

# Package: rpmodel (via r-universe)

June 28, 2024

**Type** Package

**Title** P-Model

**Description** Implements the P-model (Stocker et al., 2020 <[doi:10.5194/gmd-13-1545-2020](https://doi.org/10.5194/gmd-13-1545-2020)>), predicting acclimated parameters of the enzyme kinetics of C3 photosynthesis, assimilation, and dark respiration rates as a function of the environment (temperature, CO2, vapour pressure deficit, light, atmospheric pressure).

**Version** 1.2.3

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Depends** R (>= 3.6)

**Suggests** ggplot2, dplyr, purrr, tidyr, knitr, rmarkdown, testthat, covr

**VignetteBuilder** knitr

**URL** <https://github.com/geco-bern/rpmodel>

**BugReports** <https://github.com/geco-bern/rpmodel/issues>

**NeedsCompilation** no

**Author** Benjamin Stocker [aut, cre]  
(<<https://orcid.org/0000-0003-2697-9096>>), Koen Hufkens [ctb]  
(<<https://orcid.org/0000-0002-5070-8109>>)

**Maintainer** Benjamin Stocker <[benjamin.stocker@gmail.com](mailto:benjamin.stocker@gmail.com)>

**Repository** CRAN

**Date/Publication** 2024-06-27 20:40:02 UTC

## Contents

calc_gammastar . . . . .	2
calc_kmm . . . . .	3

calc_patm . . . . .	4
calc_soilmstress . . . . .	5
co2_to_ca . . . . .	6
dampen_vec . . . . .	7
density_h2o . . . . .	8
ftemp_arrh . . . . .	8
ftemp_inst_jmax . . . . .	9
ftemp_inst_rd . . . . .	10
ftemp_inst_vcmax . . . . .	11
ftemp_kphio . . . . .	12
rpmode1 . . . . .	14
viscosity_h2o . . . . .	18

<b>Index</b>	<b>20</b>
--------------	-----------

---

calc_gammastar	<i>Calculates the CO2 compensation point</i>
----------------	--

---

### Description

Calculates the photorespiratory CO2 compensation point in absence of dark respiration,  $\Gamma^*$  (Farquhar, 1980).

### Usage

```
calc_gammastar(tc, patm)
```

### Arguments

tc	Temperature, relevant for photosynthesis (degrees Celsius)
patm	Atmospheric pressure (Pa)

### Details

The temperature and pressure-dependent photorespiratory compensation point in absence of dark respiration  $\Gamma^*(T, p)$  is calculated from its value at standard temperature ( $T0 = 25\text{deg C}$ ) and atmospheric pressure ( $p0 = 101325\text{ Pa}$ ), referred to as  $\Gamma^*0$ , quantified by Bernacchi et al. (2001) to 4.332 Pa (their value in molar concentration units is multiplied here with 101325 Pa to yield 4.332 Pa).  $\Gamma^*0$  is modified by temperature following an Arrhenius-type temperature response function  $f(T, \Delta Ha)$  (implemented by `ftemp_arrh`) with activation energy  $\Delta Ha = 37830\text{ J mol}^{-1}$  and is corrected for atmospheric pressure  $p(z)$  (see `calc_patm`) at elevation  $z$ .

$$\Gamma^* = \Gamma^*0 f(T, \Delta Ha) p(z) / p_0$$

$p(z)$  is given by argument `patm`.

### Value

A numeric value for  $\Gamma^*$  (in Pa)

## References

Farquhar, G. D., von Caemmerer, S., and Berry, J. A.: A biochemical model of photosynthetic CO<sub>2</sub> assimilation in leaves of C<sub>3</sub> species, *Planta*, 149, 78–90, 1980.

Bernacchi, C. J., Singaas, E. L., Pimentel, C., Portis, A. R. J., and Long, S. P.: Improved temperature response functions for models of Rubisco-limited photosynthesis, *Plant, Cell and Environment*, 24, 253–259, 2001

## Examples

```
print("CO2 compensation point at 20 degrees Celsius and standard atmosphere (in Pa):")
print(calc_gammastar(20, 101325))
```

---

calc_kmm	<i>Calculates the Michaelis Menten coefficient for Rubisco-limited photosynthesis</i>
----------	---

---

## Description

Calculates the Michaelis Menten coefficient of Rubisco-limited assimilation as a function of temperature and atmospheric pressure.

