

# Contents

<b>4 A data assimilating state-space model for algal growth under controlled conditions within a photobioreactor</b>	<b>2</b>
4.1 Introduction . . . . .	2
4.2 Methods . . . . .	2
4.2.1 Data Model: Data description and collection methods . . . . .	2
4.2.2 Process model: Carbon chemistry . . . . .	2
4.2.3 Process model: Gas transfer equilibrium concentrations for O <sub>2</sub> and CO <sub>2</sub> . . . . .	3
4.2.4 Process model: Photosynthesis and respiration . . . . .	5
4.2.5 Process model: Ordinary differential equations . . . . .	6
4.2.6 Parameter Model: Priors . . . . .	9
4.3 Results . . . . .	10
4.3.1 Carbon chemistry iterative solution . . . . .	10
4.3.2 Posteriors . . . . .	15
<b>A LiBbi model code</b>	<b>20</b>
<b>Bibliography</b>	<b>33</b>

## Chapter 4

# A data assimilating state-space model for algal growth under controlled conditions within a photobioreactor

### 4.1 Introduction

Microalgae are tiny organisms..

### 4.2 Methods

#### 4.2.1 Data Model: Data description and collection methods

O<sub>2</sub>, pH, DIC and alk observations collected over 4 days.

Gas, temperature, light and dilution rate were used as forcings.

#### 4.2.2 Process model: Carbon chemistry

To calculate the carbon chemistry of the photo-bioreactor, we would ideally use CO2SYS [4] to calculate  $\text{HCO}_3^-$ , CO<sub>2</sub>, CO<sub>3</sub> and pH. CO2SYS is a program developed for CO<sub>2</sub> system calculations (CO2SYS) that calculates and returns a detailed state of the carbonate system of oceanographic water samples in seawater and freshwater [4]. It uses two of the four measurable carbonate system parameters (total alkalinity, total inorganic CO<sub>2</sub>, pH, and either fugacity fCO<sub>2</sub> or partial pressure of CO<sub>2</sub>) to calculate the other two parameters at a set of input conditions (temperature and pressure).

To incorporate CO2SYS into LiBbi for solving carbon chemistry on the timescale of the microalgae model, we explicitly define 2 iterations of the Newton-Raphson

method for finding approximations to roots of real valued functions. The Newton-Raphson method is an iterative process considering a function, its derivative and an initial starting value. Vital to the convergence of the Newton-Raphson method is a good starting value. To provide a good starting value, we randomly sample from a range of CO2SYS input parameters (temperature = 20-30, salinity = 30-40, DIC = 200-2500, and alkalinity = 1500-3000) and fit an approximating equation to pH as a function of DIC, S and T (alk?). This gives us a close initial starting value for the Newton-Raphson method.

Converges in 2-3 iterations

Choice of H<sub>2</sub>CO<sub>3</sub> and HCO<sub>3</sub><sup>-</sup> dissociation constants K<sub>1</sub> and K<sub>2</sub> was Mehrbach (refit BY DICKSON AND MILLERO) [BM: After the iterative approach is finalised, the K<sub>1</sub> and K<sub>2</sub> constants are adjusted based on measurements taken during the experiment, K<sub>1</sub>\*1.23 and K<sub>2</sub>\*0.53 measured during experiment] temperature: 2-35, salinity: 20-40, Seawater scale, Artificial seawater.

The CO2SYS Matlab version [6] was used to produce values of CO<sub>2</sub> and HCO<sub>3</sub><sup>-</sup> across DIC range 200-2500. Approximating equations were fit

Total inorganic CO<sub>2</sub> (TCO<sub>2</sub>) is the sum of the dissolved CO<sub>2</sub>, the carbonate (CO<sub>3</sub><sup>2-</sup>), and the bicarbonate (HCO<sub>3</sub><sup>-</sup>).

#### **4.2.3 Process model: Gas transfer equilibrium concentrations for O<sub>2</sub> and CO<sub>2</sub>**

The equilibrium concentration for CO<sub>2</sub> solubility in water CO<sub>2H</sub> (μmol/L) is calculated using Henry's law,

$$CO_{2H} = K_{CO_2} * fCO_2 * 1.0220 * 1e6 \quad (4.1)$$

where fCO<sub>2</sub> (atm) is the fugacity or approximately the partial pressure of CO<sub>2</sub>, 1.0220 is the density of seawater (kg/L) at salinity 34 ppt and temperature 27°C

[5] [2].  $K_{CO_2}$  (mol/kg<sub>soln</sub>/atm) is the solubility of gas in seawater [BM: ask Chris: solubility of gas? is this right] and is calculated from the fitted van't Hoff equation and the logarithmic Setchenow salinity dependence [7],

$$K_{CO_2} = \exp(-60.2409 + 93.4517(100/T_K) + 23.3585 * \ln(T_K/100) + S(0.023517 - 0.023656(T_K/100) + 0.0047036(T_K/100)^2)) \quad (4.2)$$

where  $T_K$  is the temperature (K) and  $S$  is salinity (ppt).

