N2PZDQ

Generated by Doxygen 1.8.4

March 10, 2014

Contents

Chapter 1

N2PZDQ: 2-Nutrients, Quota resolving NPZD model

1.1 General Overview

Description

State variables

N2PZDQ model resolves:

- 2 nutrients, nitrogen and phosphorus in dissolved and detrital form
- Phytoplankton with flexible C:N:P stoichiometry
- Zooplankton with fixed stoichiometry

Fluxes

The NPZD (nutrient-phytoplankton-zooplankton-detritus) model described here consists of $I=4$ state variables. Nutrient uptake (phytoplankton growth) is limited by light and nutrient availability, the latter of which is modelled by means of Michaelis-Menten kinetics, see eq. (dnp}). The half-saturation nutrient concentration $$ used in this formulation has typically a value between 0.2 and 1.5 mmol N\, m^{-3}. Zooplankton grazing which is limited by the phytoplankton standing stock is modelled by means of an Ivlev formulation, see eq. (dpz}). All other processes are based on linear first-order kinematics, see eqs. (dpn}) - (dzd}).

For all details of the NPZD model implemented here, see**[Burchardetal2005b]**}.

Here is a diagram of fluxes:

Conventions

For the sake of readibility of the formulas, conventional symbols (e.g., Greek letters, sub and superscripts, etc) are used. For the correspondance between the symbols and parameters in the model code, see the Nomenclature section at the end of this report.

```
maybe some script here
```

1.2 References

-?

Chapter 2

Data Type Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Data Type Index

3.1 Data Types List

Here are the data types with brief descriptions:

Chapter 4

Data Type Documentation

4.1 fabm_hzg_n2pzdq Module Reference

This modeule describes an NPZD model extended with 2 nutrients and variable stoichiometry of phyto.

Data Types

- type type_hzg_n2pzdq

    This is the derived model type.

Public Member Functions

- subroutine initialize (self, configunit)

    here the n2pzdq namelist is read,variables exported by the model are registered in FABM and variables imported from FABM are made available

Private Member Functions

- subroutine do (self, _ARGUMENTS_DO_)

    This is the main routine where right-hand-sides are calculated.
- subroutine get_light_extinction (self, _ARGUMENTS_GET_EXTINCTION_)

    to calculate light extinction when kc chnages with depth
- subroutine fprod (self, par, temp_fact, qnc, qpc, primprod, Nlim, Plim, Llim)

    subroutine: primary production
- pure real(rk) function fupn (self, DIN, qnc)

    nitrogen uptake function
- pure real(rk) function fupp (self, DIP, qpc)

    phosphorus uptake function

4.1.1 Detailed Description

This modeule describes an NPZD model extended with 2 nutrients and variable stoichiometry of phyto.

Author

Lena Spruch, Kai Wirtz, Onur Kerimoglu

Copyright

HZG

See Section 1 for a general overview to see what the model is about.

Definition at line 11 of file n2pzdq.F90.

4.1.2 Member Function/Subroutine Documentation

4.1.2.1 fabm_hzg_n2pzdq::do (class (type_hzg_n2pzdq), intent(in) *self*, *_ARGUMENTS_DO_*) [private]

This is the main routine where right-hand-sides are calculated.

Here details about specific processes are provided.

1. Phytoplankton processes: -Phyto quatas are calculated as: $phy_{QN} = phy_N / phy_C$, $phy_{QP} = phy_P / phy_C$
   - Production: $phy_{prod}, Nlim, Plim$ is obtained by calling fprod()
   - N-uptake: $V_N =$ fupn() function is called
   - P-uptake: $V_P =$ fupp() function is called
   - Mortality: $r_{pd} = \text{mortpar\_phy} * e^{(-\text{mortpar\_phy}*Nlim,Plim)} * det_N$
2. Remineralisation of detritus into nutrients:
   - r_{dn,N} rem_N∗temp_fact
   - r_{dn,P}= rem_P∗temp_fact
3. Zooplankton processes:
   - $G = G_{max} * (1 - e^{-\text{self\%iv}*phy_C^2}) * temp\_fact$
   - $r_{zd} = (\text{mort\_zoo} * zoo_C^n + \text{mort\_zoo2} * zoo_C^n 2) * \text{temp\_fact}$
   - calc X,C assimilation efficiencies: e_X, ee_C as a function of zoo_{QX}, phy_{QX}:

```
e_n = self%e_C*self%qzn/qnc
e_p = self%e_C*self%qzp/qpc
ee_c = self%e_C

if (max(e_n,e_p) > 1) then
  e_n = 1
  ee_c = qnc/self%qzn
  e_p = ee_c*self%qzp/qpc
 if (e_p > 1) then
   e_p = 1
   ee_c = qpc/self%qzp
   e_n = ee_c*self%qzn/qnc
  if (e_n > 1) then
    e_n = 1
    ee_c = qnc/self%qzn
    e_p = ee_c*self%qzp/qpc
  end if
 end if
end if
```

