CHARISMA 2.0

Model description

Egbert van Nes

January 2000

Contents

1. Introduction

1.1 What is Charisma?

Charisma is an individual-based simulation model for submerged macrophytes. That means that all plants are modeled individually. Charisma is mainly a combination of two previous models: ArtiVeg (Van Nes & Scheffer, 1996) and MEGAPLANT (Scheffer et al., 1993). The rules of the model are based upon the approach of MEGAPLANT, but the model is reimplemented in the framework of ArtiVeg.

The model is spatially explicit. The location of each plant is defined by two coordinates. Environmental conditions are stored in a two-dimensional grid. Although the model is spatially explicit, it is possible to run it as a more traditional age structured population model. This is achieved by selecting only one grid cell.

Charisma is a multi-species model. In the default mode, it has two species, *Potamogeton* and *Chara*, but the user can add many new species (limited by computer memory only).

1.2 Why Charisma?

The model is a tool to study macrophytes. Macrophytes play an important role in aquatic ecosystems, since they can stabilize clear water. Physiological data of macrophytes species and environmental variables are used as input of the model. The model produces spatial and temporal patterns of macrophytes which should be compared with field data.

The model can be used to study the following kind of questions:

- Could the results of laboratory experiments such as photosynthesis measurements explain what we see in the field?
- Which physiological features make *Potamogeton pectinatus* a better competitor than *Chara* in turbid water?
- Which processes and parameters are most important to explain the growth of macrophytes? What important information is currently missing?
- What would we suspect if we compare competition between species in different climates?
- The model can generate hypotheses about the mechanisms of alternative stable states in lakes.

A weak point of a deterministic model like Charisma is that it is hard to include all relevant information and processes. You never will be sure that the core of the physiological data is correct. Therefore, the results of the model should be used with care.

A strong point of an individual based model as Charisma is that there are many ways to check the results, because the model generates detailed information. Moreover, Charisma has a strong graphical user interface which makes it easy to explore the model, showing all graphical results instantaneous. During the run, the user has full access to all relevant parameters in the model.

1.3 The computer program

The computer program is implemented in Borland Delphi 4.0, which uses Object Pascal, an object-oriented computer language. It is a language that generates code that is nearly as fast as C++, but with a strong type checking, avoiding bugs. Using objects helps to structure the design of the application and to reuse modules in other applications. About 80% of the program is shared with the models ArtiVeg and Piscator. The program uses Microsoft Windows 95 for the extended user interface.
The previous version (1.0) of the program was implemented in Windows 3.1. In future, model changes will only be supported in the Windows 95 version.

The user-friendly interface has the following features:

- The parameters of the model can be inspected and changed in a dialog box.
- The model can be interrupted any time.
- During the run, the model results are shown in graph windows simultaneously. The graphs include various time plots and three-dimensional representations of the grid. The graphs can be printed or exported to several Windows applications without loss of quality.
- Model results and parameter values can be saved to files.
- A help system provides the user with on-line information about the application, including an extended model description. It is a standard Windows help system, which includes hyper text (cross-references to related topics) and keyword search. To get context sensitive help, the user may press the *F1*-key in all situations. In many dialog boxes, a Help button can be used alternatively.

1.4 Analysers

Several flexible types of special analyses are available for more elaborate studies of model behaviour:

Parameter analysis - change parameters in a structured way
Monte-Carlo sensitivity analyser - sensitivity of the model to parameter change
Bayesian uncertainty analyser - uncertainty of the model outcome
Parameter calibration CRS - automatic calibration
Equilibrium analyser - study the model in equilibrium
Stochasticy analyser - rerun the model to detect stochastic mechanisms

1.4.1 Parameter analyser

Change one or few parameters gradually and plot the effects on the results. A special version of this analysis can be used to detect alternative stable states. A conditioning parameter that causes the shift to an alternative stable state is first gradually increased and after the shift decreased again.

The parameter *NStepsAnal* defines the number of steps to be taken after an initial stabilising period of *NInitStabilise* years.
There can be 3 different stages in each step:

1. Stabilise a period *NStabilise* years
2. Disturb the equilibrium (optional for studying resilience) *Ndisturbance* years
3. Write the results *NYears* years

The parameter *ParamArray* holds all information that is needed for each parameter to be changed (among others: min, max, nsteps, see: Parameter for analysis dialog) The parameter *ParamCount* determines the number of parameters to be changed. The parameter *ResetEachStep* determines the whether the model should be reset after each step.

RunType can be used to make the analyser inactive.

The parameter *Dummy* has no function in the model. It can be used as a dummy variable to be grouped with a parameter group to produce graphs with another scale than the parameters that were varied.

1.4.2 Monte-Carlo sensitivity analyser

This analyser does a sensitivity analysis like Klepper (1989). Change several parameters at random and independently. By analysing the impact of the changes on the results, the most critical parameters can be detected. Moreover the parameters can be clustered into groups that have the same or opposite effects on the model results.

The parameter *NStepsAnal* defines the number of steps to be taken after an initial stabilising period of *NInitStabilise* years.
There can be two different stages in each step:

1. Stabilise a period *NStabilise* years
2. Write the results *NYears* years

The parameter *ParamArray* holds all information that is needed for each parameter to be changed (among others: mean, standard deviation and distribution, see: Parameter of Monte-Carlo analysis dialog) The parameter *ParamCount* determines the number of parameters to be changed.

The parameters *ClusMethod* and *DistMeasure* determine how the cluster analysis is performed. These parameters can also be changed while creating a

1.4.3 Bayesian uncertainty analysis

Bayesian uncertainty analysis. The same as a Monte-Carlo sensitivity analysis, but with a different target. Use this analysis to estimate the uncertainty in the model output, by drawing all parameters from statistical distributions. The model results are presented as percentiles (5%, 25%, 50% 75% and 95%).