Similarly the equilibrium concentration for  $O_2$  solubility in water  $O_{2H}$  is calculated using Henry's law,

$$O_{2H} = K_{O_2} * f_{O_2} * 1.0220 * 1e - 6 \quad (4.3)$$

where  $f_{O_2}$  (atm) is the fugacity or approximately the partial pressure of  $O_2$ , 1.0220 is the density of seawater (kg/L) at salinity 34 ppt and temperature 27°C [5] [2], and  $K_{O_2}$  (mol/kg<sub>soln</sub>/atm) is the solubility of oxygen in seawater with an adjusted salinity dependence [1],

$$K_{O_2} = (\exp(-1282.8704 + 36619.96/T_K + 223.1396 * \log(T_K) - 0.354707 * T_K + S * (5.957e - 3 - 3.7353/T_K) + 3.68e - 6 * S^2))/0.2094e - 6 \quad (4.4)$$

where  $T_K$  is the temperature (K) and  $S$  is salinity (ppt).

The equilibrium concentrations for  $O_2$  and  $CO_2$  are modelled together with the gas turning on and off during the experiment, as

$$kLA_{O_2}\xi(O_{2H} - O_2) \quad (4.5)$$

$$0.893kLA_{O_2}\xi(CO_{2H} - CO_2) \quad (4.6)$$

where  $\xi$  is the gas state (1= on, 0= off), and  $kLA_{O_2}$  is the mass transfer coefficient for air (d<sup>-1</sup>), and 0.893 is the ratio between measured  $O_2$  and  $CO_2$  mass transfer constants [3].

#### 4.2.4 Process model: Photosynthesis and respiration

Net photosynthesis

$$dDIC/dt = -P_1 * I * mm + R_1 \quad (4.7)$$

$$dO_2/dt = \frac{P_1 * I * mm - R_1}{R_Q} \quad (4.8)$$

Photosynthesis ( $P_1$ ) and respiration ( $R_1$ ) are both modelled as random walks, by taking  $P$  and  $R$ , previously constant parameters, and replacing them by  $P_1(t)$  and  $R_1(t)$ . Here, we take  $P_1(t)$  and  $R_1(t)$  to be such that

$$P_1(t + \Delta t) = P_1(t) + r_P$$

$$R_1(t + \Delta t) = R_1(t) + r_R$$

where  $r_P \sim N(0, \sigma_{r_P})$ ,  $r_R \sim N(0, \sigma_{r_R})$ , and  $\Delta t$  is the length of discrete time-step. For the purpose of the Bayesian analysis here,  $\sigma_{r_P}$  and  $\sigma_{r_R}$  are treated as a parameter to be inferred.

$R_Q$  is the respiratory quotient, the ratio of  $CO_2$  produced and  $O_2$  consumed by a cell.

PAC is Photosynthetically Active Carbon, this is the type of carbon that the microalgae use for photosynthesis. This can be  $CO_2$ ,  $HCO_3^-$ , or a combination of both, eg  $PAC = CO_2 + HCO_3^-$  if the microalgae are using both carbon dioxide and bicarbonate for photosynthesis.

$$PAC = HCO_3^- \quad (4.9)$$

$$mm = \frac{PAC}{K_m + PAC} \quad (4.10)$$

#### 4.2.5 Process model: Ordinary differential equations

Ode's:

$$\begin{aligned} \frac{\partial DIC}{\partial t} = & \quad \text{Rate} \quad \text{flux into cells} \quad \text{gas transfer} \quad \text{dilution} \\ & -(P - R) + \hat{Q}^{air} kLa_{CO_2}^{air} (CO_2^{air} - CO_2) + \frac{Q^M}{V} (DIC^M - DIC) \\ & + \hat{Q}^{co2} kLa_{CO_2}^{co2} (CO_2^{co2} - CO_2) \end{aligned} \quad (4.11)$$

$$\begin{aligned} \frac{\partial O_2}{\partial t} = & \frac{1}{R_Q} (P - R) + \hat{Q}^{air} kLa_{O_2}^{air} (O_2^{air} - O_2) + \frac{Q^M}{V} (O_2^M - O_2) \\ & + \hat{Q}^{co2} kLa_{O_2}^{co2} (O_2^{co2} - O_2) \end{aligned} \quad (4.12)$$

$$\begin{aligned} \frac{\partial TA}{\partial t} = & R_R (P - R) + \frac{Q^M}{V} (TA^M - TA) \end{aligned} \quad (4.13)$$