## Usage

```
calc_kmm(tc, patm)
```

## Arguments

tc	Temperature, relevant for photosynthesis (deg C)
patm	Atmospheric pressure (Pa)

## Details

The Michaelis-Menten coefficient  $K$  of Rubisco-limited photosynthesis is determined by the Michaelis-Menten constants for O<sub>2</sub> and CO<sub>2</sub> (Farquhar, 1980) according to:

$$K = Kc(1 + pO_2/Ko)$$

where  $Kc$  is the Michaelis-Menten constant for CO<sub>2</sub> (Pa),  $Ko$  is the Michaelis-Menten constant for O<sub>2</sub> (Pa), and  $pO_2$  is the partial pressure of oxygen (Pa), calculated as  $0.209476p$ , where  $p$  is given by argument `patm`.  $Kc$  and  $Ko$  follow a temperature dependence, given by the Arrhenius Equation  $f$  (implemented by [ftemp\\_arrh](#)):

$$Kc = Kc25f(T, \Delta Hkc)$$

$$Ko = Ko25f(T, \Delta Hko)$$

Values  $\Delta Hkc$  (79430 J mol<sup>-1</sup>),  $\Delta Hko$  (36380 J mol<sup>-1</sup>),  $Kc25$  (39.97 Pa), and  $Ko25$  (27480 Pa) are taken from Bernacchi et al. (2001) and have been converted from values given therein to units of Pa by multiplication with the standard atmosphere (101325 Pa).  $T$  is given by the argument `tc`.

**Value**

A numeric value for  $K$  (in Pa)

**References**

Farquhar, G. D., von Caemmerer, S., and Berry, J. A.: A biochemical model of photosynthetic CO<sub>2</sub> assimilation in leaves of C<sub>3</sub> species, *Planta*, 149, 78–90, 1980.

Bernacchi, C. J., Singaas, E. L., Pimentel, C., Portis, A. R. J., and Long, S. P.: Improved temperature response functions for models of Rubisco-limited photosynthesis, *Plant, Cell and Environment*, 24, 253–259, 2001

**Examples**

```
print("Michaelis-Menten coefficient at 20 degrees Celsius and standard atmosphere (in Pa):")
print(calc_kmm(20, 101325))
```

---

calc_patm	<i>Calculates atmospheric pressure</i>
-----------	--

---

**Description**

Calculates atmospheric pressure as a function of elevation, by default assuming standard atmosphere (101325 Pa at sea level)

**Usage**

```
calc_patm(elv, patm0 = 101325)
```

**Arguments**

elv	Elevation above sea-level (m.a.s.l.)
patm0	(Optional) Atmospheric pressure at sea level (Pa), defaults to 101325 Pa.

**Details**

The elevation-dependence of atmospheric pressure is computed by assuming a linear decrease in temperature with elevation and a mean adiabatic lapse rate (Berberan-Santos et al., 1997):

$$p(z) = p_0(1 - Lz/TK_0)^{gM/(RL)}$$

where  $z$  is the elevation above mean sea level (m, argument `elv`),  $g$  is the gravity constant (9.80665 m s<sup>-2</sup>),  $p_0$  is the atmospheric pressure at 0 m a.s.l. (argument `patm0`, defaults to 101325 Pa),  $L$  is the mean adiabatic lapse rate (0.0065 K m<sup>-2</sup>),  $M$  is the molecular weight for dry air (0.028963 kg mol<sup>-1</sup>),  $R$  is the universal gas constant (8.3145 J mol<sup>-1</sup> K<sup>-1</sup>), and  $TK_0$  is the standard temperature (298.15 K, corresponds to 25 deg C).

**Value**

A numeric value for  $p$

**References**

Allen, R. G., Pereira, L. S., Raes, D., Smith, M.: FAO Irrigation and Drainage Paper No. 56, Food and Agriculture Organization of the United Nations, 1998

**Examples**

```
print("Standard atmospheric pressure, in Pa, corrected for 1000 m.a.s.l.:")
print(calc_patm(1000))
```

---

calc_soilmstress	<i>Calculates an empirical soil moisture stress factor</i>
------------------	--

---

**Description**

Calculates an empirical soil moisture stress factor as a function of relative soil moisture (fraction of field capacity).

**Usage**

```
calc_soilmstress(soilm, meanalpha = 1, apar_soilm = 0, bpar_soilm = 0.685)
```

**Arguments**

soilm	Relative soil moisture as a fraction of field capacity (unitless). Defaults to 1.0 (no soil moisture stress).
meanalpha	Local annual mean ratio of actual over potential evapotranspiration, measure for average aridity. Defaults to 1.0.
apar_soilm	(Optional, used only if do_soilmstress==TRUE) Parameter determining the sensitivity of the empirical soil moisture stress function. Defaults to 0.0, the empirically fitted value as presented in Stocker et al. (2019) Geosci. Model Dev. for model setup 'FULL' (corresponding to a setup with method_jmaxlim="wang17", do_ftemp_kphio=TRUE, do_soilmstress=TRUE).
bpar_soilm	(Optional, used only if do_soilmstress==TRUE) Parameter determining the sensitivity of the empirical soil moisture stress function. Defaults to ~0.6, the empirically fitted value as presented in Stocker et al. (2019) Geosci. Model Dev. for model setup 'FULL' (corresponding to a setup with method_jmaxlim="wang17", do_ftemp_kphio=TRUE, do_soilmstress=TRUE).