The parameter *NStepsAnal* defines the number of steps to be taken after an initial stabilising period of *NInitStabilise* years.
There can be 2 different stages in each step:

1. Stabilise a period *NStabilise* years
2. Write the results *NYears* years

The parameter *ParamArray* holds all information that is needed for each parameter to be changed (among others: mean, standard deviation and distribution, see: Parameter of Monte-Carlo analysis dialog) The parameter *ParamCount* determines the number of parameters to be changed.

1.4.4 Controlled Random Search (CRS) analyser

Controlled random search (Price, 1979; Klepper & Hendrix, 1994a; Klepper & Hendrix, 1994b). Calibration of one or more parameters. The user must provide one or more data files with data that are compatible with data generated by the time graphs. The parameters are varied within preset fixed boundaries. The average or minimal adjusted R2 of each data set is used as goodness of fit criterion. During the process the boundaries are narrowed to increase the efficiency. Note that the amount of computing time increases at least quadratic with the number of parameters considered (Klepper & Hendrix, 1994b). Therefore, the number of parameters that are calibrated simultaneously, should be kept low. De Hoop et al. (1992) describe the technical details of implementing the method in a clear way.

This analyser may use a number of initial stabilising years (usually not needed in the analysis) *NInitStabilise* and a number of stabilising years (*NStabilise*, not needed either).

NYears defines the number of years to be run (should correspond with the years with data). The parameters *ParamCount* and *ParamArray* define the parameters and their ranges in which they are changed. The parameter *NVase* defines the number of elements in the "vase". The higher this number the slower convergence is achieved, but the smaller the risk of sticking in a local optimum. The parameter *GofCrit* defines the convergence criterion. The model results are compared with data from files, see *SamplerFiles*. The parameter *ConvCrit* sets the convergence criterion ((best parameter setting - worst setting in vase)/best parameter setting), but this parameter can be overruled by the maximum number of steps *NSteps*.

1.4.5 Equilibrium analyser

Equilibrium analyser This simple analyser lets the model first stabilise for some years. Thereafter, the equilibrium state of the model is analysed.
After the initial stabilising period of *NInitStabilise* years, the program writes the results of *NYears* years

1.4.6 Stochasticy analyser

Stochasticy analyser - This simple analyser repeats a certain run many times to analyse the stochasticy of the model.
After an optional initial stabilising period of *NInitStabilise* years, a simulation run of *NYears* is repeated *NStepsAnal* times.

1.5 Simulation approach

Photosynthesis follows the daily cycle of light intensity and varies with depth. We integrate over both time and depth to obtain the total daily production. We apply Gaussian integration (Goudriaan, 1986), a very efficient technique to approximate both integrals.

Simple simulations starting from a fixed biomass can be used to produce a seasonal growth curve. Several flexible types of special analyses are available for more elaborate studies of model behavior:

1. *Monte-Carlo sensitivity analysis* (Klepper, 1989). Change several parameters at random and independently. By analyzing the impact of the changes on the results, the most critical parameters can be detected. Moreover the parameters can be clustered into groups that have the same or opposite effects on the model results.
2. *Bayesian uncertainty analysis* The same as a Monte-Carlo sensitivity analysis, but with a different target. Use this analysis to estimate the uncertainty in the model output, by drawing all parameters from statistical distributions. The model results are presented as percentiles (5%, 25%, 50% 75% and 95%).
3. *Parameter analysis*. Change one or few parameters gradually and plot the effects on the results. A special version of this analysis can be used to detect alternative stable states. A conditioning parameter that causes the shift to an alternative stable state is first gradually increased and after the shift decreased again.
4. *Controlled random search* (Price, 1979; Klepper & Hendrix, 1994). Calibration of one or more parameters. The user must provide one or more data files with data that are compatible with data generated by the time graphs. The parameters are varied within preset fixed boundaries. The average or minimal adjusted R2 of each data set is used as goodness of fit criterion. During the process the boundaries are narrowed to increase the efficiency. Note that the amount of computing time increases at least quadratic with the number of parameters considered (Klepper & Hendrix, 1994). Therefore, the

number of parameters that are calibrated simultaneously, should be kept low. De Hoop (1992) describes the technical details of implementing the method in a clear way.

5. *Equilibrium analyser* This simple analyser lets the model first stabilize for some years. Thereafter, the equilibrium state of the model is analysed.
6. *Stochasticy analyser* - This simple analyser repeats a certain run many times to analyse the stochasticy of the model.

1.5.1 Super-individuals

A general problem with individual based models is the large number of individuals that is required. This is necessary to avoid loss of variation, irregular dynamics, and large sensitivity to the value of random generator seeds. These large numbers of individuals result in impractical large computation times. We follow the practical super-individual approach (Scheffer et al., 1995).

That means that each model individual has one extra feature: the amount of population individuals that it actually represents. Theoretically, 10^{30} individuals can be represented by one super-individual.
However, in a spatially explicit model like Charisma, the use of super-individuals is limited, since the location is an important feature. Therefore, we use one super-individual per year class per grid cell. Furthermore, the use of super-individuals is optional and can be introduced gradually by setting the parameter *SplitNumber* which defines the number below which the super-individual splits in separate individuals.

Mortality is treated in a special way in super-individuals. Traditionally in individual based models, it is realized as the probability of dying each time step. This is done by drawing a random number and deciding from the value of this number to die or not to die (that is drawing from a binomial distribution with k=1). In super-individuals, this can be done repetitively while subtracting the number of dead individuals from the internal amount of the super-individual (that is equivalent to drawing from a binomial distribution with k = the internal number of the super-individual).