	Symbol	Description	Prior / Value	Unit
State variable initial conditions	$DIC^0$	Dissolved inorganic carbon		$\mu\text{M/L}$
	$O_2^0$	Oxygen		$\mu\text{M/L}$
	$TA^0$	Total alkalinity		$\mu\text{M/L}$
	$P^0$	Rate of photosynthesis		$\mu\text{M/L/day}$
	$R^0$	Rate of respiration		$\mu\text{M/L/day}$
	$\text{pH}^0$	-		$\log_{10}(-\text{mol/L H}^+)$
	$CO_2^0$	Carbon dioxide		$\mu\text{M/L}$
	$HCO_3^{-0}$	Bicarbonate		$\mu\text{M/L}$
	$CO_3^{2-0}$	Carbonate		$\mu\text{M/L}$
Gas transfer terms	$\hat{Q}^{air}$	indicator for flow in air line	0 or 1	-
	$x_{CO_2}^{air}$	mole fraction of $CO_2$ atmosphere	400	ppm
	$CO_{2H}$	Equilibrium $CO_2$ concentration	Eq. 4.1	$\mu\text{M/L}$
	$CO_2^{air}$	sat $CO_2$ conc with atmosphere	$x_{CO_2}^{air} CO_{2H}$	
	$kLa_{CO_2}^{air}$	Mass transfer coefficient for $CO_2$	$0.893 kLa_{O_2}^{air}$	$\text{day}^{-1}$
	$x_{O_2}^{air}$	mole fraction of $O_2$ atmosphere	0.2094	atm

	$O_{2H}$	Equilibrium O <sub>2</sub> concentration	Eq. 4.3	$\mu\text{M/L}$
	$O_2^{air}$	sat O <sub>2</sub> conc with atmosphere	$x_{O_2}^{air} O_{2H}$	
	$\tau$	half-life of $kLa^{air}_{O_2}$	range(2-20)	$\text{min}^{-1}$
	$kLa^{air}_{O_2}$	Mass transfer coefficient for O <sub>2</sub>	$\ln(2) * 24 * 60 / \tau$	$\text{day}^{-1}$
Dilution terms	$Q^M$	rate		ml/day
	$V$	Volume of the reactor	500	ml
	$DIC^M$	Media dissolved inorganic carbon	1724.20	$\mu\text{M/L}$
	$O_2^M$	Media oxygen concentration	226.65	$\mu\text{M/L}$
	$TA^M$	Media total alkalinity	1797.90	$\mu\text{M/L}$
Other dilution terms	$\hat{Q}^{CO_2}$	indicator for dilution	0 or 1	-
	$x_{O_2}^{CO_2}$	mole fraction of	0	-
	$O_2^{CO_2}$	sat CO <sub>2</sub> conc with CO <sub>2</sub>	$x_{O_2}^{CO_2} O_{2H}$	
	$kLa_{O_2}^{CO_2}$	mass transfer coefficient		$\text{day}^{-1}$
	$O_2^{CO_2}$			
	$x_{CO_2}^{CO_2}$	mole fraction of	1	ppm
	$CO_2^{CO_2}$	sat CO <sub>2</sub> conc with CO <sub>2</sub>	$x_{CO_2}^{CO_2} CO_{2H}$	
	$kLa_{CO_2}^{CO_2}$	mass transfer coefficient	$0.893 kLa_{O_2}^{CO_2}$	$\text{day}^{-1}$
	$CO_2^{CO_2}$			

