

# The PRIVATE functions of package AquaEnv

Andreas F. Hofmann

April 5, 2011

---

**Abi**

*Abi*

---

## Description

PRIVATE function: calculates  $[A(2-)]$  of a bivalent acid

## Usage

`Abi(Sum, K1, K2, H)`

## Arguments

Sum	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)

## Author(s)

Andreas F. Hofmann ([a.hofmann@nioo.knaw.nl](mailto:a.hofmann@nioo.knaw.nl))

---

Atri

*Atri*

---

### Description

PRIVATE function: calculates  $[A(3-)]$  of a trivalent acid

### Usage

Atri(Sum, K1, K2, K3, H)

### Arguments

Sum	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K3	the third dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

Auni

*Auni*

---

### Description

PRIVATE function: calculates  $[A(-)]$  of an univalent acid

### Usage

Auni(Sum, K, H)

### Arguments

Sum	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K	the dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

Cl

*Cl*

---

**Description**

PRIVATE function: calculates chlorinity Cl from salinity S

**Usage**

Cl(S)

**Arguments**

S                      salinity S in practical salinity units (i.e. no unit)

**Value**

chlorinity Cl in permil

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

**References**

DOE1994, Zeebe2001

---

ConcRelCl

*ConcRelCl*

---

**Description**

PUBLIC data frame: a collection of concentrations of key chemical species in seawater, relative with respect to chlorinity (DOE1994))

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

H2Abi

*H2Abi*

---

### Description

PRIVATE function: calculates [H2A] of a bivalent acid

### Usage

H2Abi(Sum, K1, K2, H)

### Arguments

Sum	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

H2Atri

*H2Atri*

---

### Description

PRIVATE function: calculates [H2A(-)] of a trivalent acid

### Usage

H2Atri(Sum, K1, K2, K3, H)

### Arguments

Sum	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)

K3	the third dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)

#### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

H3Atri	<i>H3Atri</i>
--------	---------------

---

#### Description

PRIVATE function: calculates [H3A] of a trivalent acid

#### Usage

H3Atri(Sum, K1, K2, K3, H)

#### Arguments

Sum	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K3	the third dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)

#### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

HAbi	<i>HAbi</i>
------	-------------

---

### Description

PRIVATE function: calculates [HA(-)] of a bivalent acid

### Usage

HAbi(Sum, K1, K2, H)

### Arguments

Sum	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

HAtri	<i>HAtri</i>
-------	--------------

---

### Description

PRIVATE function: calculates [HA(2-)] of a trivalent acid

### Usage

HAtri(Sum, K1, K2, K3, H)

### Arguments

Sum	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)

K3	the third dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
----	--

H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)
---	---

#### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

HAuni	<i>HAuni</i>
-------	--------------

---

#### Description

PRIVATE function: calculates [HA] of an univalent acid

#### Usage

HAuni(Sum, K, H)

#### Arguments

Sum	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
-----	--

K	the dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
---	--

H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)
---	---

#### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

I	<i>I</i>
---	----------

---

#### Description

PRIVATE function: calculates the ionic strength I as a function of salinity S

#### Usage

I(S)

#### Arguments

S	salinity S in practical salinity units (i.e. no unit)
---	---

**Value**

ionic strength in mol/kg-H<sub>2</sub>O (molality)

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

**References**

DOE1994, Zeebe2001, Roy1993b (the carbonic acid paper)

---

<code>Iterms</code>	<i>Iterms</i>
---------------------	---------------

---

**Description**

PRIVATE function: returns the ionic strenght  $I$ ,  $I(2)$ ,  $\text{sqrt}(I)$ , and  $I*\text{sqrt}(I)$

**Usage**

```
Iterms(S)
```

**Arguments**

<code>S</code>	salinity in practical salinity units (i.e. no unit)
----------------	---

**Value**

a list containing:

<code>I</code>	the ionic strength
<code>I^2</code>	the square of the ionic strength
<code>sqrtI</code>	the square root of the ionic strength
<code>I*sqrtI</code>	the ionic strength times its square root

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)



---

<b>Sterms</b>	<i>Sterms</i>
---------------	---------------

---

### Description

PRIVATE function: returns S, S(2), sqrt(S), and S\*sqrt(S)

### Usage

**Sterms**(S)

### Arguments

**S**                      salinity in practical salinity units (i.e. no unit)

### Value

a list containing:

<b>S^2</b>	the square of S
<b>sqrtS</b>	the square root of S
<b>S*sqrtS</b>	S times its square root

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

<b>T</b>	<i>T</i>
----------	----------

---

### Description

PRIVATE function: calculates the temperature in Kelvin from the temperature in degrees centigrade

### Usage

**T**(t)

### Arguments

**t**                      temperature in degrees centigrade

### Value

temperature in Kelvin

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

**att**

*att*

---

**Description**

PRIVATE function: sets the attributes for calculated dissociation constants (Ks)

**Usage**

`att(K)`

**Arguments**

**K**                      the calculated dissociation constant K

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

**basicplot**

*basicplot*

---

**Description**

PRIVATE function: basic wrapper for the R plot function for plotting objects of class aquaenv; no return value, just side-effect

**Usage**

```
basicplot(aquaenv, xval, type="l", mgp=c(1.8, 0.5, 0),
          mar=c(3,3,0.5,0.5), oma=c(0,0,0,0), size=c(15,13),
          mfrow=c(11,10), device="x11", filename="aquaenv",
          newdevice, setpar,...)
```

**Arguments**

**aquaenv**              object of class aquaenv

**xval**                  x-value: the independent variable describing a change in elements of an object of class aquaenv

**type**                  standard plot parameter; default: plot lines

**mgp**                   standard plot parameter; default: axis title on line 1.8, axis labels on line 0.5, axis on line 0

<b>mar</b>	standard plot parameter; default: margin of 3 lines bottom and left and 0.5 lines top and right
<b>oma</b>	standard plot parameter; default: no outer margin
<b>size</b>	the size of the plot device; default: 15 (width) by 13 (height) inches
<b>mfrow</b>	standard plot parameter; default: 11 columns and 10 rows of plots
<b>device</b>	the device to plot on; default: "x11" (can also be "eps" or "pdf")
<b>filename</b>	filename to be used if "eps" or "pdf" is selected for device
<b>newdevice</b>	flag: if TRUE, new plot device is opened
<b>setpar</b>	flag: if TRUE parameters are set with the function par
<b>...</b>	further arguments will be passed

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

<b>bjerrumplot</b>	<i>bjerrumplot</i>
--------------------	--------------------

---

### Description

PRIVATE function: creates a bjerrumplot from the elements of an object of class aquaenv given in what; no return value, just side-effect

### Usage

```
bjerrumplot(aquaenv, what, log=FALSE, palette=NULL, device="x11",
            filename="aquaenv", size=c(12,10), ylim=NULL,
            lwd=2, xlab="free scale pH",
            mgp=c(1.8, 0.5, 0), mar=c(3,3,0.5,0.5), oma=c(0,0,0,0),
            legendposition="bottomleft", legendinset=0.05, legendlwd=4,
            bg="white", newdevice, setpar,...)
```

### Arguments

<b>aquaenv</b>	object of class aquaenv
<b>what</b>	vector of names of elements of aquaenv that should be plotted; if not specified: what <- c("CO2", "HCO3", "CO3", "BOH3", "BOH4", "OH", "H3PO4", "H2PO4", "HPO4", "PO4", "SiOH4", "SiOOH3", "SiO2OH2", "H2S", "HS", "S2min", "NH4", "NH3", "H2SO4", "HSO4", "SO4", "HF", "F", "HNO3", "NO3", "HNO2", "NO2")
<b>log</b>	should the plot be on a logarithmic y axis?
<b>palette</b>	a vector of colors to use in the plot (either numbers or names given in colors())
<b>device</b>	the device to plot on; default: "x11" (can also be "eps" or "pdf")