2. The grid

All plants in Charisma are associated with grid cells. Each grid cell has a seed bank and environmental variables associated with it (light, water level and bicarbonate).

The size of the grid is defined by four parameters:

GridLength, *GridWidth*
    The length and width of each cell in m.
NumX, *NumY*
    The numbers of cells in the x and the y direction.
Although these parameters may be changed during a run, it could lead to a loss of data and accuracy.

In spatial calculations, the grid cells with the maximum y coordinate are neighboring grid cells with the minimum y coordinate. The x-direction is used for gradients in environmental variables; the edges are not bordering on each other.

3. Vegetation

3.1 Overwintering structures

In winter the modeled vegetation can survive as shoots (optionally) and in the form of two types of overwintering structures, named *seed* (Characea: oospores) and *tubers* (Characea: bulbils). These structures are characterized by their individual weight (*SeedBiomass*, *TuberBiomass*).

The biomasses of these hibernacula are initiated to preset values (*SeedInitialBiomass*, *TuberInitialBiomass*).

The hibernacula may have a daily mortality (*SeedMortality*, *TuberMortality*) and a yearly import (*SeedImport*, *TuberImport*).

Yearly, the seeds are dispersed by each plant. The seed biomass is than subtracted from each plant. At a preset day a part of the seeds germinates.

3.1.1 Germination

At a preset day *GerminationDay* a part of the hibernacula begins to transform itself to young macrophytes. From that moment onwards, the growth of the sprout of the seedlings is incremented with a fixed daily percentage of the remaining hibernacula biomass (*cTuber* default 10%). Consequently, plants get an exponentially decreasing input from their hibernacula during early growth (cf. Hodgson, 1966).

The proportion of the overwintering structures that germinates is defined with a parameter (*SeedGermination*, *TuberGermination*). The remaining part stays in the seed bank and may germinate next year if it is not lost by mortality.

3.1.2 The number of seeds per plant

At a (preset) age (*SeedsStartAge*, *TuberStartAge*) the plants start allocating biomass for reproduction organs surviving the winter. At a second age (*SeedsEndAge*, *TuberEndAge*) the plants have allocated a fixed part of their biomass (*SeedFraction* or *Tuberfraction*). After this age the allocated **fraction** stays the same.
At the end of the growing season *ReproDay*, the allocated biomass is transformed into seeds and tubers and translocated in the seed bank.
The resulting number of hibernacula, and hence the density of young plants in the next spring, depends on the chosen size of the structures.

The total number of seeds or tubers that is produced by each plant is defined as:

$$N_j = \frac{a_s B}{b_s}$$

in which:
N_j total number of seeds/tubers dispersed by plant j (No yr^{-1})
a_s fraction of the plant biomass allocated to seeds (g yr^{-1}) *SeedFraction*
b_s individual biomass of a seed (g) *SeedBiomass*
B Total plant biomass (g).

Yearly, the seeds are dispersed by each plant. The seed biomass is then subtracted from each plant. It is distributed using the seed dispersal function.

3.1.3 Seed dispersal

The seeds can be dispersed over the grid in two ways:

1. The seeds are not dispersed to neighboring grid cells. The parameter *SeedRadius* should be 0 for this option.

2. The seeds are dispersed spatially around the ‘parent plants’ (see also: Van Nes & Scheffer, 1996) (*SeedRadius* > 0).

In the spatially explicit mode, we assume that seed dispersal leads to a normal distribution of seeds around the ‘parent plants’. The optimum of the distribution is at the parent plant. Because we use half a normal distribution, the area under the distribution is 0.5. Therefore, we use multiply the distribution with 2 to get a frequency distribution. So at distance D, twice a normal distribution is dispersed. We assume that the seeds are dispersed in all directions equally. Therefore, at longer distances from the plant, seeds also become more diluted in space. We correct by dividing the frequency distribution with the circumference of the circle with radius D. The fraction of seeds that is dispersed to a location in the grid is calculated with this three-dimensional normal distribution:

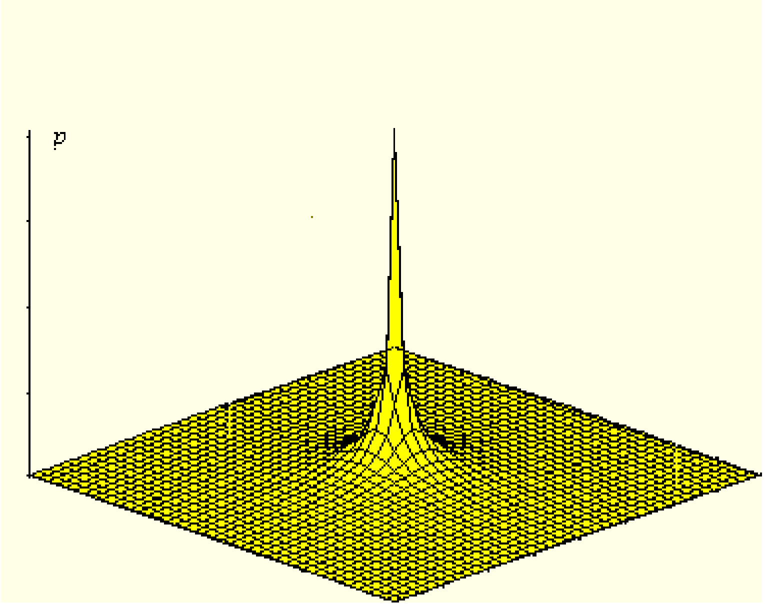
$$p_{i,j} = \frac{1}{2\pi\, D_{i,j}} \;\; 2\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{D_{i,j}^2}{2\sigma^2}}$$

In which:

$p_{i,j}$ proportion of the seeds dispersed from plant j to location i.