Symbol	Variable	Units
DIC	Dissolved inorganic carbon concentration	$\mu\text{mol/L}$
O <sub>2</sub>	Oxygen	$\mu\text{mol/L}$
pH	-	$\log_{10}(-\text{mol/L H}^+)$
CO <sub>2</sub>	Carbon dioxide	$\mu\text{mol/L}$
HCO <sub>3</sub> <sup>-</sup>	Bicarbonate	$\mu\text{mol/L}$
CO <sub>3</sub> <sup>2-</sup>	Carbonate	$\mu\text{mol/L}$
PAC	Photosynthetically active carbon	$\mu\text{mols/L}$
mm	-	-
kLA <sub>O2</sub>	Mass transfer coefficient for O <sub>2</sub>	d <sup>-1</sup>
CO <sub>2H</sub>	Equilibrium CO <sub>2</sub> concentration	$\mu\text{mols/L}$
K <sub>O2</sub>	Solubility of gas	mol/kg <sub>soln</sub> /atm
K <sub>CO2</sub>	Solubility of gas	mol/kg <sub>soln</sub> /atm
TA	Total alkalinity	$\mu\text{mols/L}$
S	Salinity	ppt
fCO2	Fugacity/CO <sub>2</sub> partial pressure	atm
fO2	Fugacity/O <sub>2</sub> partial pressure	atm
K <sub>m</sub>	Carbon restriction	$\mu\text{mols/L}$
P	Photosynthesis rate	$\mu\text{mols/L/day}$
R	Respiration rate	$\mu\text{mols/L/day}$
R <sub>R</sub>	Redfield ratio	-
R <sub>Q</sub>	Respiratory quotient	-

Table 4.2 : Table of variables and parameters.



Symbol	Variable	Units
I	Light Intensity	normalised to 0-1
T	Temperature	◦ C
$T_K$	Temperature	K
$\xi$	gasflow	on/off (1,0)

Table 4.3 : Table of Forcings

#### 4.2.6 Parameter Model: Priors

Decide whether the parameters vary in time or not.

Parameter	Prior	Proposal
S	34	*
fCO2	397e-6	*
fO2	0.21	*
$kLA_{O_2}$	$\text{LogNormal}(\log(200.0), 0.5)$	$\text{LogNormal}(\log(kLA_{O_2}), 0.5\text{prop}_{std})$
$K_m$	$\text{LogNormal}(\log(200.0), 0.8)$	$\text{LogNormal}(\log(K_m), 0.8\text{prop}_{std})$
$R_R$	$\text{Uniform}(0.0001, 0.2)$	$\text{TrunNormal}(R_R, 0.2\text{prop}_{std}, 0.0001, 0.2)$
$R_Q$	$\text{Uniform}(0.66, 1)$	$\text{TrunNormal}(R_Q, 0.2\text{prop}_{std}, 0.66, 1.0)$
$\sigma_P$	$\text{Normal}(0.05, 0.01)$	$\text{Normal}(\sigma_P, 0.01\text{prop}_{std})$
$\sigma_R$	$\text{Normal}(0.01, 0.001)$	$\text{Normal}(\sigma_R, 0.001\text{prop}_{std})$

Table 4.4 : Table of Parameters, their priors and proposal distributions. \* indicates the parameter was held fixed. ( $\text{prop}_{std}=0.1$ )

## 4.3 Results

### 4.3.1 Carbon chemistry iterative solution

Total Sulfur

$$\begin{aligned}
 TS &= \frac{0.14}{96.062} * \frac{S}{1.8065} \\
 IS &= 19.924 * \frac{S}{(1000.0 - 1.005 * S)} \\
 KS_{int} &= -\frac{4276.1}{T_K} + 141.328 - 23.093 * \log(T_K) + \left(-\frac{13856.0}{T_K} + 324.57 \right. \\
 &\quad \left. - 47.986 * \log(T_K)\right) * \sqrt{IS} + \left(\frac{35474}{T_K} - 771.54 + 114.723 * \log(T_K)\right) * IS \\
 &\quad - \frac{2698}{T_K} * IS^{1.5} + \frac{1776}{T_K} * IS^2 \\
 KS &= \exp(KS_{int}) * (1 - 0.001005 * S)
 \end{aligned}$$

Fluorine

$$\begin{aligned}
 TF &= 0.000067 * S / 18.9984 / 1.80655 \\
 KF &= \exp\left(-\left(-\frac{874.0}{T_K} - 0.111 * \sqrt{S} + 9.68\right)\right) \\
 SWS_{2T} &= \frac{\left(1 + \frac{TS}{KS}\right)}{\left(1 + \frac{TS}{KS} + \frac{TF}{KF}\right)} \\
 Free_{2T} &= 1 + \frac{TS}{KS}
 \end{aligned}$$

H2O dissoc

$$KW = \exp(148.9802 - \frac{13847.26}{T_K} - 23.6521 * \log(T_K) + (\frac{118.67}{T_K} - 5.977 + 1.0495 * \log(T_K)) * \sqrt{S} - 0.01615 * S)$$