**Details**

The soil moisture stress factor is calculated using a quadratic function that is 1 above `soilm = 0.6` and has a sensitivity, given by the y-axis cutoff, (zero soil moisture), determined by average aridity (argument `meanalpha`) as:

$$\beta = q(\theta - \theta^*)^2 + 1$$

for  $\theta < \theta^*$  and  $\beta = 1.0$  otherwise.  $\theta^*$  is fixed at 0.6.  $q$  is the sensitivity parameter and is calculated as a linear function of average aridity, quantified by the local annual mean ratio of actual over potential evapotranspiration, termed  $\alpha$ :

$$q = (\beta_0 - 1)/(\theta^* - \theta_0)^2$$

$\theta_0$  is 0.0, and

$$\beta_0 = a + b\alpha$$

$a$  is given by argument `apar`,  $b$  is given by argument `bpar`.

**Value**

A numeric value for  $\beta$

**References**

Stocker, B. et al. Geoscientific Model Development Discussions (in prep.)

**Examples**

```
## Relative reduction (%) in GPP due to soil moisture stress at
## relative soil water content ('soilm') of 0.2:
print((calc_soilmstress(0.2)-1)*100 )
```

---

co2\_to\_ca

*CO2 partial pressure*

---

**Description**

Calculates CO2 partial pressure from concentration in ppm.

**Usage**

```
co2_to_ca(co2, patm)
```

**Arguments**

<code>co2</code>	Atmospheric CO2 concentration (ppm)
<code>patm</code>	Atmospheric pressure (Pa).

**Value**

CO2 partial pressure in Pa.

---

dampen_vec	<i>Dampen inputs of rmodel</i>
------------	--------------------------------

---

**Description**

Applies an exponential dampening input time series with specified time scale.

**Usage**

```
dampen_vec(vec, tau)
```

**Arguments**

vec	A numeric vector for the time series of a daily meteorological variable used as input for rmodel (temperature, vapour pressure deficit, CO2, or atmospheric pressure). The length of x must be at least 365, i.e., corresponding to one year.
tau	The time scale of dampening (e-folding time scale of a perturbation). Must be smaller or equal to 365 d.

**Value**

A numeric vector of equal length as x with damped variation. The dampening is calculated as:

$$S(t + 1) - S(t) = (X(t + 1) - S(t))/\tau$$

Where  $X$  is the daily varying time series given by argument x,  $S$  is the dampened time returned by this function, and  $\tau$  is the decay time scale of a perturbation, given by argument tau.

**Examples**

```
## Not run:  
dampen_vec(  
  vec = 20 * (sin(doy*pi/(365)))^2 + rnorm(365, mean = 0, sd = 5),  
  tau = 40  
)  
  
## End(Not run)
```

---

density_h2o	<i>Density of water</i>
-------------	-------------------------

---

**Description**

Calculates the density of water as a function of temperature and atmospheric pressure, using the Tumlirz Equation.

**Usage**

```
density_h2o(tc, p)
```

**Arguments**

tc	numeric, air temperature (tc), degrees C
p	numeric, atmospheric pressure (p), Pa

**Value**

numeric, density of water, kg/m<sup>3</sup>

**References**

F.H. Fisher and O.E Dial, Jr. (1975) Equation of state of pure water and sea water, Tech. Rept., Marine Physical Laboratory, San Diego, CA.

**Examples**

```
# Density of water at 20 degrees C and standard atmospheric pressure  
print(density_h2o(20, 101325))
```

---

ftemp_arrh	<i>Calculates the Arrhenius-type temperature response</i>
------------	---

---

**Description**

Given a kinetic rate at a reference temperature (argument tkref) this function calculates its temperature-scaling factor following Arrhenius kinetics.

**Usage**

```
ftemp_arrh(tk, dha, tkref = 298.15)
```



**Arguments**

tk	Temperature (Kelvin)
dha	Activation energy (J mol <sup>-1</sup> )
tkref	Reference temperature (Kelvin)

**Details**

To correct for effects by temperature following Arrhenius kinetics, and given a reference temperature  $T_0$ ,  $f$  calculates the temperature scaling. Arrhenius kinetics are described by an equation of form  $x(T) = \exp(c - \Delta H_a / (TR))$ . The temperature-correction function  $f(T, \Delta H_a)$  is thus given by  $f = x(T)/x(T_0)$  which is:

$$f = \exp(\Delta H_a(T - T_0)/(T_0RT_K))$$

$\Delta H_a$  is given by argument dha.  $T$  is given by argument tk and has to be provided in Kelvin.  $R$  is the universal gas constant and is 8.3145 J mol<sup>-1</sup> K<sup>-1</sup>. Note that this is equivalent to

$$f = \exp((\Delta H_a/R)(1/T_0 - 1/T))$$

**Value**

A numeric value for  $f$

**Examples**

```
# Relative rate change from 25 to 10 degrees Celsius (percent change)
print( (1.0-ftemp_arrh( 283.15, 100000, tkref = 298.15))*100 )
```

---

ftemp_inst_jmax	<i>Calculates the instantaneous temperature response of Jmax</i>
-----------------	--