$D_{i,j}$ the distance between location i and the plant j (m).

σ the standard deviation of the normal distribution, determines the maximum dispersal radius. The parameter *SeedRadius* is 1.66 σ, which is the radius of the circle to which 90% of the seeds is dispersed.



Three-dimensional normal distribution for the spatial dispersal of seeds from one plant.

The seed that is dispersed to each grid cell is obtained by integrating the seed that is dispersed to each location in the grid cell. We use an effective and simple algorithm named Gaussian integration (Goudriaan, 1986).

In case that the grid is narrow, many of the seeds may drop outside the grid. In the y direction however the grid is rounded. That means that the grid cells with the maximum y coordinate are neighboring grid cells with the minimum y coordinate (see: 2.1). Therefore, the grid cells distribute seeds to a distance *SeedRadius* 'outside' the grid in the y direction.

3.2 Growth form

The root/shoot ratio is fixed on a user-defined level (*RootShootRatio*, default 0.1). The height of the young sprout increases proportionally with its biomass (*MaxWeightLenRatio*) until it reaches either the water surface or its maximum length (*MaxLength*). By default, further growth causes a proportional increase in biomass over the whole length axis of the plant. Optionally, however, the user can let part of the production be invested in shoots spreading just under the water surface *SpreadFrac*.

3.3 Respiration

Respiration can be divided into maintenance respiration and growth respiration. The latter is not treated separately here. Maintenance respiration is arbitrarily taken in the middle of the range of values published by Madsen and Adams (1989) and Ikusima (1970) for miscellaneous submerged plants (0.024 g g^{-1} d^{-1} at 20 °C). Temperature dependence of the respiration is formulated using a Q_{10} of 2:

$$R_m = r_{20}\, Q_{10}^{\frac{T-20}{10}}$$

Where:
R_m maintenance respiration (g g^{-1} d^{-1})
r_{20} respiration at 20 °C (g g^{-1} d^{-1}) *Resp20*
Q_{10} temperature factor *Q10*
T temperature (°C)

3.4 Primary production

Net production depends on photosynthesis and respiration. In the gross production, respiration is included. The maximum production rate (P_{max}) refers to the gross production. Only the shoots take part in the primary production, roots never (see: *RootShootRatio*), and the part of the plants allocated to seeds and tubers.

Photosynthesis in Charisma depends on:
1. in situ light (*l*)
2. temperature (*T*)

3. the distance (D) from the tissue to the top of the plant
4. bicarbonate concentration (C):
5. (optionally) limiting nutrient concentration (N):

$$P = P_{\max} f(I) f(T) f(D) f(C) f(N)$$

The parameter P_{max} *pMax* represents the specific daily production of the plant top at 20 °C not having light limitation. The default value (0.01 g g^{-1} h^{-1}) is arbitrarily chosen from a range of experimental data (Ikusima, 1970; Søndergaard, 1988) using cases where nutrients and carbon were presumably not limiting. Note that since nutrient limitation is not formulated explicitly in the model, P_{max} is the parameter with which the performance of a species under certain nutrient conditions can be set.

3.4.1 Effect of light on photosynthesis

The response of photosynthesis to light intensity (I) is formulated in a Monod fashion:

$$f(I) = \frac{I}{I + H_I}$$

H_I half-saturation constant (μE m^{-2} s^{-1}) *hPhotoLight*
I light intensity at plant leaves (μE m^{-2} s^{-1})
The default half-saturation constant for light (H_I=100 μE m^{-2} s^{-1}) is based on data from Søndergaard (1988) and Ikusima (1970).

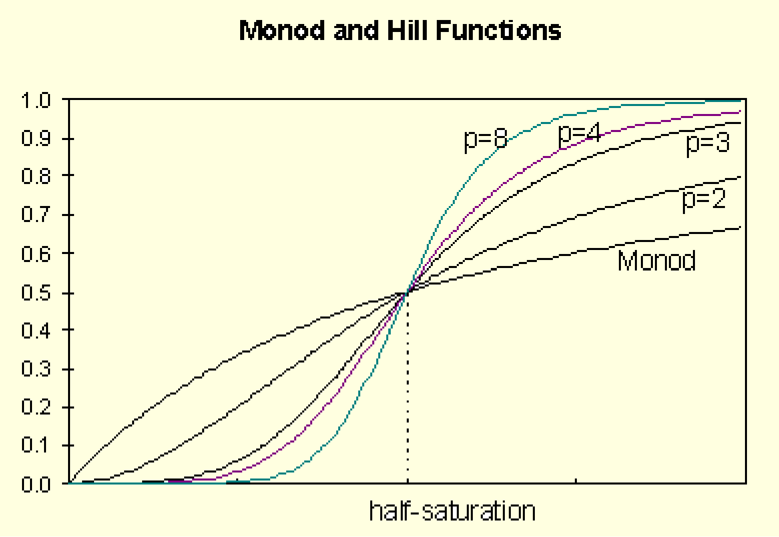
3.4.2 Temperature dependence of photosynthesis

Temperature dependence of photosynthesis is described by a Hill function, that should be fitted to data:

$$f(T) = S \frac{T^{p_T}}{T^{p_T} + H_T^{p_T}}$$

T Temperature of water (°C)
S Temperature factor (*sPhotoTemp* = 1.35, Scheffer et al., 1993)
p_T Power of Hill function (*pPhotoTemp* for *Potamogeton* fitted to 3 (Scheffer et al., 1993)
H_T Half-saturation of temperature effect (*hPhotoTemp* for *Potamogeton* fitted to 14)

Hill functions and Monod equations



Charisma makes extensively use of Hill functions and Monod equations. Monod equations (also referred to as *Michaelis-Menten equations*) are commonly used in ecology to describe resource limitation. These functions have only one parameter, the half-saturation coefficient, which indicates the resource concentration where growth is 50 % reduced.