Boron

$$KB = \exp((-8966.90 - 2890.53 * \sqrt{S} - 77.942 * S + 1.728 * S * \sqrt{S} - 0.0996 * S^2)/T_K + 148.0248 + 137.1942 * \sqrt{S} + 1.62142 * S - (24.4344 + 25.085 * \sqrt{S} + 0.2474 * S) * \log(T_K) + 0.053105 * \sqrt{S} * T_K)$$

$$TB = 0.0004326 * \frac{S}{35}$$

Carbon eq constants

$$K1 = 10^{(-(\frac{3633.86}{T_K} - 61.2172 + 9.6777 * \log(T_K) - 0.011555 * S + 0.0001152 * S^2))} * 1.23$$

$$K2 = 10^{(-(\frac{471.8}{T_K} + 25.9290 - 3.16967 * \log(T_K) - 0.01781 * S + 0.0001122 * S^2))} * 0.53$$

1.23 and 0.53 were experiment specific and measured

Initial guess at the pH

$$pH_{init} = 12.26 - 0.0030605 * DIC - 0.043752 * T - 0.013625 * S + 0.00011315 * TA + 1.3463e - 5 * DIC * T + 5.2215e - 7 * DIC * TA$$

Iterations:

For 1 to n iterations:

$$\begin{aligned}
h &= 10^{-pH_{init}} \\
h_{free} &= \frac{h}{Free_{2T}} \\
f0 &= (DIC * 1e - 6 * \frac{K1 * h + 2 * K1 * K2}{h * h + K1 * h + K1 * K2} \\
&\quad - h_{free} + \frac{KW}{h} - Alk * 1e - 6 + \frac{TB}{1 + \frac{h}{KB}}) * 1e6 \\
df0 &= (DIC * 1e - 6 * \frac{K1 + 2 * K1 * K2}{h^2 + K1 * h + K1 * K2} \\
&\quad - DIC * 1e - 6 * \frac{(K1 * h + 2 * K1 * K2)}{(h^2 + K1 * h + K1 * K2)^2} * (2 * h + K1) \\
&\quad - TB * \frac{1}{(1 + \frac{h}{KB})^2} / KB \\
&\quad - \frac{KW}{h^2} - \frac{1}{Free_{2T}}) * 1e6 * (-\log(10) * 10^{-pH}) \\
pH &= pH - \frac{f0}{df0} \\
H &= 10^{-pH} \\
denom &= H^2 + K1 * H + K1 * K2 \\
CO2 &= \frac{DIC * H^2}{denom} \\
HCO3 &= \frac{DIC * H * K1}{denom} \\
CO3 &= \frac{DIC * K1 * K2}{denom}
\end{aligned}$$

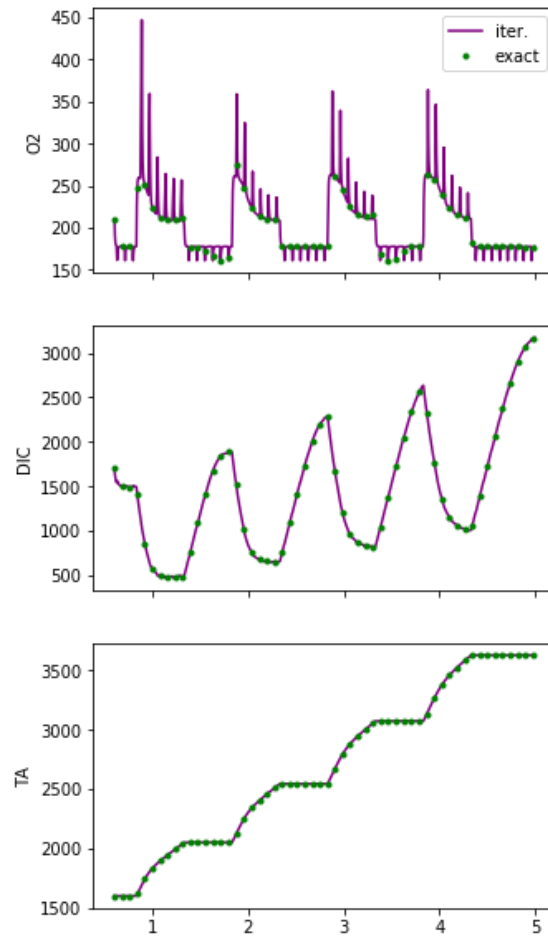


Figure 4.1 : Iterative vs exact solution for state variables  $O_2$ ,  $DIC$ , and  $TA$ .

