SolveSAPHE-r2: a new versatile 4x4 engine for carbonate system pH calculations

EGU21-9477

Guy Munhoven

Dept. of Astrophysics, Geophysics and Oceanography, University of Liège
eMail: Guy.Munhoven@uliege.be

vEGU21: Gather Online – Session BG4.2 – 27th April 2021

Outline of this Display

Chemical equilibria between carbonate species

$$\begin{aligned} CO_{2(aq)} + 2\ H_2O &\rightleftharpoons HCO_3^- + H_3O^+ \\ HCO_3^- + H_2O &\rightleftharpoons CO_3^{2-} + H_3O^+ \end{aligned}$$

Special roles of different species

- atmospheric $p_{CO_2} \longleftrightarrow [CO_{2(aq)}]_{\text{surface}}$
- $CaCO_3$ deposition $\longleftrightarrow [CO_3^{2-}]_{\text{deep}}$

Speciation requirements

Four unknowns: $[CO_{2(aq)}]$, $[HCO_3^-]$, $[CO_3^{2-}]$, $[H^+]$ (pH)

& two equilibrium relationships

$\Rightarrow$ two independent measurables required

Carbonate Chemistry: Measurables

- Dissolved Inorganic Carbon (DIC):

  $$C_\mathrm{T} = [\mathrm{CO}_{2(\mathrm{aq})}] + [\mathrm{HCO}_3^-] + [\mathrm{CO}_3^{2-}]$$

- Total Alkalinity:

  $$\mathrm{Alk}_\mathrm{T} = [\mathrm{HCO}_3^-] + 2\,[\mathrm{CO}_3^{2-}] + [\mathrm{B(OH)}_4^-] + \ldots$$

- $p\mathrm{H}$
- $[\mathrm{CO}_{2(\mathrm{aq})}]$
- $[\mathrm{CO}_3^{2-}]$

The Alkalinity-pH Equation

Total Alkalinity Definition

$$\mathrm{Alk_T} = \mathrm{Alk_C} + \mathrm{Alk_B} + \cdots + [\mathrm{OH^-}] - [\mathrm{H^+}]_\mathrm{f}$$

where

$$\mathrm{Alk_C} = [\mathrm{HCO_3^-}] + 2\,[\mathrm{CO_3^{2-}}] = \frac{K_1[\mathrm{H^+}] + 2K_1K_2}{[\mathrm{H^+}]^2 + K_1[\mathrm{H^+}] + K_1K_2}\,C_\mathrm{T}$$

is *carbonate alkalinity*, K_1, K_2 being the first and second dissociation constants for carbonic acid and C_T the DIC concentration;

$$\mathrm{Alk_B} = [\mathrm{B(OH)_4^-}] = \frac{K_\mathrm{B}}{[\mathrm{H^+}] + K_\mathrm{B}}\,B_\mathrm{T}$$

is *borate alkalinity*, K_B being the dissociation constant for boric acid, and B_T is the total borate concentration;

Total Alkalinity Definition . . .

$$\mathrm{Alk_T} = \mathrm{Alk_C} + \mathrm{Alk_B} + \cdots + [\mathrm{OH^-}] - [\mathrm{H^+}]_\mathrm{f}$$

"$\cdots$" stands for further contributions from additional acid systems, all of which can be expanded into similar expressions than $\mathrm{Alk_C}$ and $\mathrm{Alk_B}$ (see Munhoven (2013) for details);

and where

$$[\mathrm{OH^-}] - [\mathrm{H^+}]_\mathrm{f} = \frac{K_\mathrm{W}}{[\mathrm{H^+}]} + \frac{[\mathrm{H^+}]}{s}$$

is the *water self-ionisation alkalinity*, $[\mathrm{H^+}]_\mathrm{f}$ being the *free* concentration $\mathrm{H^+}$, K_W the dissociation constant for water and s a factor to convert $[\mathrm{H^+}]$ (dependent on the *p*H scale used) to $[\mathrm{H^+}]_\mathrm{f}$.

Converting to the Alkalinity-*p*H Equation

- Rewrite the definition of $\mathrm{Alk_T}$ with its contributions expanded in terms of $[\mathrm{H}^+]$ and the total acid system concentrations as a parametric equation in $[\mathrm{H}^+]$:

$$R([\mathrm{H}^+];\ C_\mathrm{T}, \mathrm{B_T}, \ldots, \mathrm{Alk_T}) = 0$$

- Metadata required:
  T, S, P to calculate K_1, K_2, K_B, . . . , K_W and s
- Given $C_\mathrm{T}, \mathrm{B_T}, \ldots, \mathrm{Alk_T} \Rightarrow$ solve for $[\mathrm{H}^+]$ . . . if possible (?)