Hill functions are a logical extension of Monod functions but less commonly used. Hill functions have one extra parameter, a power p. Hill functions provide a convenient way to describe a threshold or a transition from one state to another. If p is low (<3) it describes a gradual transition, if p is high it becomes a threshold function (see Figure). The Monod function is a special case of the Hill function, namely with p=1.

3.4.3 Effect of plant aging on photosynthesis

Photosynthesis efficiency is assumed to decrease with the distance from the plant top (*D*) due to the decrease in activity with tissue aging:

$$f(D) = \frac{H_D}{D + H_D}$$

H_D half-saturation of aging effect (m) *hPhotoDist*
The half-saturation distance (H_D=1 m) is set using data from Ikusima (1970).

3.4.4 Effect of bicarbonate concentration on photosynthesis

The photosynthesis is dependent on the available bicarbonate concentration. We assume that the effect can be described with a Monod function. For flexibility we modeled the Monod function as a Hill function with a power of 1. The parameters are to be fitted on data.

$$f(C)=\frac{C^{p_C}}{C^{p_C}+H_C^{p_C}}$$

C bicarbonate concentration (mg l^{-1})
H_C half-saturation of carbonate effect (mg l^{-1}) *hCarbonate*
p_C power of Hill function *pCarbonate*

3.4.5 Effect of concentration of limiting nutrient on photosynthesis

The photosynthesis can optionally made dependent on the available nutrient concentration (limiting nutrient, either P or N). However, macrophytes in most eutrophic lakes this will not be limited by nutrients (then *hNutrient* should be NAN). We assume that the effect can be described with a Monod function. For flexibility we modeled the Monod function as a Hill function with a power of 1. The parameters are to be fitted on data.

$$f(N)=\frac{N_N^{p_N}}{N^{p_N}+H_N^{p_N}}$$

N nutrient concentration (mg l^{-1})
H_N half-saturation of nutrient effect (mg l^{-1}) NAN is no effect *hNutrient*
P_N power of Hill function *pNutrient*

3.5 Mortality factors

Four mortality causes are explicitly included in the model: wave damage, losses due to herbivory, mortality due to competition at high plant densities and seasonal die-off. Moreover a fixed background mortality can optionally be used *BackGroundMort*. Obviously a negative growth rate leads also to mortality.

Mortality in macrophytes can lead to loss in number of plants or a loss in individual weight. Since the concept of individual is somewhat obscure for vegetation, we use the following pragmatic approach:

- negative growth always leads to a loss in number of plants.
- herbivory leads to a loss in individual weight only.
- background mortality and wave mortality lead to a loss in numbers if the plants have not yet reached the water surface. Adult plants only lose weight.

Note that the balance between the number of plants and their individual weight is only essential for young sprouts since that balance determines the sprout length.

3.5.1 The thinning law

Mortality due to competition at high plant densities can optionally (set value of parameter *Thinning* to *True*) follow the thinning law (see review by Westoby, 1984). We use the thinning law only to balance the number of plants and their individual weight; it does not lead to a loss of weight.

$$N^* = \left(\frac{5950}{W}\right)^{2/3}$$

Simultaneously, the individual weight is adjusted as follows:

$$W^* = \frac{N}{N^*}W$$

N^* adjusted number of individuals (m^{-2})
W^* adjusted individual weight (g)
N number of individuals (m^{-2})
W individual weight (g)

3.5.2 Wave mortality

Losses due to wave damage are maximal at the shoreline and decrease with rooting depth following a Hill function with a default half-saturation depth of 0.15 m:

$$M_{wave} = M_{\max} \frac{H_m{}^{p_m}}{H_m{}^{p_m} + D^{p_m}}$$

M_{wave} wave mortality (d^{-1})
M_{max} maximum wave mortality along the shoreline (d^{-1}) *MaxWaveMort*
H_m half saturation depth for wave mortality (m) *HWaveMort*
p_m exponent in the Hill function *pWaveMort*
D water depth (m)

3.6 Grazing

Two types of grazing are implemented in the model:

1. Grazing of sprouts by an added grazer/harvester
2. Winter grazing of tubers

3.6.1 Grazing of sprouts

In the model you can add a grazer/harvester, which can graze all species with several strategies (functional responses). Note that you first have to add the species “Grazer” before grazing strategies can be set. The amount of plants that is consumed by the grazer is saved as rate and cumulative per year. The parameter *ResetCumConsDay* is the day that the cumulative consumption is reset to the value 0.

Functional responses (*Strategy*):

***block*:**

Grazing during the growing season is simulated by setting a fixed grazing rate *Maxgrazing* in g m^{-2} representing, for instance, the consumption capacity of the local coot population, and a limit in terms of vegetation density below which grazing ceases.

The parameter *GrazingThres* sets that limit. Set this parameter to NAN (not available number) if there is no grazing.

Holling type 1 functional response:

This functional response is implemented as linearly increasing with vegetation biomass:

$$\frac{dB}{dt} = g\ B$$

In which:

B Vegetation biomass (g m^{-2})

g Relative grazing *RelativeGrazing* (d^{-1})

t time (d)

Holling type 2 functional response:

A Holling type 2 functional response is a saturation function, which means that the grazing rate is increasing with vegetation till it reaches a maximum.