<b>filename</b>	filename to be used if "eps" or "pdf" is selected for device
<b>size</b>	the size of the plot device; default: 12 (width) by 10 (height) inches
<b>ylim</b>	standard plot parameter; if not supplied it will be calculated by range() of the elements to plot
<b>lwd</b>	standard plot parameter; width of the lines in the plot
<b>xlab</b>	x axis label
<b>mgp</b>	standard plot parameter; default: axis title on line 1.8, axis labels on line 0.5, axis on line 0
<b>mar</b>	standard plot parameter; default: margin of 3 lines bottom and left and 0.5 lines top and right
<b>oma</b>	standard plot parameter; default: no outer margin
<b>legendposition</b>	position of the legend
<b>legendinset</b>	standard legend parameter inset
<b>legendlwd</b>	standard legend parameter lwd: line width of lines in legend
<b>bg</b>	standard legend parameter: default background color: white
<b>newdevice</b>	flag: if TRUE, new plot device is opened
<b>setpar</b>	flag: if TRUE parameters are set with the function par
<b>...</b>	further arguments will be passed

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

calcH\_CO2

*calcH\_CO2*

---

### Description

PRIVATE function: calculates [H+] from an object of class aquaenv and a given [CO2]: by analytically solving the resulting quadratic equation

### Usage

```
calcH_CO2(aquaenv, CO2)
```

### Arguments

aquaenv	object of class aquaenv
CO2	given [CO2] in mol/kg-solution

### Value

calculated [H+] in mol/kg-solution

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

`calcH_TA`

*calcH\_TA*

---

**Description**

PRIVATE function: calculates [H+] from an object of class aquaenv and a given [TA]: first according to Follows2006, if no solution is found after Technicals\$maxiter iterations, uniroot is applied

**Usage**

```
calcH_TA(aquaenv, TA)
```

**Arguments**

aquaenv	object of class aquaenv
TA	given [TA] in mol/kg-solution

**Value**

calculated [H+] in mol/kg-solution

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

`calcSumCO2_TA_CO2`

*calcSumCO2\_TA\_CO2*

---

**Description**

PRIVATE function: calculates [SumCO2] from an object of class aquaenv, a given [TA], and a given [CO2]: by analytically solving the resulting quadratic equation

**Usage**

```
calcSumCO2_TA_CO2(aquaenv, TA, CO2)
```

**Arguments**

aquaenv	object of class aquaenv
TA	given [TA] in mol/kg-solution
CO2	given [CO2] in mol/kg-solution

**Value**

calculated [SumCO2] in mol/kg-solution

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

<code>calcSumCO2_pH_CO2</code>	<i>calcSumCO2_pH_CO2</i>
--------------------------------	--------------------------

---

**Description**

PRIVATE function: calculates [SumCO2] from an object of class aquaenv, a given pH, and a given [CO2]: by analytically solving the resulting equation

**Usage**

```
calcSumCO2_pH_CO2(aquaenv, pH, CO2)
```

**Arguments**

<code>aquaenv</code>	object of class aquaenv
<code>pH</code>	given pH on the free proton scale
<code>CO2</code>	given [CO2] in mol/kg-solution

**Value**

calculated [SumCO2] in mol/kg-solution

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

<code>calcSumCO2_pH_TA</code>	<i>calcSumCO2_pH_TA</i>
-------------------------------	-------------------------

---

**Description**

PRIVATE function: calculates [SumCO2] from an object of class aquaenv, a given pH, and a given [TA]: by analytically solving the resulting quadratic equation

**Usage**

```
calcSumCO2_pH_TA(aquaenv, pH, TA)
```

**Arguments**

aquaenv	object of class aquaenv
pH	given pH on the free proton scale
TA	given [TA] in mol/kg-solution