Variable	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Iter. 5
$O_2$	0.308389964	0.016044284	4.18E-05	6.89E-05	7.59E-05
$DIC$	16.78775711	0.958511825	0.005229318	0.002305054	0.002333411
$TA$	2.607767674	0.160897272	0.000688102	0.001257725	0.001218981
$pH$	0.036092734	0.002355758	1.41E-05	6.93E-06	6.93E-06
$CO_2$	2.109401968	0.145719349	0.001222812	0.000866728	0.000866727
$HCO_3$	19.81869214	1.21021115	0.008016765	0.001025002	0.001025139
$CO_3$	20.89660704	1.307061652	0.00867642	0.001102278	0.001102434

Table 4.5 : RMSE for 5 iterations of the Newton-raphson carbon chemistry iterative solution.

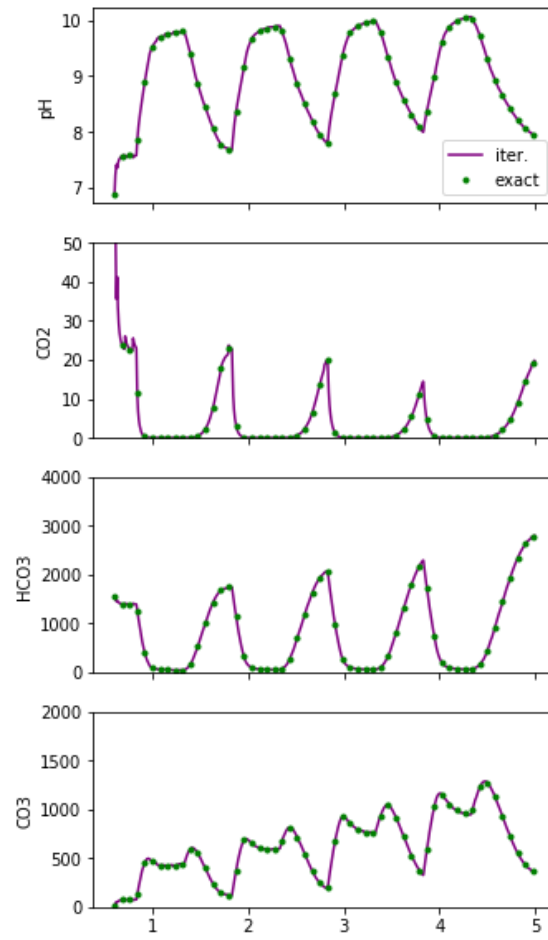


Figure 4.2 : Iterative vs exact solution for carbon chemistry  $CO_2$ ,  $HCO_3$ ,  $CO_3$ , and  $pH$ .

### 4.3.2 Posteriors

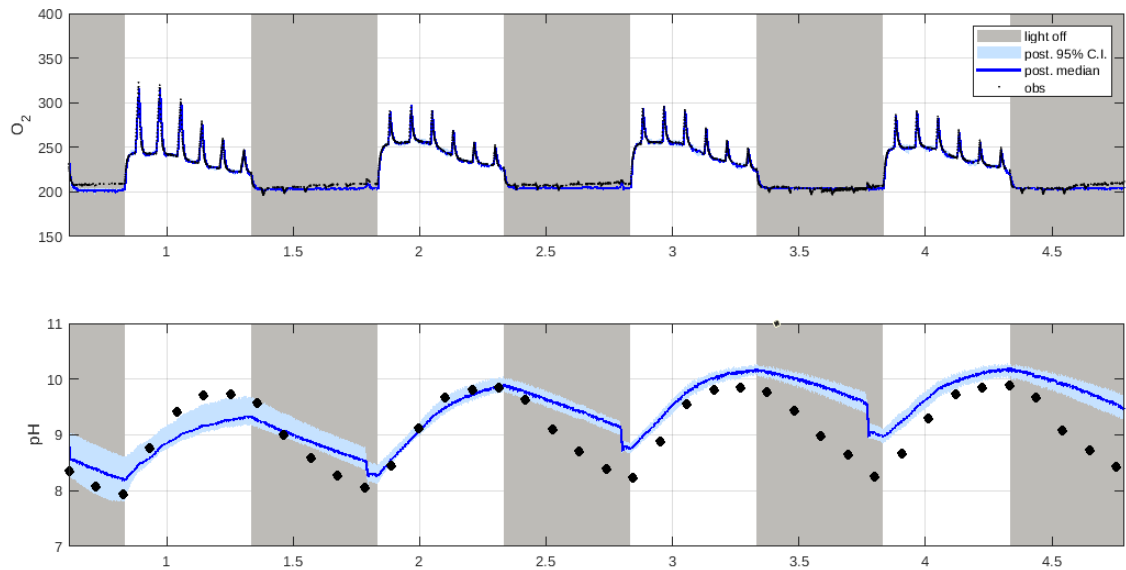


Figure 4.3 : Posteriors for  $O_2$  and  $pH$ .