   - $zoo_{exc,detX} = \gamma * (1 - e_X) * phy_{QX} * G$
   - $zoo_{exc,nutX} = (1 - \gamma) * (1 - e_X) * phy_{QX} * G$
4. Right-hand sides:
   - $d(N) = r_{dn,N} * det_N - V_N * phy_C + zoo_{exc,nutN} * zoo_C$
   - $d(P) = r_{dn,P} * det_P - V_P * phy_C + zoo_{exc,nutP} * zoo_C$
   - $d(phy_C) = phy_{prod} * phy_C - r_{pd} * phy_C - G * zoo_C$
   - $d(phy_N) = V_N * phy_C - r_{pd} * phy_N - G * zoo_C * phy_{QN}$
   - $d(phy_P) = V_P * phy_C - r_{pd} * phy_P - G * zoo_C * phy_{QP}$

- $d(det_N) = r_{pd} * phy_N + r_{zd} * zoo_C * zoo_{QN} - r_{dn,N} * det_N + zoo_{exc,det_N} * zoo_C$
- $d(det_P) = r_{pd} * phy_P + r_{zd} * zoo_C * zoo_{QP} - r_{dn,N} * det_N + zoo_{exc,det_P} * zoo_C$
- $d(zoo_C) = G * zoo_C * ee_C - r_{zd} * zoo_C$

Definition at line 280 of file n2pzdq.F90.

4.1.2.2 subroutine fabm_hzg_n2pzdq::get_light_extinction (class (type_hzg_n2pzdq), intent(in) *self,* *_ARGUMENTS_GET_EXTINCTION_*) [private]

to calculate light extinction when kc chnages with depth

get_light: some more description here?

Definition at line 451 of file n2pzdq.F90.

4.1.2.3 subroutine fabm_hzg_n2pzdq::fprod (type (type_hzg_n2pzdq), intent(in) *self,* real(rk), intent(in) *par,* real(rk), intent(in) *temp_fact,* real(rk), intent(in) *qnc,* real(rk), intent(in) *qpc,* real(rk), intent(out) *primprod,* real(rk), intent(out) *Nlim,* real(rk), intent(out) *Plim,* real(rk), intent(out) *Llim*) [private]

subroutine: primary production

- Light limitation, $Llim = (-\alpha * par)/\sqrt{grow_max^2 + \alpha^2}$
- N-limitation, $Nlim = 1 - qmin_N/phy_{QN}$
- N-limitation, $Plim = 1 - qmin_P/phy_{QP}$
- primary production, $phy_{prod} = P_{max} * min(Nlim, Plim) * Llim * \text{temp\_fact}$

Definition at line 489 of file n2pzdq.F90.

4.1.2.4 pure real(rk) function fabm_hzg_n2pzdq::fupn (type (type_hzg_n2pzdq), intent(in) *self,* real(rk), intent(in) *DIN,* real(rk), intent(in) *qnc*) [private]

nitrogen uptake function

Process description: N-uptake (forced to stay above 0)

- Q-dependent uptake regulation term, $q_N = (1 - (phy_{QN} - Q_{min,N})/(Q_{max,N} - Q_{min,N}))$
- $V_N = max(0, V_{max,N} * q_N) * DIN/(DIN + K_N)$

Definition at line 514 of file n2pzdq.F90.

4.1.2.5 pure real(rk) function fabm_hzg_n2pzdq::fupp (type (type_hzg_n2pzdq), intent(in) *self,* real(rk), intent(in) *DIP,* real(rk), intent(in) *qpc*) [private]

phosphorus uptake function

Process description: P-uptake (forced to stay above 0)

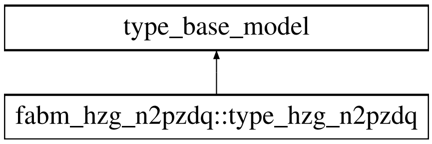
- Q-dependent uptake regulation term, $q_P = (1 - (phy_{QP} - Q_{min,P})/(Q_{max,P} - Q_{min,P}))$
- $V_P = max(0, V_{max,P} * q_N) * DIP/(DIP + K_P)$

Definition at line 531 of file n2pzdq.F90.

4.2 fabm_hzg_n2pzdq::type_hzg_n2pzdq Type Reference

This is the derived model type.

Inheritance diagram for fabm_hzg_n2pzdq::type_hzg_n2pzdq:



4.2.1 Detailed Description

This is the derived model type.

Definition at line 25 of file n2pzdq.F90.

Index