**Value**

calculated [SumCO2] in mol/kg-solution

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

calcTA

*calcTA*

---

**Description**

PRIVATE function: calculates [TA] from an object of class aquaenv and a given [H+]

**Usage**

calcTA(aquaenv, H)

**Arguments**

aquaenv	object of class aquaenv
H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)given [H+] in mol/kg-solution

**Value**

the calculated [TA]

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

`calcTAMinor`*calcTAMinor*

---

**Description**

PRIVATE function: calculates minor contributions to [TA] from an object of class aquaenv and a given [H+]

**Usage**

```
calcTAMinor(aquaenv, H)
```

**Arguments**

`aquaenv` object of class aquaenv

`H` the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution) given [H+] in mol/kg-solution

**Value**

calculated minor contributions to [TA]

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

`cloneaquaenv`*cloneaquaenv*

---

**Description**

PRIVATE function: clones an object of class aquaenv: it is possible to supply a new value for either TA or pH; the switches speciation, skeleton, revelle, and dsa are obtained from the object to be cloned

**Usage**

```
cloneaquaenv(aquaenv, TA=NULL, pH=NULL, k_co2=NULL, k1k2="roy", khf="dickson")
```



## Arguments

<code>aquaenv</code>	object of class <code>aquaenv</code>
<code>TA</code>	optional new value for TA
<code>pH</code>	optional new value for pH
<code>k_co2</code>	used for TA fitting: give a <code>K_CO2</code> and NOT calculate it from T and S: i.e. <code>K_CO2</code> can be fitted in the routine as well
<code>k1k2</code>	either "roy" (default, Roy1993a) or "lueker" (Lueker2000, calculated with seacarb) for <code>K_CO2</code> and <code>K_HCO3</code> .
<code>khf</code>	either "dickson" (default, Dickson1979a) or "perez" (Perez1987a, calculated with seacarb) for <code>K_HF</code>

## Value

cloned object of class `aquaenv`

## Author(s)

Andreas F. Hofmann ([a.hofmann@nioo.knaw.nl](mailto:a.hofmann@nioo.knaw.nl))

---

<code>convert.standard</code>	<i>convert.standard</i>
-------------------------------	-------------------------

---

## Description

PRIVATE function: converts either the pH scale of a pH value, the pH scale of a dissociation constant ( $K^*$ ), or the unit of a concentration value

## Usage

```
convert.standard(x, vartype, what, S, t, p=0, SumH2SO4=NULL,
                 SumHF=NULL, khf="dickson")
```

## Arguments

<code>x</code>	the object to be converted (pH value, $K^*$ value, or concentration value)
<code>vartype</code>	the type of x, either "pHscale", "KHscale", or "conc"
<code>what</code>	the type of conversion to be done, for pH scales one of "free2tot", "free2sws", "free2nbs", ... (any combination of "free", "tot", "sws", and "nbs"); for concentrations one of "molar2molal", "molar2molin", ... (any combination of "molar" (mol/l), "molal" (mol/kg-H <sub>2</sub> O), and "molin" (mol/kg-solution))
<code>S</code>	salinity (in practical salinity units: no unit)
<code>t</code>	temperature in degrees centigrade
<code>p</code>	gauge pressure (total pressure minus atmospheric pressure) in bars

SumH2SO4	total sulfate concentration in mol/kg-solution; if not supplied this is calculated from S
SumHF	total fluoride concentration in mol/kg-solution; if not supplied this is calculated from S
khf	either "dickson" (default, Dickson1979a) or "perez" (using seacarb, Perez1987a) for K_HF

### Value

converted pH, K\*, or concentration value, attributed with the new unit/pH scale

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

cumulativeplot	<i>cumulativeplot</i>
----------------	-----------------------

---

### Description

PRIVATE function: creates a cumulative plot from the elements of an object of class aquaenv given in what; no return value, just side-effect