This functional response is implemented as a Monod equation:

$$\frac{dB}{dt} = g_{\max} \frac{V}{V + H_V}$$

B Vegetation biomass (g m^{-2})

g_{max} Maximal grazing rate *MaxGrazing* (g m^{-2} d^{-1})

H_v Vegetation biomass with half the maximum grazing rate *HGrazing* (g m^{-2})

t time (d)

Holling type 3 functional response:

A Holling type 2 functional response resembles type 2 but is now sigmoidal.

This functional response is implemented as a Hill function:

$$\frac{dB}{dt} = g_{\max} \frac{V^p}{V^p + H_V^p}$$

B Vegetation biomass (g m^{-2})

g_{max} Maximal grazing rate *MaxGrazing* (g m^{-2} d^{-1})

H_v Vegetation biomass with half the maximum grazing rate *HGrazing* (g m^{-2})

p exponent of the Hill function *PGrazing*

t time (d)

Harvesting actions:

With this strategy the plants are grazed down (or harvested) to a certain height in a discrete number of events per year. The number of actions is determined by the parameter *NHarvestActions*. The harvest days and the remaining heights of the macrophytes is determined by the parameter *HarvestActions*.

3.6.2 Winter grazing on tubers

In winter hibernacula can be grazed down to a fixed density, as observed for instance for bewicks swans (*Cygnus columbianus* L.) foraging on tubers of *Potamogeton pectinatus* L. (Van Eerden pers. comm.). This winter grazing is mimicked by setting the density to which hibernacula are reduced in winter. The parameters that control winter grazing (*SeedGrazingThres* and *TuberGrazingThres*) are not features of the grazer but of the separate species.

3.7 Seasonal die-off

Seasonal die-off is the most radical mortality. It is simply modeled as a maximum age in days of the plants (parameter: *MaxAge*).

4. Environment

4.1 Light

Charisma needs much information about light, as photosynthesis is the important process. Irradiation data are needed at any time of the day to integrate the total daily photosynthesis. We use total daily irradiation as basic information and use general equations to approximate the irradiation at any time of the day. Further, we need to calculate what part of the total light intensity reaches the leaves of the plants at any place below the water surface (see: effective irradiation).

The daily total irradiation follows a sine wave over the year fitted to data for Dutch shallow lakes:

$$I_{total} = I_{\max} - \frac{I_{\max} - I_{\min}}{2}\left[1 + \cos\left(\frac{2\pi}{365.25}(d - I_{delay})\right)\right]$$

I_{total} Total daily irradiation at surface of water (μE m^{-2} s^{-1}).
I_{max} Maximum irradiation (μE m^{-2} s^{-1}) *MaxI*
I_{min} Minimum irradiation (μE m^{-2} s^{-1}) *MinI*
I_{delay} no of days after January 1 where irradiation is minimal *IDelay*
d no of days after January 1.

Alternatively, the irradiation data can be read from a file (parameters: *IDataFile*, *IFileColumn*, *IDrawOutside*, *IPermuteMode*).

The irradiation at any time of the day is a function of the day length (Kirk, 1983):

$$I_t = \frac{\pi I_{total}}{2N} \sin\left(\frac{\pi t}{N}\right)$$

$I_{,t}$ irradiation just above the water surface (μE m^{-2} s^{-1})
t time in hours after sunrise (hr)
N day length in hours (hr)

The day length N can be calculated from the latitude and the time of the year (Kirk, 1983)

$$N = 0.133 \cos^{-1}(-\tan\gamma \tan\delta)$$

where:

δ solar declination at day ψ (= 0.39637-22.9133 cos ψ+4.02543 sin ψ - 0.3872 cos 2ψ + 0.052 sin 2ψ degrees) day number ψ in radians.
γ latitude in radians *Latitude*

4.2 The effective irradiation

From the total light that enters the water column only a small fraction reaches the macrophytes. The photosynthetic active fraction (PAR) is assumed to be 50% of the total irradiation (parameter: *PARFactor*. Light reflection at the water surface is about 10% *FracReflected*. Furthermore the user can change the total irradiation using a deviation factor *SunDev*:

$$I_{surf,t} = f_{PAR}\, f_{reflect} f_{dev}\, I_t$$

In the water column the light attenuation in the water column follows the Lambert-Beer law. In addition, in situ light is affected by self-shading. The formulation is similar to the Lambert-Beer law, with the specific light attenuation coefficient of plant material set to 0.01 m^{2}g^{-1} (Ikusima, 1970):

$$I_{z,t} = I_{surf,t}\, e^{-K_d z - K_p b_{<z}}$$

where

$I_{z,t}$ irradiation at time t and depth z (μE m^{-2} s^{-1})
K_d vertical light attenuation coefficient of the water (m^{-1})
z distance from the water surface (m)
K_p specific light attenuation coefficient of plant material (m^{2}g^{-1}) *PlantK*
$b_{<Z}$ biomass of plant material above z (g m^{-2})

Additional, a fixed proportion of the light (default 20%) is attenuated by periphyton *fracPeriphyton*.

4.3 Vertical light attenuation of the water

The vertical light attenuation in the water is modeled as an external value that can be reduced by vegetation. A simple way of mixing can be added too.

The ligth attenuation can be read from a file with daily water extinctions. Alternatively, the extinction can be modeled with a cosine function:

$$K_d = K_{dev}\left[K_{\max} - \frac{K_{\max} - K_{\min}}{2}\left(1 + \cos(\frac{2\pi}{365.25}(day - K_{lag})\right)\right]$$

in which:

K_d External light attenuation coefficient (extinction coefficient), that is the light attenuation without the effect of vegetation on turbidity (m^{-1})
K_{max} Maximum light attenuation coefficient (m^{-1}) *MaxKd*
K_{min} Minimum light attenuation coefficient (m^{-1}) *MinKd*
K_{lag} Delay, the day number with the minimal light attenuation coefficient (d) *KdDelay*
K_{dev} Deviation factor, a factor between 0 and 1 to change the whole light attenuation range (only to be used in analysis) *KdDev*
day Number of days after the first of January (d)

Note: as a convenient alternative the cosine can also be defined by K_{mean} (*Kd* and K_{range} (*KdRange*), which define the mean light attenuation and the range of light attenuations respectively.