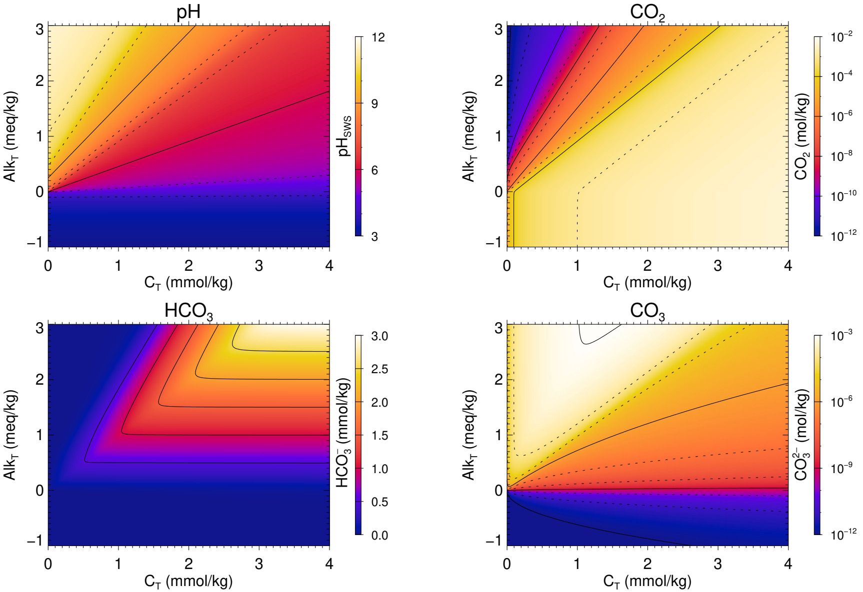
The SolveSAPHE Approach

Solver Suite for Alkalinity-PH Equations

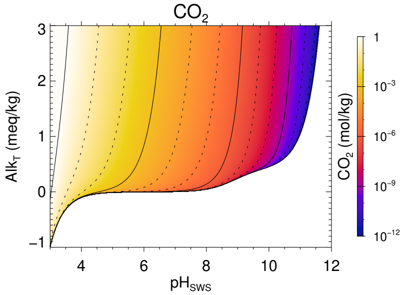
- Mathematical analysis of the $\mathrm{Alk_T}$ & C_T problem
- Exactly one pH root for every $\mathrm{Alk_T}$ & C_T pair of data
- Brackets for the root can be derived from input data alone
- Efficient starting procedure for iterative solvers
- Fast iterative solvers with convergence guarantee
- Fortran 90 codes in the supplement and on Zenodo: doi:10.5281/zenodo.3752250
- Adopted in

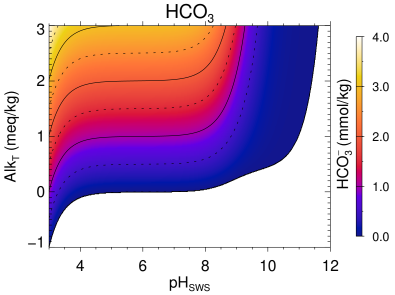
  MITgcm, Delft3D, CSIRO Environmental Modelling Suite, mocsy 2.0 ($\rightarrow$ CMIP6 Ocean Modelling Intercomparison Project (OMIP)), ...

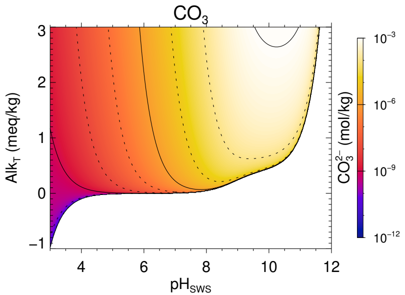
Speciation from Alk_T & C_T



Towards Alk_T & CO_2 – Alk_T & HCO_3^- – Alk_T & CO_3^{2-}







- Blank areas: $Alk_C < 0$
- CO_2 and HCO_3^- isolines continuous and monotonous:
  Alk_T & CO_2 $\rightarrow$ one pH root
  Alk_T & HCO_3^- $\rightarrow$ one pH root
- CO_3^{2-} isolines not monotonous:
  Alk_T & CO_3^{2-} $\rightarrow$ 0, 1, 2 roots

- Extend SolveSAPHE to work with
  Alk_T & CO_2, Alk_T & HCO_3^- and Alk_T & CO_3^{2-} as well
- Alkalinity-*p*H equation with alternative expansions for Alk_C:

$$\mathrm{Alk_C} = \frac{K_1[\mathrm{H^+}] + 2K_1K_2}{[\mathrm{H^+}]^2}\,[\mathrm{CO_2}]$$

$$\mathrm{Alk_C} = \frac{K_1[\mathrm{H^+}] + 2K_1K_2}{K_1[\mathrm{H^+}]}\,[\mathrm{HCO_3^-}]$$

$$\mathrm{Alk_C} = \frac{K_1[\mathrm{H^+}] + 2K_1K_2}{K_1K_2}\,[\mathrm{CO_3^{2-}}]$$

- Mathematical analysis of the properties of the three modified forms of the alkalinity-*p*H equation
- Alk_T & CO_2 and Alk_T & HCO_3^-
  - ⇒ exactly one *p*H root for any given pair of data
- Alk_T & CO_3^{2-}
  - ⇒ none, one or two roots for a given pair of data
  - ⇒ if two roots: roots can be separated
  - ⇒ diagnosis can be automated
- Brackets for any root can be derived from input data

- Two root-finding algorithms
  - Newton-Raphson – regula falsi/bisection
  - Secant – regula falsi/bisection
  - Both with guaranteed convergence
- Fully autonomous
  - Only sample characteristics required
  - Efficient starting procedures for each case
  - Complete characterisation of the Alk_T & CO_3^{2-} problem
- Not as fast as SolveSAPHE v.1
  - Mathematical properties of the equations less favourable
  - Slow-down due to additional safeguards required
- Fortran 90 codes in the supplement to the paper
- Ported to R by J.-M. Epitalon (9th March 2021)
  $\Rightarrow$ https://cran.r-project.org/package=SolveSAPHE

SolveSAPHE v.1 and SolveSAPHE-r2: To learn more

G. Munhoven (2013) Mathematics of the total alkalinity-pH equation - pathway to robust and universal solution algorithms: the SolveSAPHE package v1.0.1, *Geosci. Model Dev.*, 6, 1367–1388, doi:10.5194/gmd-6-1367-2013

G. Munhoven (2021) SolveSAPHE-r2: revisiting and extending the Solver Suite for Alkalinity-PH Equations for usage with CO_2, HCO_3^- or CO_3^{2-} input data. *Geosci. Model Dev. Discuss.*, [preprint], doi:10.5194/gmd-2020-447, in review.