### Usage

```
cumulativeplot(aquaenv, xval, what, total=TRUE, palette=NULL,
               device="x11", filename="aquaenv", size=c(12,10),
               ylim=NULL, lwd=2, mgp=c(1.8, 0.5, 0),
               mar=c(3,3,0.5,0.5), oma=c(0,0,0,0),
               legendposition="bottomleft", legendinset=0.05,
               legendlwd=4, bg="white",
               y.intersp=1.2, newdevice, setpar,...)
```

### Arguments

aquaenv	object of class aquaenv
xval	x-value: the independent variable describing a change in elements of an object of class aquaenv
what	vector of names of elements of aquaenv that should be plotted
total	should the sum of all elements specified in what be plotted as well?
palette	a vector of colors to use in the plot (either numbers or names given in colors())
device	the device to plot on; default: "x11" (can also be "eps" or "pdf")
filename	filename to be used if "eps" or "pdf" is selected for device
size	the size of the plot device; default: 12 (width) by 10 (height) inches

<code>ylim</code>	standard plot parameter; if not supplied it will be calculated by an adaptation of <code>range()</code> of the elements to plot
<code>lwd</code>	standard plot parameter; width of the lines in the plot
<code>mgp</code>	standard plot parameter; default: axis title on line 1.8, axis labels on line 0.5, axis on line 0
<code>mar</code>	standard plot parameter; default: margin of 3 lines bottom and left and 0.5 lines top and right
<code>oma</code>	standard plot parameter; default: no outer margin
<code>legendposition</code>	position of the legend
<code>legendinset</code>	standard legend parameter inset
<code>legendlwd</code>	standard legend parameter lwd: line width of lines in legend
<code>bg</code>	standard legend parameter: default background color: white
<code>y.intersp</code>	standard legend parameter; default: 1.2 lines space between the lines in the legend
<code>newdevice</code>	flag: if TRUE, new plot device is opened
<code>setpar</code>	flag: if TRUE parameters are set with the function <code>par</code>
<code>...</code>	further arguments will be passed

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

`dAdH_bi`

*dAdH\_bi*

---

### Description

PRIVATE function: calculates the derivative of  $[A(2-)]$  of a bivalent acid with respect to  $[H+]$

### Usage

```
dAdH_bi(H, SumA, K1, K2)
```

### Arguments

<code>H</code>	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)
<code>SumA</code>	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
<code>K1</code>	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
<code>K2</code>	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dAdH\_tri

*dAdH\_tri*

---

**Description**

PRIVATE function: calculates the derivative of  $[A(3-)]$  of a trivalent acid with respect to  $[H+]$

**Usage**

dAdH\_tri(H, SumA, K1, K2, K3)

**Arguments**

H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)
SumA	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K3	the third dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dAdH\_uni

*dAdH\_uni*

---

**Description**

PRIVATE function: calculates the derivative of  $[A(-)]$  of a univalent acid with respect to  $[H+]$

**Usage**

dAdH\_uni(H, SumA, K)

### Arguments

H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)
SumA	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K	the dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dH2AdH_bi	<i>dH2AdH_bi</i>
-----------	------------------

---

### Description

PRIVATE function: calculates the derivative of [H2A] of a bivalent acid with respect to [H+]

### Usage

dH2AdH\_bi(H, SumA, K1, K2)

### Arguments

H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)
SumA	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dH2AdH\_tri

*dH2AdH\_tri*

---

### Description

PRIVATE function: calculates the derivative of [H2A(-)] of a trivalent acid with respect to [H+]

### Usage

dH2AdH\_tri(H, SumA, K1, K2, K3)

### Arguments

H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)
SumA	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K3	the third dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dH3AdH\_tri

*dH3AdH\_tri*

---

### Description

PRIVATE function: calculates the derivative of [H3A] of a trivalent acid with respect to [H+]

### Usage

dH3AdH\_tri(H, SumA, K1, K2, K3)

### Arguments

H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)
SumA	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K3	the third dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dHAdH\_bi