---

**Description**

Given Jmax at a reference temperature (argument tceref) this function calculates its temperature-scaling factor following modified Arrhenius kinetics based on Kattge & Knorr (2007). Calculates  $f$  for the conversion

$$V = fV_{ref}$$

**Usage**

```
ftemp_inst_jmax(tcleaf, tcgrowth = tcleaf, tceref = 25)
```

**Arguments**

tcleaf	Leaf temperature, or in general the temperature relevant for photosynthesis (degrees Celsius)
tcgrowth	(Optional) Growth temperature, in the P-model, taken to be equal to tcleaf (in degrees Celsius). Defaults to tcgrowth = tcleaf.
tceref	Reference temperature (in degrees Celsius)

**Details**

The function is given by Kattge & Knorr (2007) as

$$fv = f(T, \Delta Hv)A/B$$

where  $f(T, \Delta Hv)$  is a regular Arrhenius-type temperature response function (see [ftemp\\_arrh](#)) with  $Hv = 49884 \text{ J mol}^{-1}$ ,

$$A = 1 + \exp((T_0\Delta S - Hd)/(T_0R))$$

and

$$B = 1 + \exp((T\Delta S - Hd)/(TKR))$$

Here,  $T$  is in Kelvin,  $T_0 = 293.15 \text{ K}$ ,  $Hd = 200000 \text{ J mol}^{-1}$  is the deactivation energy and  $R$  is the universal gas constant and is  $8.3145 \text{ J mol}^{-1} \text{ K}^{-1}$ , and

$$\Delta S = aS - bST$$

with  $aS = 659.70 \text{ J mol}^{-1} \text{ K}^{-1}$ , and  $bS = 0.75 \text{ J mol}^{-1} \text{ K}^{-2}$ , and  $T$  given in degrees Celsius (!)

**Value**

A numeric value for  $fv$

**References**

Kattge, J. and Knorr, W.: Temperature acclimation in a biochemical model of photosynthesis: a reanalysis of data from 36 species, *Plant, Cell and Environment*, 30,1176–1190, 2007.

**Examples**

```
# Relative change in Jmax going (instantaneously, i.e.
# not acclimatedly) from 10 to 25 degrees (percent change):
print((ftemp_inst_jmax(25)/ftemp_inst_jmax(10)-1)*100 )
```

---

ftemp\_inst\_rd

*Calculates the temperature response of dark respiration*

---

**Description**

Given the dark respiration at the reference temperature 25 degree Celsius, this function calculates its temperature-scaling factor following Heskell et al. 2016.

**Usage**

```
ftemp_inst_rd(tc)
```

**Arguments**

tc                      Temperature (degrees Celsius)

**Details**

To correct for effects by temperature Heskell et al. 2016, and given the reference temperature  $T_0 = 25$  deg C, this calculates the temperature scaling factor to calculate dark respiration at temperature  $T$  (argument tc) as:

$$fr = \exp(0.1012(T_0 - T) - 0.0005(T_0^2 - T^2))$$

where  $T$  is given in degrees Celsius.

**Value**

A numeric value for  $fr$

**References**

Heskell, M., O'Sullivan, O., Reich, P., Tjoelker, M., Weerasinghe, L., Penillard, A., Egerton, J., Creek, D., Bloomfield, K., Xiang, J., Sinca, F., Stangl, Z., Martinez-De La Torre, A., Griffin, K., Huntingford, C., Hurry, V., Meir, P., Turnbull, M., and Atkin, O.: Convergence in the temperature response of leaf respiration across biomes and plant functional types, *Proceedings of the National Academy of Sciences*, 113, 3832–3837, doi:10.1073/pnas.1520282113, 2016.

**Examples**

```
## Relative change in Rd going (instantaneously, i.e. not
## acclimatedly) from 10 to 25 degrees (percent change):
print( (ftemp_inst_rd(25)/ftemp_inst_rd(10)-1)*100 )
```

---

<code>ftemp_inst_vcmax</code>	<i>Calculates the instantaneous temperature response of Vcmax</i>
-------------------------------	---

---

**Description**

Given Vcmax at a reference temperature (argument tc<sub>ref</sub>) this function calculates its temperature-scaling factor following modified Arrhenius kinetics based on Kattge & Knorr (2007). Calculates  $f$  for the conversion

$$V = fV_{ref}$$

**Usage**

```
ftemp_inst_vcmax(tcleaf, tcgrowth = tcleaf, tcref = 25)
```

**Arguments**

<code>tc<sub>leaf</sub></code>	Leaf temperature, or in general the temperature relevant for photosynthesis (degrees Celsius)
<code>tc<sub>growth</sub></code>	(Optional) Growth temperature, in the P-model, taken to be equal to <code>tc<sub>leaf</sub></code> (in degrees Celsius). Defaults to <code>tc<sub>growth</sub> = tc<sub>leaf</sub></code> .
<code>tc<sub>ref</sub></code>	Reference temperature (in degrees Celsius)