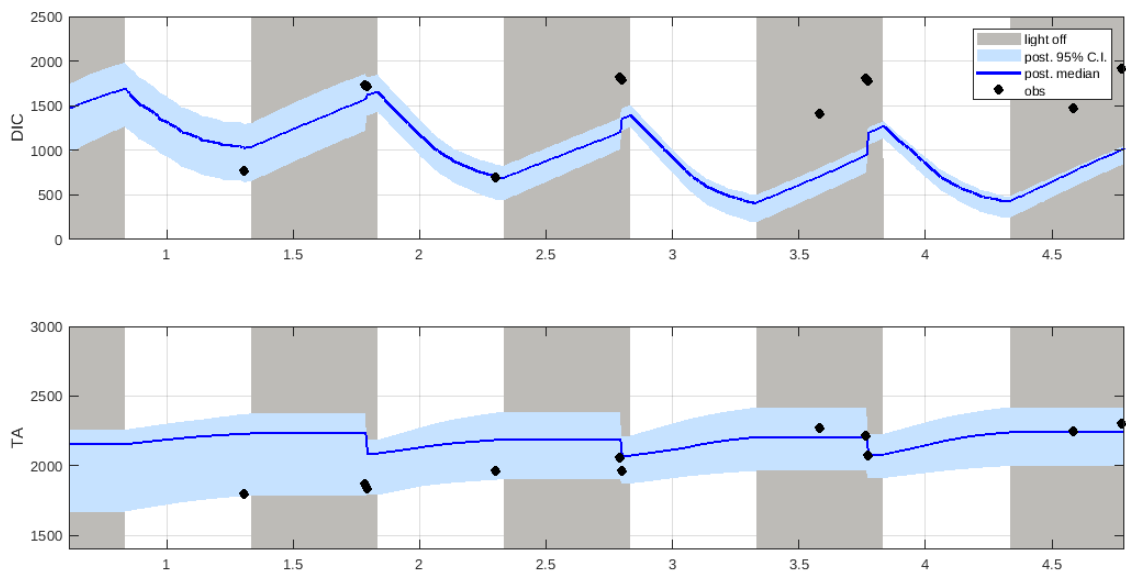


Figure 4.4 : Posteriors for  $DIC$  and  $TA$ .



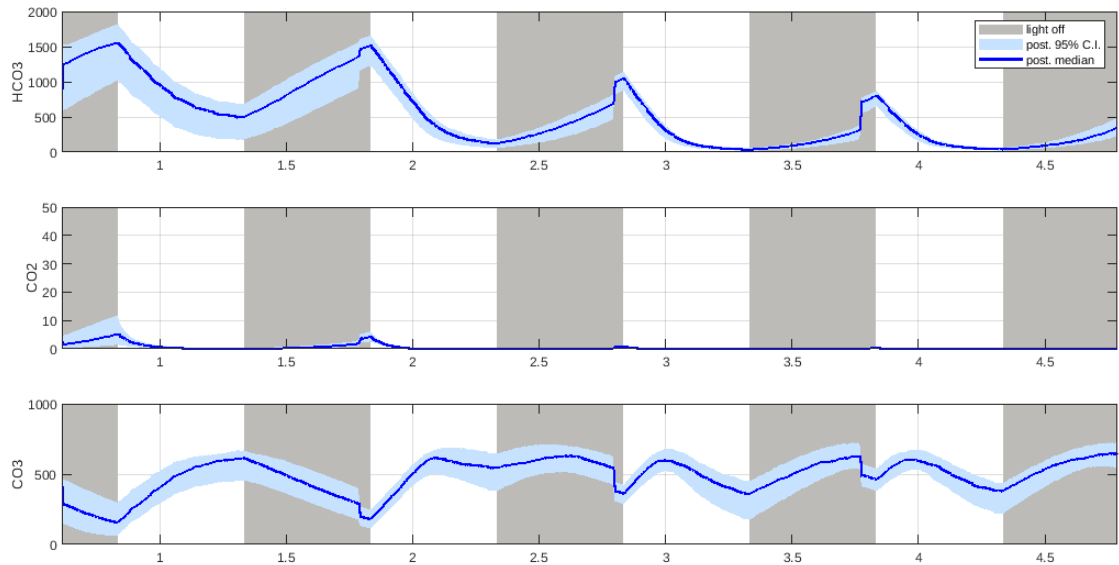


Figure 4.5 : Posteriors for carbon chem.