*dHAdH\_bi*

---

### Description

PRIVATE function: calculates the derivative of [HA(-)] of a bivalent acid with respect to [H+]

### Usage

dHAdH\_bi(H, SumA, K1, K2)

### Arguments

H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)
SumA	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dHAdH\_tri

*dHAdH\_tri*

---

### Description

PRIVATE function: calculates the derivative of [HA(2-)] of a trivalent acid with respect to [H+]

### Usage

dHAdH\_tri(H, SumA, K1, K2, K3)

### Arguments

H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)
SumA	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K1	the first dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K2	the second dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K3	the third dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dHAdH\_uni

*dHAdH\_uni*

---

### Description

PRIVATE function: calculates the derivative of [HA] of a univalent acid with respect to [H+]

### Usage

dHAdH\_uni(H, SumA, K)



### Arguments

H	the proton concentration in a unit consistent with all other input variables (e.g. mol/kg-solution)
SumA	the total concentration of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)
K	the dissociation constant of the acid in question in a unit consistent with all other input variables (e.g. mol/kg-solution)

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dTAdH	$dTAdH$
-------	---------

---

### Description

PRIVATE function: calculates the derivative of [TA] with respect to [H+]: the buffer factor

### Usage

dTAdH(ae)

### Arguments

ae                      object of class aquaenv

### Value

derivative of [TA] with respect to [H+]: the buffer factor

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dTAdKdKdS	<i>dTAdKdKdS</i>
-----------	------------------

---

### Description

PRIVATE function: calculates the derivative of [TA] with respect to changes in the dissociation constants (Ks) times the derivative of the dissociation constants with respect to salinity S

### Usage

dTAdKdKdS(ae)

### Arguments

ae                      object of class aquaenv

### Value

derivative of [TA] with respect to changes in the dissociation constants (Ks) times the derivative of the dissociation constants with respect to salinity S

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dTAdKdKdSumH2SO4	<i>dTAdKdKdSumH2SO4</i>
------------------	-------------------------

---

### Description

PRIVATE function: calculates the derivative of [TA] with respect to changes in the dissociation constants (Ks) times the derivative of the dissociation constants with respect to the total sulfate concentration (influence via scale conversion)

### Usage

dTAdKdKdSumH2SO4(ae)

### Arguments

ae                      object of class aquaenv

### Value

derivative of [TA] with respect to changes in the dissociation constants (Ks) times the derivative of the dissociation constants with respect to the total sulfate concentration (influence via scale conversion)

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dTAdKdKdSumHF

*dTAdKdKdSumHF*

---

**Description**

PRIVATE function: calculates the derivative of [TA] with respect to changes in the dissociation constants (Ks) times the derivative of the dissociation constants with respect to the total fluoride concentration (influence via scale conversion)

**Usage**

dTAdKdKdSumHF(ae)

**Arguments**

ae                      object of class aquaenv

**Value**

derivative of [TA] with respect to changes in the dissociation constants (Ks) times the derivative of the dissociation constants with respect to the total fluoride concentration (influence via scale conversion)

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dTAdKdKdT

*dTAdKdKdT*

---

**Description**

PRIVATE function: calculates the derivative of [TA] with respect to changes in the dissociation constants (Ks) times the derivative of the dissociation constants with respect to temperature T

**Usage**

dTAdKdKdT(ae)

**Arguments**

ae                      object of class aquaenv

### Value

derivative of [TA] with respect to changes in the dissociation constants (Ks) times the derivative of the dissociation constants with respect to temperature T

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

dTAdKdKdp

*dTAdKdKdp*

---

### Description

PRIVATE function: calculates the derivative of [TA] with respect to changes in the dissociation constants (Ks) times the derivative of the dissociation constants with respect to gauge pressure p

### Usage

dTAdKdKdp(ae)

### Arguments

ae                      object of class aquaenv

### Value

derivative of [TA] with respect to changes in the dissociation constants (Ks) times the derivative of the dissociation constants with respect to gauge pressure p