**Details**

The function is given by Kattge & Knorr (2007) as

$$fv = f(T, \Delta Hv)A/B$$

where  $f(T, \Delta Hv)$  is a regular Arrhenius-type temperature response function (see `ftemp_arrh`) with  $Hv = 71513 \text{ J mol}^{-1}$ ,

$$A = 1 + \exp((T0\Delta S - Hd)/(T0R))$$

and

$$B = 1 + \exp((T\Delta S - Hd)/(TKR))$$

Here,  $T$  is in Kelvin,  $T0 = 293.15 \text{ K}$ ,  $Hd = 200000 \text{ J mol}^{-1}$  is the deactivation energy and  $R$  is the universal gas constant and is  $8.3145 \text{ J mol}^{-1} \text{ K}^{-1}$ , and

$$\Delta S = aS - bST$$

with  $aS = 668.39 \text{ J mol}^{-1} \text{ K}^{-1}$ , and  $bS = 1.07 \text{ J mol}^{-1} \text{ K}^{-2}$ , and  $T$  given in degrees Celsius (!)

**Value**

A numeric value for  $fv$

**References**

Kattge, J. and Knorr, W.: Temperature acclimation in a biochemical model of photosynthesis: a reanalysis of data from 36 species, *Plant, Cell and Environment*, 30,1176–1190, 2007.

**Examples**

```
## Relative change in Vcmax going (instantaneously, i.e.
## not acclimatedly) from 10 to 25 degrees (percent change):
print((ftemp_inst_vcmax(25)/ftemp_inst_vcmax(10)-1)*100 )
```

---

ftemp\_kphio

*Calculates the temperature dependence of the quantum yield efficiency*

---

**Description**

Calculates the temperature dependence of the quantum yield efficiency following the temperature dependence of the maximum quantum yield of photosystem II in light-adapted tobacco leaves, determined by Bernacchi et al. (2003)

**Usage**

```
ftemp_kphio(tc, c4 = FALSE)
```

**Arguments**

tc	Temperature, relevant for photosynthesis (degrees Celsius)
c4	Boolean specifying whether fitted temperature response for C4 plants is used. Defaults to FALSE (C3 photosynthesis temperature response following Bernacchi et al., 2003 is used).

**Details**

The temperature factor for C3 photosynthesis (argument c4 = FALSE) is calculated based on Bernacchi et al. (2003) as

$$\phi(T) = 0.352 + 0.022T - 0.00034T^2$$

The temperature factor for C4 (argument c4 = TRUE) photosynthesis is calculated based on pers. comm. by David Orme, correcting values provided in Cai & Prentice (2020). Corrected parametrisation is:

$$\phi(T) = -0.064 + 0.03T - 0.000464T^2$$

The factor  $\phi(T)$  is to be multiplied with leaf absorptance and the fraction of absorbed light that reaches photosystem II. In the P-model these additional factors are lumped into a single apparent quantum yield efficiency parameter (argument kphio to function [rpmmodel](#)).

**Value**

A numeric value for  $\phi(T)$

**References**

Bernacchi, C. J., Pimentel, C., and Long, S. P.: In vivo temperature response functions of parameters required to model RuBP-limited photosynthesis, *Plant Cell Environ.*, 26, 1419–1430, 2003  
 Cai, W., and Prentice, I. C.: Recent trends in gross primary production and their drivers: analysis and modelling at flux-site and global scales, *Environ. Res. Lett.* 15 124050 <https://doi.org/10.1088/1748-9326/abc64e>, 2020

**Examples**

```
## Relative change in the quantum yield efficiency
## between 5 and 25 degrees celsius (percent change):
print(paste((ftemp_kphio(25.0)/ftemp_kphio(5.0)-1)*100 ))
```

---

 rmodel

*Invokes a P-model function call*


---

### Description

R implementation of the P-model and its corollary predictions (Prentice et al., 2014; Han et al., 2017).

### Usage

```
rmodel(
  tc,
  vpd,
  co2,
  fapar,
  ppfd,
  patm = NA,
  elv = NA,
  kphio = ifelse(c4, 1, ifelse(do_ftemp_kphio, ifelse(do_soilmstress, 0.087182,
    0.081785), 0.049977)),
  beta = ifelse(c4, 146/9, 146),
  soilm = stopifnot(!do_soilmstress),
  meanalpha = 1,
  apar_soilm = 0,
  bpar_soilm = 0.733,
  c4 = FALSE,
  method_jmaxlim = "wang17",
  do_ftemp_kphio = TRUE,
  do_soilmstress = FALSE,
  returnvar = NULL,
  verbose = FALSE
)
```

### Arguments

tc	Temperature, relevant for photosynthesis (deg C)
vpd	Vapour pressure deficit (Pa)
co2	Atmospheric CO2 concentration (ppm)
fapar	(Optional) Fraction of absorbed photosynthetically active radiation (unitless, defaults to NA)
ppfd	Incident photosynthetic photon flux density (mol m <sup>-2</sup> d <sup>-1</sup> , defaults to NA). Note that the units of ppfd (per area and per time) determine the units of outputs lue, gpp, vcmax, and rd. For example, if ppfd is provided in units of mol m <sup>-2</sup> month <sup>-1</sup> , then respective output variables are returned as per unit months.

