

N2PZDQ

Generated by Doxygen 1.8.4

February 25, 2014

# Contents

<b>1</b>	<b>N2PZDQ: 2-Nutrients, Quota resolving NPZD model</b>	<b>2</b>
1.1	General Overview . . . . .	2
1.2	References . . . . .	2
<b>2</b>	<b>Todo List</b>	<b>3</b>
<b>3</b>	<b>Data Type Index</b>	<b>4</b>
3.1	Class Hierarchy . . . . .	4
<b>4</b>	<b>Data Type Index</b>	<b>5</b>
4.1	Data Types List . . . . .	5
<b>5</b>	<b>File Index</b>	<b>6</b>
5.1	File List . . . . .	6
<b>6</b>	<b>Data Type Documentation</b>	<b>7</b>
6.1	fabm_hzg_n2pzdq Module Reference . . . . .	7
6.1.1	Detailed Description . . . . .	7
6.1.2	Member Function/Subroutine Documentation . . . . .	8
6.1.2.1	do . . . . .	8
6.1.2.2	fprod . . . . .	8
6.1.2.3	fupn . . . . .	8
6.1.2.4	fupp . . . . .	9
6.1.2.5	get_light_extinction . . . . .	9
6.1.2.6	initialize . . . . .	9
6.2	fabm_hzg_n2pzdq::type_hzg_n2pzdq Type Reference . . . . .	9
6.2.1	Detailed Description . . . . .	11
<b>7</b>	<b>File Documentation</b>	<b>12</b>
7.1	n2pzdq.F90 File Reference . . . . .	12
7.1.1	Detailed Description . . . . .	12

<b>Nomenclature</b>	<b>13</b>
<b>Index</b>	<b>14</b>

# 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  $K_N$  used in this formulation has typically a value between 0.2 and 1.5 mmol N l<sup>-1</sup>, m<sup>-3</sup>. 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 readability 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

## Todo List

Subprogram `fabm_hzg_n2pzdq::initialize` (self, configunit)  
here a more detailed description can be provided

## Chapter 3

# Data Type Index

### 3.1 Class Hierarchy

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

fabm_hzg_n2pzdq . . . . .	7
type_base_model	
fabm_hzg_n2pzdq::type_hzg_n2pzdq . . . . .	9

## Chapter 4

# Data Type Index

### 4.1 Data Types List

Here are the data types with brief descriptions:

<a href="#">fabm_hzg_n2pzdq</a>	This modeule describes an NPZD model extended with 2 nutrients and variable stoichiometry of phyto . . . . .	7
<a href="#">fabm_hzg_n2pzdq::type_hzg_n2pzdq</a>	This is the derived model type . . . . .	9

## Chapter 5

# File Index

### 5.1 File List

Here is a list of all documented files with brief descriptions:

<b>mainpage.doxygen</b> . . . . .	<b>??</b>
<a href="#">n2pzdq.F90</a> This file contains the <a href="#">fabm_hzg_n2pzdq</a> module . . . . .	<a href="#">12</a>



## Chapter 6

# Data Type Documentation

### 6.1 fabm\_hzg\_n2pzdq Module Reference

This module 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 changes 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*

#### 6.1.1 Detailed Description

This module 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.

## 6.1.2 Member Function/Subroutine Documentation

**6.1.2.1** `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.

Zooplankton processes:  $G = d_{zd} = r_{zd}C_z$ .

Phytoplankton processes:

Phyto quotas are calculated as:  $Q_N = P_N/P_C$ ,  $Q_P = P_P/P_C$

Production:  $phy_{prod}, Nlim, Plim$  is obtained by calling `fprod`

Mortality:  $phy_{mort} = phy_{mort0} * e^{(-mort_{par} * phy * Nlim, Plim)} * det_N$

N-uptake: `fupN` function is called

P-uptake: `fupP` function is called.

Here details about specific processes are provided.

Definition at line 284 of file n2pzdq.F90.

References `fprod()`, `fupn()`, and `fupp()`.

**6.1.2.2** `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,  $Llim = (-\alpha * par) / \sqrt{grow_m ax^2 + \alpha^2}$

N-limitation,  $Nlim = 1 - qmin_N / qnc$

N-limitation,  $Plim = 1 - qmin_P / qpc$

primary production,  $primprod = rmax * min(Nlim, Plim) * Llim * temp_{fact}$

Definition at line 474 of file n2pzdq.F90.

Referenced by `do()`.

**6.1.2.3** `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: quota-dependent regulation of uptake rate (forced to stay above 0) times the limitation dependent on external concentration

$fupN = \max(0, upmax_N (1 - (Q_N - Q_{min}) / (Q_{max_N} - Q_{min_N})) * DIN / (DIN + K_N))$

Definition at line 499 of file n2pzdq.F90.

Referenced by `do()`.

6.1.2.4 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

Calculated as:  $fupP = \max(x, y)$

Definition at line 515 of file n2pzdq.F90.

Referenced by do().

6.1.2.5 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 436 of file n2pzdq.F90.

6.1.2.6 subroutine fabm\_hzg\_n2pzdq::initialize ( class (type\_hzg\_n2pzdq), intent(inout), target *self*, integer, intent(in) *configunit* )

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

**Todo** here a more detailed description can be provided

Definition at line 61 of file n2pzdq.F90.

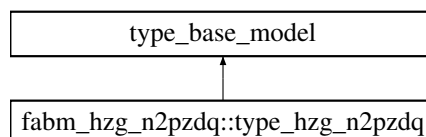
The documentation for this module was generated from the following file:

- [n2pzdq.F90](#)

## 6.2 fabm\_hzg\_n2pzdq::type\_hzg\_n2pzdq Type Reference

This is the derived model type.

