% mortality per year
Initial length	small	1.00	cm		~0.75* reflength kokkels, , afgerond naar beneden ivm snelle groei kleine individuen
	medium	3.00	cm		~0.75* reflength mosselen
	large	7.00	cm		~0.75* reflength oesters, afgerond naar boven ivm langzame groei grote individuen
Initial structural volume (individual)	small	2.36E-02	cm3		calc obv shapefactor=0.287
	medium	6.38E-01	cm3		calc obv shapefactor=0.287
	large	1.84E+00	cm3		calc obv shapefactor=0.175
Initial density	small	8.02E+11	#	number of individuals	calc
	medium	4.38E+09	#	number of individuals	calc
	large	2.38E+10	#	number of individuals	calc
saturatieconstante Xk	small	0.17394	mgC/l	mgFood/l	=2.23 ugChl/l * C/Chl ratio Deze waarde correspondeert met alleen plankton, geen detc, en dus moet prefdetc=0
	medium	0.17394	mgC/l	mgFood/l	=2.23 ugChl/l *C/Chl ratio Deze waarde correspondeert met alleen plankton, geen detc, en dus moet prefdetc=0
	large	0.234	mgC/l	mgFood/l	=3 ugChl/l * C/Chl ratio Deze waarde correspondeert met alleen plankton, geen detc, en dus moet prefdetc=0
Fysiologische parameters	small				parameterwaardes mosselen (Van der Veer et al 2006), zie tabel 2.2
	medium				parameterwaardes mosselen (Van der Veer et al 2006), zie tabel 2.2
	large				parameterwaardes oesters (Van der Veer et al 2006), zie tabel 2.2

D Validation of system specific values

Data from the flume-measurements (van Duren & Troost 2010) are used to derive the saturation-constant for oysters, and the energy content of algae. Results are compared to the data estimated and/or calibrated with the GEM. For underlying formulas, see AfleidingXkOestersUitFlumeExperiment.xls

D.1 Input data

1 parameterwaardes (van der Veer et al, 2006) see Table 2.2

	shape-coeff	{J_im}
mossel	0.287	147.6
kokkel	0.381	68.6
oester	0.175	420
ensis	?	?

2 Data van flume experiments (van Duren & Troost 2010)

	biomass	#	density	length	clearance rate (CR-)	clearance rate (CR+)	X+	X-
	gDW	-	#/m2	mm	l/hr	l/hr	ug/l	ug/l
mossel	330.7	1246	1038.3	42.7	2900	4000	0.7	0.9
kokkel	375.8	543	452.5	33.6	700	900	1.1	1.6
oester	341.7	376	313.3	102.2	3100	3200	0.7	0.9
ensis	735.6	402	335.0	123.3	1800	1900	1	1.5

(+ is de gemiddelde waarde gemeten bij 0.1 m/s, - is die bij 0.3 m/s)

3 Data van calibration study (Troost et al, 2010).

	Xk	J/ugchl
mossel	2.23	3.49
kokkel	2.74	2.30
oester	?	2.38
ensis		

4 DEB-vergelijkingen en parameters

$p\_A = \{P\_Am\} * f * V^{(2/3)}$

$f = X/(Xk + X)$

p_A	J/d/m2	assimilation rate
{P_Am}	J/d/cm2	surface-area specific assimilation rate
f	-	scaled functional response
V_ind	cm3	individual structural volume
V	cm3/m2	total structural volume
V^(2/3)	cm2/m2	total surface area
X	ug/l	chl-conc
Xk	ug/l	saturation constant
a	J/ugCHL	energy-content algae

D.2 Deriving the half-satuarion constant for oysters

1 Berekening DEB parameters op basis van invoergegevens

f   =X/(X+Xk)

V_ind   =(length*shape)^3

V^(2/3)   =density*(length*shape)^(2/3)

p_A1   =clearance rate * hr/d * X * a

p_A2   ={P_Am}*f*V^(2/3)

2 Afleiding energycontent algen (parameter a)

voor mosselen en kokkels zijn de saturatie-constantes en andere benodigde parameters bekend

voor deze soorten kan parameter a worden afgeleid dmv gelijkstellen p_A1(a)=p_A2

vervolgens kan met goal-seek de waarde van a worden gefit (goal: cel X54 =0 door aanpassen cel H54)

	current-velocity	CR		X	a	p_A1
	m/s	l/hr	hr/dag	ugchl/l	J/ugchl	J/d/m2
mossel	0.1	2900	24	0.7	1.16	56303.54
	0.1	4000	24	0.9	0.78	67536.76
kokkel	0.1	700	24	1.1	0.80	14870.28
	0.1	900	24	1.6	0.55	19170.13

{P_Am}	density	length	shape	Vol_ind	Opp_ind	V2/3
J/d/cm2	#	cm	-	cm3/ind	cm2/ind	cm/m2
147.6	1038.3	4.27	0.287	1.84	1.50	1559.35
147.6	1038.3	4.27	0.287	1.84	1.50	1559.35
68.6	452.5	3.36	0.381	2.10	1.64	741.56
68.6	452.5	3.36	0.381	2.10	1.64	741.56

X	Xk	f	p_A2		diff (p_A1-p_A2)^2
ug/l	ug/l	-	J/d/m2		
0.7	2.23	0.24	54987		0.00057335
0.9	2.23	0.29	66180		0.00042027
1.1	2.74	0.29	14572		0.00041769
1.6	2.74	0.37	18754		0.00049155

3 Afleiding saturatie constante Xk voor oesters

neem aan a=gemiddelde van optimale waardes (a is in principe een soorts-onafhankelijke parameter)
vervolgens kan met goal-seek de optimale waarde voor Xk van oesters worden bepaald
(goal: cel X66 = 0 door
aanpassen cel T66)

	current-velocity	CR		X	a	p_A1	
	m/s	l/hr	hr/dag	ugchl/l	J/ugchl	**J/d/m2**	
oester	0.1	3100	24	0.7	0.82	**42922.91**	
	0.3	3200	24	0.9	0.82	**56966.82**	

	{P_Am}	density	length	shape	Vol_ind	Opp_ind	V2/3
	J/d/cm2	#	cm	-	cm3/ind	cm2/ind	cm/m2
	420	313.3	10.22	0.175	5.72	3.20	1002.16
	420	313.3	10.22	0.175	5.72	3.20	1071.58

	X	Xk	f	p_A2		diff (p_A1-p_A2)^2
	ug/l	ug/l	-	**J/d/m2**		
	0.7	6.01	0.10	**43920**		0.00051543
	0.9	6.03	0.13	**58421**		0.00061969

	Xk	
	mgC/l	Overeenkomst met de waarde geschat/gecalibreerd in de GEM (0.5 mgC/l)
	0.47	Goed!
	0.47	Goed!

D.3 Deriving the energy content of algae

1 validatie energy content algen

conversie naar de energy-content parameter gebruikt in het Deltakennis model (structural biomass grazers: 1.74E-05 gC/J, algae: 10E-05gC/J)

J/ugchl	gC/J	Overeenkomst met de waarde geschat/gecalibreerd in de GEM
1.155656	6.75E-05	Factor 5 lager
0.781675	9.98E-05	Goede overeenkomst!
0.80	9.69E-05	Goede overeenkomst!
0.554691	1.41E-04	Factor 15 hoger