patm	Atmospheric pressure (Pa). When provided, overrides elv, otherwise patm is calculated using standard atmosphere (101325 Pa), corrected for elevation (argument elv), using the function <code>calc_patm</code> .
elv	Elevation above sea-level (m.a.s.l.). Is used only for calculating atmospheric pressure (using standard atmosphere (101325 Pa), corrected for elevation (argument elv), using the function <code>calc_patm</code> ), if argument patm is not provided. If argument patm is provided, elv is overridden.
kphio	Apparent quantum yield efficiency (unitless). Defaults to 0.081785 for <code>method_jmaxlim="wang17"</code> , <code>do_ftemp_kphio=TRUE</code> , <code>do_soilmstress=FALSE</code> , 0.087182 for <code>method_jmaxlim="wang17"</code> , <code>do_ftemp_kphio=TRUE</code> , <code>do_soilmstress=TRUE</code> , and 0.049977 for <code>method_jmaxlim="wang17"</code> , <code>do_ftemp_kphio=FALSE</code> , <code>do_soilmstress=FALSE</code> , corresponding to the empirically fitted value as presented in Stocker et al. (2019) Geosci. Model Dev. for model setup 'BRC', 'FULL', and 'ORG' respectively, corresponding to $(a_L b_L)/4$ in Eq.20 in Stocker et al. (2020) for C3 photosynthesis. For C4 photosynthesis ( <code>c4 = TRUE</code> ), kphio defaults to 1.0, corresponding to the parametrisation by Cai & Prentice (2020).
beta	Unit cost ratio. Defaults to 146.0 (see Stocker et al., 2019) for C3 plants and 146/9 for C4 plants.
soilm	(Optional, used only if <code>do_soilmstress==TRUE</code> ) Relative soil moisture as a fraction of field capacity (unitless). Defaults to 1.0 (no soil moisture stress). This information is used to calculate an empirical soil moisture stress factor ( <code>calc_soilmstress</code> ) whereby the sensitivity is determined by average aridity, defined by the local annual mean ratio of actual over potential evapotranspiration, supplied by argument <code>meanalpha</code> .
meanalpha	(Optional, used only if <code>do_soilmstress==TRUE</code> ) Local annual mean ratio of actual over potential evapotranspiration, measure for average aridity. Defaults to 1.0. Only scalar numbers are accepted. If a vector is provided, only the first element will be used.
apar_soilm	(Optional, used only if <code>do_soilmstress==TRUE</code> ) Parameter determining the sensitivity of the empirical soil moisture stress function. Defaults to 0.0, the empirically fitted value as presented in Stocker et al. (2019) Geosci. Model Dev. for model setup 'FULL' (corresponding to a setup with <code>method_jmaxlim="wang17"</code> , <code>do_ftemp_kphio=TRUE</code> , <code>do_soilmstress=TRUE</code> ).
bpar_soilm	(Optional, used only if <code>do_soilmstress==TRUE</code> ) Parameter determining the sensitivity of the empirical soil moisture stress function. Defaults to 0.7330, the empirically fitted value as presented in Stocker et al. (2019) Geosci. Model Dev. for model setup 'FULL' (corresponding to a setup with <code>method_jmaxlim="wang17"</code> , <code>do_ftemp_kphio=TRUE</code> , <code>do_soilmstress=TRUE</code> ).
c4	(Optional) A logical value specifying whether the C3 or C4 photosynthetic pathway is followed. Defaults to FALSE. If TRUE, the leaf-internal CO <sub>2</sub> concentration is still estimated using beta but $m$ (returned variable <code>mj</code> ) tends to 1, and $m'$ tends to 0.669 (with $c = 0.41$ ) to represent CO <sub>2</sub> concentrations within the leaf. With <code>do_ftemp_kphio = TRUE</code> , a C4-specific temperature dependence of the quantum yield efficiency is used (see <code>ftemp_kphio</code> ).
method_jmaxlim	(Optional) A character string specifying which method is to be used for factoring in Jmax limitation. Defaults to "wang17", based on Wang Han et al. 2017 Nature Plants and (Smith 1937). Available is also "smith19", following the method by

	Smith et al., 2019 Ecology Letters, and "none" for ignoring effects of Jmax limitation.
do_ftemp_kphio	(Optional) A logical specifying whether temperature-dependence of quantum yield efficiency is used. See <a href="#">ftemp_kphio</a> for details. Defaults to TRUE. Only scalar numbers are accepted. If a vector is provided, only the first element will be used.
do_soilmstress	(Optional) A logical specifying whether an empirical soil moisture stress factor is to be applied to down-scale light use efficiency (and only light use efficiency). Defaults to FALSE.
returnvar	(Optional) A character string of vector of character strings specifying which variables are to be returned (see return below).
verbose	Logical, defines whether verbose messages are printed. Defaults to FALSE.