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

deltaPlnK

*deltaPlnK*

---

### Description

PRIVATE function: the generic function for the pressure correction for dissociation constants and solubility products according to Millero1995

### Usage

deltaPlnK(T, d, coeff)

### Arguments

<code>T</code>	temperature in Kelvin
<code>d</code>	the depth in meters
<code>coeff</code>	a vector containing the coefficients <code>a0</code> , <code>a1</code> , <code>a2</code> , <code>b0</code> , <code>b1</code> , <code>b2</code> for the respective dissociation constant or solubility product

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

### References

Millero1995, corrected by Lewis1998

---

<code>from.data.frame</code>	<i>from.data.frame</i>
------------------------------	------------------------

---

### Description

PRIVATE function: creates an object of class `aquaenv` from a data frame (e.g. as supplied from the numerical solver of a dynamic model)

### Usage

```
from.data.frame(df)
```

### Arguments

<code>df</code>	data frame
-----------------	------------

### Value

object of class `aquaenv`

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

<code>lnK</code>	<i>lnK</i>
------------------	------------

---

**Description**

PRIVATE function: generic formula (see publication associated with AquaEnv) for K calculations that use the natural logarithm (ln)

**Usage**

`lnK(A, B, C, D, E, T)`

**Arguments**

A	coefficient A
B	coefficient B
C	coefficient C
D	coefficient D
E	coefficient E
T	temperature in Kelvin

**Value**

the ln of the K associated with the coefficients

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

<code>logK</code>	<i>logK</i>
-------------------	-------------

---

**Description**

PRIVATE function: generic formula (see publication associated with AquaEnv) for K calculations that use the decadal logarithm (log)

**Usage**

`logK(A, B, C, D, E, T)`

**Arguments**

A	coefficient A
B	coefficient B
C	coefficient C
D	coefficient D
E	coefficient E
T	temperature in Kelvin

**Value**

the log of the K associated with the coefficients

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

`molal2molin`

*molal2molin*

---

**Description**

PRIVATE function: calculates the conversion factor converting from molality (mol/kg-H<sub>2</sub>O) to molinity (mol/kg-solution) from salinity S

**Usage**

`molal2molin(S)`

**Arguments**

S	salinity S in practical salinity units (i.e. no unit)
---	---

**Value**

the conversion factor from molality (mol/kg-H<sub>2</sub>O) to molinity (mol/kg-solution)

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

**References**

Roy1993b (the carbonic acid paper), DOE1994

---

`opendevic`

*opendevic*

---

### Description

PRIVATE function: opens a device for plotting; no return value, just side-effect

### Usage

```
opendevic(device, size, filename)
```

### Arguments

<code>device</code>	either "x11", "eps", or "pdf"
<code>size</code>	size of the plot device in the form c(width, height)
<code>filename</code>	filename to use if "eps" or "pdf" is used

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

`plotall`

*plotall*

---

### Description

PRIVATE function: plots all elements of an object of class aquaenv; no return value, just side-effect

### Usage

```
plotall(aquaenv, xval, ...)
```

### Arguments

<code>aquaenv</code>	object of class aquaenv
<code>xval</code>	x-value: the independent variable describing a change in elements of an object of class aquaenv
<code>...</code>	further arguments will be passed

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)



---

revelle

*revelle*

---

### Description

PRIVATE function: calculates the revelle factor

### Usage

```
revelle(ae)
```

### Arguments

ae                      object of class aquaenv

### Value

the revelle factor

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

### References

Sundquist1979, Zeebe2001, Emerson2008

---

scaleconvert

*scaleconvert*

---

### Description

PRIVATE function: provides pH scale conversion factors (caution: the activity coefficient for H<sup>+</sup> (needed for NBS scale conversions) is calculated with the Davies equation (Zeebe2001) which is only accurate up to ionic strengthes of  $I = 0.5$ )