Inheritance diagram for fabm\_hzg\_n2pzdq::type\_hzg\_n2pzdq:



### Public Member Functions

- procedure **initialize**
- procedure **do**
- procedure **get\_light\_extinction**

### Public Attributes

- type(type\_state\_variable\_id) **id\_din**

- type(type\_state\_variable\_id) **id\_dip**
- type(type\_state\_variable\_id) **id\_phyc**
- type(type\_state\_variable\_id) **id\_phyn**
- type(type\_state\_variable\_id) **id\_phyp**
- type(type\_state\_variable\_id) **id\_detn**
- type(type\_state\_variable\_id) **id\_detp**
- type(type\_state\_variable\_id) **id\_zooc**
- type(type\_state\_variable\_id) **id\_dic**
- type(type\_dependency\_id) **id\_par**
- type(type\_dependency\_id) **id\_temp**
- type(type\_horizontal\_dependency\_id) **id\_i\_0**
- type(type\_diagnostic\_variable\_id) **id\_gpp**
- type(type\_diagnostic\_variable\_id) **id\_ncp**
- type(type\_diagnostic\_variable\_id) **id\_ppr**
- type(type\_diagnostic\_variable\_id) **id\_npr**
- type(type\_diagnostic\_variable\_id) **id\_dpar**
- type(type\_diagnostic\_variable\_id) **id\_dmort**
- type(type\_diagnostic\_variable\_id) **id\_dllim**
- type(type\_diagnostic\_variable\_id) **id\_dnlim**
- type(type\_diagnostic\_variable\_id) **id\_dplim**
- type(type\_diagnostic\_variable\_id) **id\_dqnc**
- type(type\_diagnostic\_variable\_id) **id\_dqpc**
- type(type\_diagnostic\_variable\_id) **id\_den**
- type(type\_diagnostic\_variable\_id) **id\_dep**
- type(type\_diagnostic\_variable\_id) **id\_dec**
- type(type\_diagnostic\_variable\_id) **id\_dgraz**
- type(type\_diagnostic\_variable\_id) **id\_dmortz**
- type(type\_conserved\_quantity\_id) **id\_totn**
- type(type\_conserved\_quantity\_id) **id\_totp**
- real(rk) **p0**
- real(rk) **upmax\_n**
- real(rk) **upmax\_p**
- real(rk) **grow\_max**
- real(rk) **iv**
- real(rk) **halfsatn**
- real(rk) **halfsatp**
- real(rk) **rem\_n**
- real(rk) **rem\_p**
- real(rk) **mort0\_phy**
- real(rk) **mortpar\_phy**
- real(rk) **qmax\_n**
- real(rk) **qmax\_p**
- real(rk) **qmin\_n**
- real(rk) **qmin\_p**
- real(rk) **kc**
- real(rk) **w\_p**
- real(rk) **w\_d**
- real(rk) **rpn**
- real(rk) **grazmax**
- real(rk) **mort\_zoo**
- real(rk) **n**
- real(rk) **qzn**
- real(rk) **qzp**
- real(rk) **eff**
- real(rk) **e\_c**

- real(rk) **k\_detn**
- real(rk) **k\_detp**
- real(rk) **mort\_zoo2**
- real(rk) **n2**
- real(rk) **zexcdetfr**
- real(rk) **affin\_par**
- real(rk) **dic\_per\_n**
- logical **use\_dic**

### 6.2.1 Detailed Description

This is the derived model type.

#### Parameters

<i>affin_par</i>	$\alpha$ - initial slope of the P-I curve
<i>upmax_N</i>	$V_{max,N}$ -maximum nitrogen uptake rate

Definition at line 28 of file n2pzdq.F90.

The documentation for this type was generated from the following file:

- [n2pzdq.F90](#)

## Chapter 7

# File Documentation

### 7.1 n2pzdq.F90 File Reference

This file contains the [fabm\\_hzg\\_n2pzdq](#) module.

```
#include "fabm_driver.h"
```

#### Data Types

- module [fabm\\_hzg\\_n2pzdq](#)  
*This module describes an NPZD model extended with 2 nutrients and variable stoichiometry of phyto.*
- type [fabm\\_hzg\\_n2pzdq::type\\_hzg\\_n2pzdq](#)  
*This is the derived model type.*

#### 7.1.1 Detailed Description

This file contains the [fabm\\_hzg\\_n2pzdq](#) module.

Definition in file [n2pzdq.F90](#).

# Nomenclature

$\alpha$       affin\_par

$V_{max,N}$     upmax\_N

# Index

- do
  - fabm\_hzg\_n2pzdq, [8](#)
- fabm\_hzg\_n2pzdq, [7](#)
  - do, [8](#)
  - fprod, [8](#)
  - fupn, [8](#)
  - fupp, [8](#)
  - get\_light\_extinction, [9](#)
  - initialize, [9](#)
- fabm\_hzg\_n2pzdq::type\_hzg\_n2pzdq, [9](#)
- fprod
  - fabm\_hzg\_n2pzdq, [8](#)
- fupn
  - fabm\_hzg\_n2pzdq, [8](#)
- fupp
  - fabm\_hzg\_n2pzdq, [8](#)
- get\_light\_extinction
  - fabm\_hzg\_n2pzdq, [9](#)
- initialize
  - fabm\_hzg\_n2pzdq, [9](#)
- n2pzdq.F90, [12](#)