### Value

A named list of numeric values (including temperature and pressure dependent parameters of the photosynthesis model, P-model predictions, including all its corollary). This includes :

- ca: Ambient CO2 expressed as partial pressure (Pa)
- gammastar: Photorespiratory compensation point  $\Gamma^*$ , (Pa), see [calc\\_gammastar](#).
- kmm: Michaelis-Menten coefficient  $K$  for photosynthesis (Pa), see [calc\\_kmm](#).
- ns\_star: Change in the viscosity of water, relative to its value at 25 deg C (unitless).

$$\eta^* = \eta(T)/\eta(25degC)$$

This is used to scale the unit cost of transpiration. Calculated following Huber et al. (2009).

- chi: Optimal ratio of leaf internal to ambient CO2 (unitless). Derived following Prentice et al.(2014) as:

$$\chi = \Gamma^* / ca + (1 - \Gamma^* / ca)\xi / (\xi + \sqrt{D})$$

with

$$\xi = \sqrt{(\beta(K + \Gamma^*) / (1.6\eta^*))}$$

$\beta$  is given by argument beta,  $K$  is kmm (see [calc\\_kmm](#)),  $\Gamma^*$  is gammastar (see [calc\\_gammastar](#)).  $\eta^*$  is ns\_star.  $D$  is the vapour pressure deficit (argument vpd),  $ca$  is the ambient CO2 partial pressure in Pa (ca).

- ci: Leaf-internal CO2 partial pressure (Pa), calculated as  $(\chi ca)$ .
- lue: Light use efficiency (g C / mol photons), calculated as

$$LUE = \phi(T)\phi_0 m' Mc$$

where  $\phi(T)$  is the temperature-dependent quantum yield efficiency modifier ([ftemp\\_kphio](#)) if do\_ftemp\_kphio==TRUE, and 1 otherwise.  $\phi_0$  is given by argument kphio.  $m' = m$  if method\_jmaxlim=="none", otherwise

$$m' = m\sqrt{1 - (c/m)^{(2/3)}}$$

with  $c = 0.41$  (Wang et al., 2017) if method\_jmaxlim=="wang17".  $Mc$  is the molecular mass of C (12.0107 g mol<sup>-1</sup>).  $m$  is given returned variable mj. If do\_soilmstress==TRUE,  $LUE$  is multiplied with a soil moisture stress factor, calculated with [calc\\_soilmstress](#).



- mj: Factor in the light-limited assimilation rate function, given by

$$m = (c_i - \Gamma^*) / (c_i + 2\Gamma^*)$$

where  $\Gamma^*$  is given by `calc_gammastar`.

- mc: Factor in the Rubisco-limited assimilation rate function, given by

$$mc = (c_i - \Gamma^*) / (c_i + K)$$

where  $K$  is given by `calc_kmm`.

- gpp: Gross primary production (g C m<sup>-2</sup>), calculated as

$$GPP = IabsLUE$$

where  $Iabs$  is given by `fapar*ppfd` (arguments), and is NA if `fapar==NA` or `ppfd==NA`. Note that `gpp` scales with absorbed light. Thus, its units depend on the units in which `ppfd` is given.

- iwue: Intrinsic water use efficiency (iWUE, Pa), calculated as

$$iWUE = ca(1 - \chi) / (1.6)$$

- gs: Stomatal conductance (gs, in mol C m<sup>-2</sup> Pa<sup>-1</sup>), calculated as

$$gs = A / (ca(1 - \chi))$$

where  $A$  is `gpp/Mc`.

- vcmax: Maximum carboxylation capacity  $Vcmax$  (mol C m<sup>-2</sup>) at growth temperature (argument `tc`), calculated as

$$Vcmax = \phi(T)\phi_0Iabsn$$

where  $n$  is given by  $n = m' / mc$ .

- vcmax25: Maximum carboxylation capacity  $Vcmax$  (mol C m<sup>-2</sup>) normalised to 25 deg C following a modified Arrhenius equation, calculated as  $Vcmax25 = Vcmax / fv$ , where  $fv$  is the instantaneous temperature response by  $Vcmax$  and is implemented by function `ftemp_inst_vcmax`.

- jmax: The maximum rate of RuBP regeneration () at growth temperature (argument `tc`), calculated using

$$A_J = A_C$$

- rd: Dark respiration  $Rd$  (mol C m<sup>-2</sup>), calculated as

$$Rd = b_0Vcmax(fr/fv)$$

where  $b_0$  is a constant and set to 0.015 (Atkin et al., 2015),  $fv$  is the instantaneous temperature response by  $Vcmax$  and is implemented by function `ftemp_inst_vcmax`, and  $fr$  is the instantaneous temperature response of dark respiration following Heskell et al. (2016) and is implemented by function `ftemp_inst_rd`.