The parameter *KdDataFile* determines if data are read from a file. The parameter should contain the file name or NAN if no files are read. The format of the data files is flexible. The way that data are read from the file is determined by the following parameters:

1. *KdFileColumn* is the column that contains light attenuation coefficients. The first column always contains the dates. If the light attenuation data are in the second column then the parameter is 1.
2. *KdDrawOutside* determines what happens outside the scope of the data file. If this parameter is *False*, the cosine function is used outside the scope of the data file, else random years are drawn (with replication) from the data file. Only full years are drawn.

3. *KdPermuteMode* determines what happens inside the scope of the data file. This parameter can have three values:

a) *RealData* the real data are used and remain unchanged.
b) *PermuteYears* the years in the file are permuted (drawn without replication)
c) *DrawYears* the years are resampled with replication

If one file is used also for other parameters (in other columns), the file is only read once in the computer memory. Then, the same years are drawn for these parameters, so all resampling is then linked. If this behavior is not wished, use copies of the data file.

Optionally a clear water period may be defined. That is a period (mostly in spring) in which the light attenuation coefficient (either entered as cosine or file) is lowered temporally.
The start of that period is defined as a fixed daynumber (*Clear-water timing*)
The parameter *ClearWaterFraction* defines how much the light attenuation is lowered, as a fraction of the 'normal' light attenuation.
The clear water period as a fixed length (*Clear-water period*). If there is no clear water period, this parameter should be 0.

4.3.1 The effect of vegetation on the light attenuation

Many studies show that plants reduce turbidity. Although there can be several different mechanisms be involved, the general pattern is clear (Scheffer, 1997). We use the Hill equation to describe the reduction of the light attenuation coefficient within vegetation in each grid cell:

$$K_p = K_b + (K_d - K_b)\frac{{H_K}^{p_K}}{B^{p_K} + {H_K}^{p_K}}$$

K_p adjusted extinction coefficient (m^{-1})
K_d the 'external' extinction coefficient of water (m^{-1})
K_b background extinction *BackgrKd* (m^{-1}).
H_K half-saturation coefficient of extinction reduction by plant biomass *HTurbReduction* ($g\ m^{-2}$)
p_K power of extinction reduction by plant biomass *pTurbReduction* ($g\ m^{-2}$)
B total plant biomass ($g\ m^{-2}$)

The equation shows that vegetation cannot reduce light attenuation below the background light attenuation. K_p is calculated daily for each grid cell.

4.3.2 Mixing the light attenuation coefficient in grids

Mixing between grid cells can be mimicked in a simple way. The actual light attenuation coefficient is calculated by averaging the light attenuation coefficients of neighbouring cells. The number of cells that is used in the running average is determined by the mixing length *KdDiffusion*.

4.4 Temperature

The temperature can be read from a file with daily water temperatures. Alternatively, the temperature can be modeled with a cosine function:

$$T = T_{dev}\left[T_{\max} - \frac{T_{\max} - T_{\min}}{2}\left(1 + \cos(\frac{2\pi}{365.25}(day - T_{lag}))\right)\right]$$

in which:

T Temperature (°C)
T_{max} Maximum temperature (°C) *TempMax*
T_{min} Minimum temperature (°C) *TempMin*
T_{lag} Temperature lag (d) *TempLag*
T_{dev} Deviation factor, a factor between 0 and 1 to change the whole temperature range (only to be used in analysis) *TempDev*
day Number of days after the first of January (d)

The parameter *DataFile* determines if data are read from a file. The parameter should contain the file name or NAN if no files are read. The format of the data files is flexible (see box).
The way that data are read from the file is determined by the following parameters:

1. *TempFileColumn* is the column that contains temperature. The first column always contains the dates. If the temperature data are in the second column then the parameter is 1.
2. *TempDrawOutside* determines what happens outside the scope of the data file. If this parameter is *False*, the cosine function is used outside the scope of the data file, else random years are drawn (with replication) from the data file. Only full years are drawn.
3. *TempPermuteMode* determines what happens inside the scope of the data file. This parameter can have three values:
   a) RealData the real data are used and remain unchanged.
   b) PermuteYears the years in the file are permuted (drawn without replication)
   c) DrawYears the years are resampled with replication

If one file is used also for other parameters (in other columns), the file is only read once in the computer memory. Then, the same years are drawn for these parameters, so all resampling is then linked. If this behavior is not wished, use copies of the data file.

4.5 The water depth

Water depth at any location in the model is calculated by subtracting the level of the grid from the water level. Therefore, both values should have the same reference point.

4.6 The level of the grid

The level of the grid is the most important parameter of the grid. There are different ways to enter the levels. The value of the parameter *LeveInputType* defines the way in which the data is entered. This parameter can have 3 values:

- *Xgradient* – Use this method for input of a gradient of levels in the x-direction of the grid. To create a depth profile in the x direction of the grid, two parameters are used: *nPoints* and *Profile*. *Profile* is an array of records that describe the profile. Each record contains the **relative** position in the x-direction (0 = minimum, 1 = maximum) and the level of the grid in m. The parameter *nPoints* sets the number of points in the profile.

- *Whole grid* – Use this method for manually input of the depths of a grid. All values are then entered in an array representing the whole grid (*LevelGrid*).