### Usage

```
scaleconvert(S, t, p=0, SumH2SO4=NULL, SumHF=NULL, khf="dickson")
```

### Arguments

<b>S</b>	salinity S in practical salinity units (i.e. no unit)
<b>t</b>	temperature in degrees centigrade
<b>p</b>	gauge pressure (total pressure minus atmospheric pressure) in bars
<b>SumH2SO4</b>	total sulfate concentration in mol/kg-solution; if not supplied this is calculated from S
<b>SumHF</b>	total fluoride concentration in mol/kg-solution; if not supplied this is calculated from S
<b>khf</b>	either "dickson" (default, Dickson1979a) or "perez" (using seacarb, Perez1987a) for K_HF

### Value

a list of conversion factors "free2tot", "free2sws", etc.

### Author(s)

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

### References

Dickson1984, DOE1994, Zeebe2001

---

<b>seaconc</b>	<i>seaconc</i>
----------------	----------------

---

### Description

PRIVATE function: calculates concentrations of constituents of natural seawater from a given salinity S

### Usage

`seaconc(spec, S)`

### Arguments

<b>spec</b>	constituent of seawater (chemical species) of which the concentration should be calculated. can be any name of the vectors ConcRelCl and MeanMolecularWeight: "Cl", "SO4", "Br", "F", "Na", "Mg", "Ca", "K", "Sr", "B", "S"
<b>S</b>	salinity S in practical salinity units (i.e. no unit)

### Value

concentration of the constituent of seawater specified in spec in mol/kg-solution (molinity): this is determined by the data in ConcRelCl and MeanMolecularWeight

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

**References**

DOE1994

---

`seadensity`

*seadensity*

---

**Description**

PRIVATE function: calculates seawater density (in kg/m3) from temperature (in degrees centigrade) and salinity

**Usage**

`seadensity(S, t)`

**Arguments**

<code>S</code>	salinity S in practical salinity units (i.e. no unit)
<code>t</code>	temperature in degrees centigrade

**Value**

seawater density in kg/m3

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

**References**

Millero1981, DOE1994

---

<code>selectplot</code>	<i>selectplot</i>
-------------------------	-------------------

---

### Description

PRIVATE function: plots just the elements of an object of class `aquaenv` given in `what`; no return value, just side-effect

### Usage

```
selectplot(aquaenv, xval, what, mfrow=c(1,1), size=c(7,7), ...)
```

### Arguments

<code>aquaenv</code>	object of class <code>aquaenv</code>
<code>xval</code>	x-value: the independent variable describing a change in elements of an object of class <code>aquaenv</code>
<code>what</code>	vector of names of elements of <code>aquaenv</code> that should be plotted
<code>mfrow</code>	standard plot parameter; default: just one plot
<code>size</code>	the size of the plot device; default: 7 (width) by 7 (height) inches
<code>...</code>	further arguments will be passed

### Author(s)

Andreas F. Hofmann ([a.hofmann@nioo.knaw.nl](mailto:a.hofmann@nioo.knaw.nl))

---

<code>splitS_K_CO2</code>	<i>splitS_K_CO2</i>
---------------------------	---------------------

---

### Description

PRIVATE function: returns the intersection of the formulae for `K_CO2` for  $S < 5$  and  $S \geq 5$

### Usage

```
splitS_K_CO2(T)
```

### Arguments

<code>T</code>	temperature in Kelvin
----------------	-----------------------

### Value

the value for `S` where the two formulae intersect at temperature `T`

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)

---

`splitS_K_HCO3`

*splitS\_K\_HCO3*

---

**Description**

PRIVATE function: returns the intersection of the formulae for K\_HCO3 for  $S < 5$  and  $S \geq 5$

**Usage**

`splitS_K_HCO3(T)`

**Arguments**

T                      temperature in Kelvin

**Value**

the value for S where the two formulae intersect at temperature T

**Author(s)**

Andreas F. Hofmann (a.hofmann@nioo.knaw.nl)