Additional variables are contained in the returned list if argument `method_jmaxlim=="smith19"`

- omega: Term corresponding to  $\omega$ , defined by Eq. 16 in Smith et al. (2019), and Eq. E19 in Stocker et al. (2019).
- omega\_star: Term corresponding to  $\omega^*$ , defined by Eq. 18 in Smith et al. (2019), and Eq. E21 in Stocker et al. (2019).

patm

## References

- Bernacchi, C. J., Pimentel, C., and Long, S. P.: In vivo temperature response functions of parameters required to model RuBP-limited photosynthesis, *Plant Cell Environ.*, 26, 1419–1430, 2003
- and their drivers: analysis and modelling at flux-site and global scales, *Environ. Res. Lett.* 15 124050 <https://doi.org/10.1088/1748-9326/abc64e>, 2020
- Heskel, M., O'Sullivan, O., Reich, P., Tjoelker, M., Weerasinghe, L., Penillard, A., Egerton, J., Creek, D., Bloomfield, K., Xiang, J., Sinca, F., Stangl, Z., Martinez-De La Torre, A., Griffin, K., Huntingford, C., Hurry, V., Meir, P., Turnbull, M., and Atkin, O.: Convergence in the temperature response of leaf respiration across biomes and plant functional types, *Proceedings of the National Academy of Sciences*, 113, 3832–3837, doi:10.1073/pnas.1520282113, 2016.
- Huber, M. L., Perkins, R. A., Laesecke, A., Friend, D. G., Sengers, J. V., Assael, M. J., Metaxa, I. N., Vogel, E., Mares, R., and Miyagawa, K.: New international formulation for the viscosity of H<sub>2</sub>O, *Journal of Physical and Chemical Reference Data*, 38, 101–125, 2009
- Prentice, I. C., Dong, N., Gleason, S. M., Maire, V., and Wright, I. J.: Balancing the costs of carbon gain and water transport: testing a new theoretical framework for plant functional ecology, *Ecology Letters*, 17, 82–91, 10.1111/ele.12211, <http://dx.doi.org/10.1111/ele.12211>, 2014.
- Wang, H., Prentice, I. C., Keenan, T. F., Davis, T. W., Wright, I. J., Cornwell, W. K., Evans, B. J., and Peng, C.: Towards a universal model for carbon dioxide uptake by plants, *Nat Plants*, 3, 734–741, 2017. Atkin, O. K., et al.: Global variability in leaf respiration in relation to climate, plant functional types and leaf traits, *New Phytologist*, 206, 614–636, doi:10.1111/nph.13253, <https://nph.onlinelibrary.wiley.com/doi/abs/10.1111/nph.13253>.
- Smith, N. G., Keenan, T. F., Colin Prentice, I., Wang, H., Wright, I. J., Niinemets, U., Crous, K. Y., Domingues, T. F., Guerrieri, R., Yoko Ishida, F., Kattge, J., Kruger, E. L., Maire, V., Rogers, A., Serbin, S. P., Tarvainen, L., Togashi, H. F., Townsend, P. A., Wang, M., Weerasinghe, L. K. and Zhou, S. (2019), Global photosynthetic capacity is optimized to the environment. *Ecol Lett*, 22: 506-517. doi:10.1111/ele.13210
- Stocker, B. et al. Geoscientific Model Development Discussions (in prep.)

## Examples

```
## Not run:
  rpmodel(
    tc = 20,
    vpd = 1000,
    co2 = 400,
    ppfd = 30,
    elv = 0)

## End(Not run)
```

**Description**

Calculates the viscosity of water as a function of temperature and atmospheric pressure.

**Usage**

```
viscosity_h2o(tc, p)
```

**Arguments**

tc	numeric, air temperature (tc), degrees C
p	numeric, atmospheric pressure (p), Pa

**Value**

numeric, viscosity of water ( $\mu$ ), Pa s

**References**

Huber, M. L., R. A. Perkins, A. Laesecke, D. G. Friend, J. V. Sengers, M. J. Assael, ..., K. Miyagawa (2009) New international formulation for the viscosity of H<sub>2</sub>O, J. Phys. Chem. Ref. Data, Vol. 38(2), pp. 101-125.

**Examples**

```
print("Density of water at 20 degrees C and standard atmospheric pressure:")  
print(density_h2o(20, 101325))
```

# Index

calc\_gammastar, [2](#), [16](#)  
calc\_kmm, [3](#), [16](#)  
calc\_patm, [2](#), [4](#), [15](#)  
calc\_soilmstress, [5](#), [15](#), [16](#)  
co2\_to\_ca, [6](#)

dampen\_vec, [7](#)  
density\_h2o, [8](#)

ftemp\_arrh, [2](#), [3](#), [8](#), [10](#), [12](#)  
ftemp\_inst\_jmax, [9](#)  
ftemp\_inst\_rd, [10](#), [17](#)  
ftemp\_inst\_vcmax, [11](#), [17](#)  
ftemp\_kphio, [12](#), [15](#), [16](#)

rpmode1, [13](#), [14](#)

viscosity\_h2o, [18](#)