- *File* – Not yet implemented. In future it will be possible to read the level of the grid from a file.

4.7 Water level

The water level values in the model are generated in a very similar way as the temperature values. Values can either be read from a file with water levels or the following cosine function:

$$W = W_{corr} + W_{dev}\left[W_{\max} - \frac{W_{\max} - W_{\min}}{2}\left(1 + \cos(\frac{2\pi}{365.25}(day - W_{lag}))\right)\right]$$

in which:

W Water level (m)
W_{max} Maximum water level (m) *MaxW*
W_{min} Minimum water level (°C) *MinW*
W_{lag} Water level lag (d) *WDelay*
W_{corr} Correction for reference level of the water level data (m) *LevelCorrection*
W_{dev} Deviation factor, a factor between 0 and 1 to change the whole water level range (only to be used in analysis) *WDev*
day Number of days after the first of January (d)

The parameter *WDataFile* determines if data are read from a file. The parameter should contain the file name or NAN if no files are read. The format of the data files is flexible (see box).
The way that data are read from the file is determined by the following parameters:

1. *WFileColumn* the column that contains the water level. The first column always contains the dates. If the dates are in the second column *WfileColumn* = 1.

2. *WDrawOutside* determines what happens outside the scope of the data file. If this parameter is *False*, the cosine function is used outside the scope of the data file, else random years are drawn (with replication) from the data file. Only full years are drawn.
3. *WPermuteMode* determines what happens inside the scope of the data file. This parameter can have three values:
   a) *RealData* the real data are used and remain unchanged.
   b) *PermuteYears* the years in the file are permuted (drawn without replication)
   c) *DrawYears* the years are resampled with replication

If one file is used also for other parameters (in other columns), the file is only read once in the computer memory. Then, the same years are drawn for these parameters, so all resampling is then linked. If this behavior is not wished, use copies of the data file.

4.8 Bicarbonate

The bicarbonate concentration (HCO_3^-) is simply modeled as a fixed value *MaxCarbonate* that can be reduced if there is vegetation. This reduction of the bicarbonate concentration is modeled as a simple Monod function, analogous to the reduction of the light attenuation coefficient:

$$C_p = C_{\max} \frac{H_C}{B + H_C}$$

C_p adjusted bicarbonate concentration (mg l^{-1})
C_{max} basic bicarbonate concentration (mg l^{-1}) *MaxCarbonate*
H_K half-saturation coefficient of bicarbonate concentration by plant biomass *HCarboReduction* (g m^{-2})
B total plant biomass (g m^{-2})

4.9 Limiting nutrient

The limiting nutrient concentration (P or N) is simply modeled as a fixed value *MaxNutrient* that can be reduced if there is vegetation. This reduction of the nutrient concentration is modeled as a simple Monod function, analogous to the reduction of the light attenuation coefficient:

$$N_p = N_{\max} \frac{H_N}{B + H_N}$$

N_p adjusted nutrient concentration (mg l^{-1})
N_{max} basic nutrient concentration (mg l^{-1}) *MaxNutrient*
H_K half-saturation coefficient of nutrient concentration by plant biomass *HNutrReduction* (g m^{-2})
B total plant biomass (g m^{-2})

5. References

De Hoop, B.J., P.M.J. Herman, H. Scholten, & K. Soetaert, 1992. SENECA 1.5. A simulation environment for ecological application. Manual. Netherlands Institute of Ecology 180 pp.

Goudriaan, J., 1986. A simple and fast numerical method for the computation of daily totals of crop photosynthesis. Agr.Forest Meteorol. 38: 249-254.

Hodgson, R.H., 1966. Growth and carbohydrate status of Sago pondweed. Weeds. 14: 263-268.

Ikusima, I., 1970. Ecological studies on the productivity of aquatic plant communities IV. Light condition and community photosynthetic production. Bot.Mag.-Tokyo. 83: 330-341.

Kirk, J.T.O., 1983. Light and photosynthesis in aquatic ecosystems. Cambridge University Press. Cambridge. 401 pp.

Klepper, O., 1989. A model of carbon folws in relation to macrobenthic food supply in the Oosterschelde estuary (S.W. Netherlands). Thesis LUW Wageningen, The Netherlands.

Klepper, O. & E.M.T. Hendrix, 1994. A comparison of algorithms for global characterization of confidence regions for nonlinear models. Environmental Toxicology and Chemistry. 13: 1887-1899.

Madsen, J.D. & M.S. Adams, 1989. The light and temperature dependence of photosynthesis and respiration in *Potamogeton pectinatus* L. Aquat.Bot. 36: 23-31.

Price, W.L., 1979. A Controlled Random Search procedure for global optimization. The Computer Journal. 20: 367-370.

Scheffer, M., 1998. Ecology of Shallow Lakes. Chapman and Hall. New York.

Scheffer, M., A.H. Bakema, & F.G. Wortelboer, 1993. MEGAPLANT - A simulation model of the dynamics of submerged plants. Aquat.Bot. 45: 341-356.

Scheffer, M., J.M. Baveco, D.L. DeAngelis, K.A. Rose, & E.H. Van Nes, 1995. Super-individuals a simple solution for modelling large populations on an individual basis. Ecol.Model. 80: 161-170.

Søndergaard, M., 1988. Photosynthesis of aquatic plants under natural conditions. In: Symoens, J.J.(Ed.) Vegetation of Inland Waters. Kluwer Academic. 63-113.

Van Nes, E.H. & M. Scheffer, 1996. ArtiVeg 1.0, a tool for spatial explicit, individual based vegetation modelling. RIZA Werkdocument 96.064X. 19 pp.

Westoby, M., 1984. The self-thinning rule. Adv.Ecol.Res. 14: 167-255.