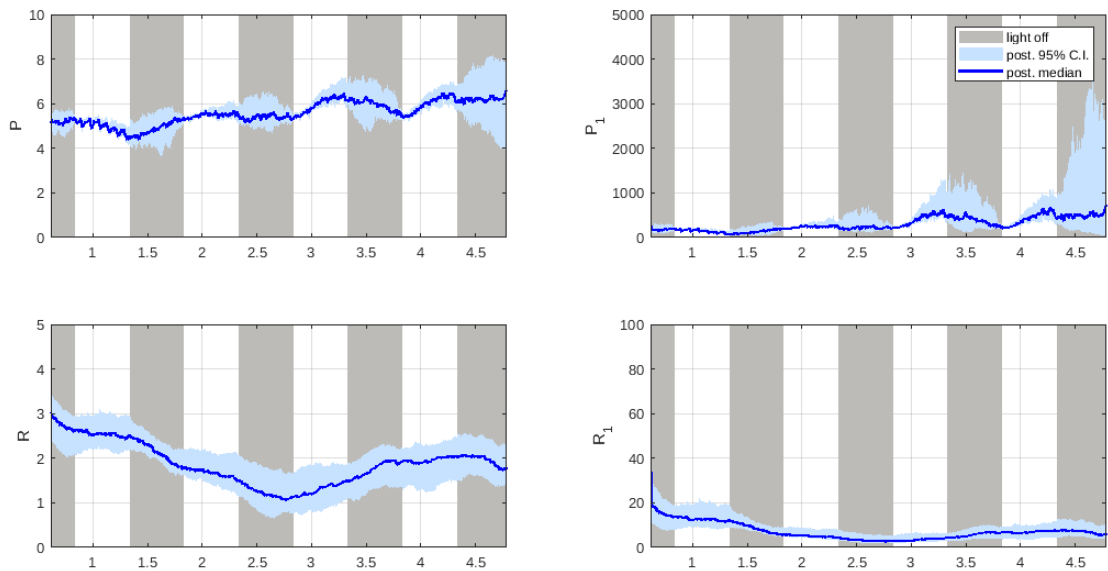


Figure 4.6 : Posteriors for photosynthesis and respiration.

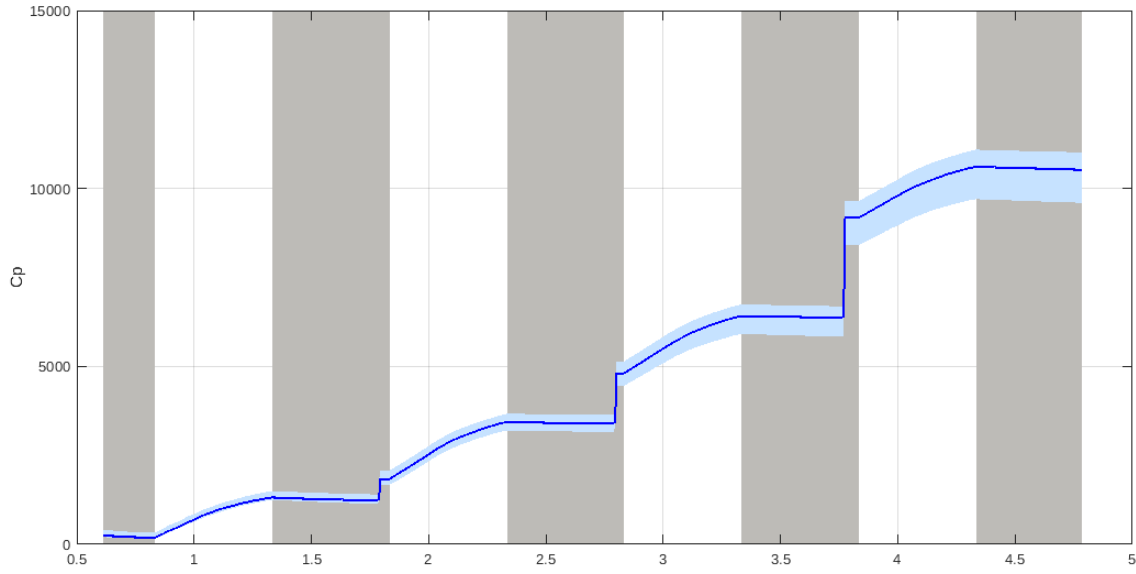
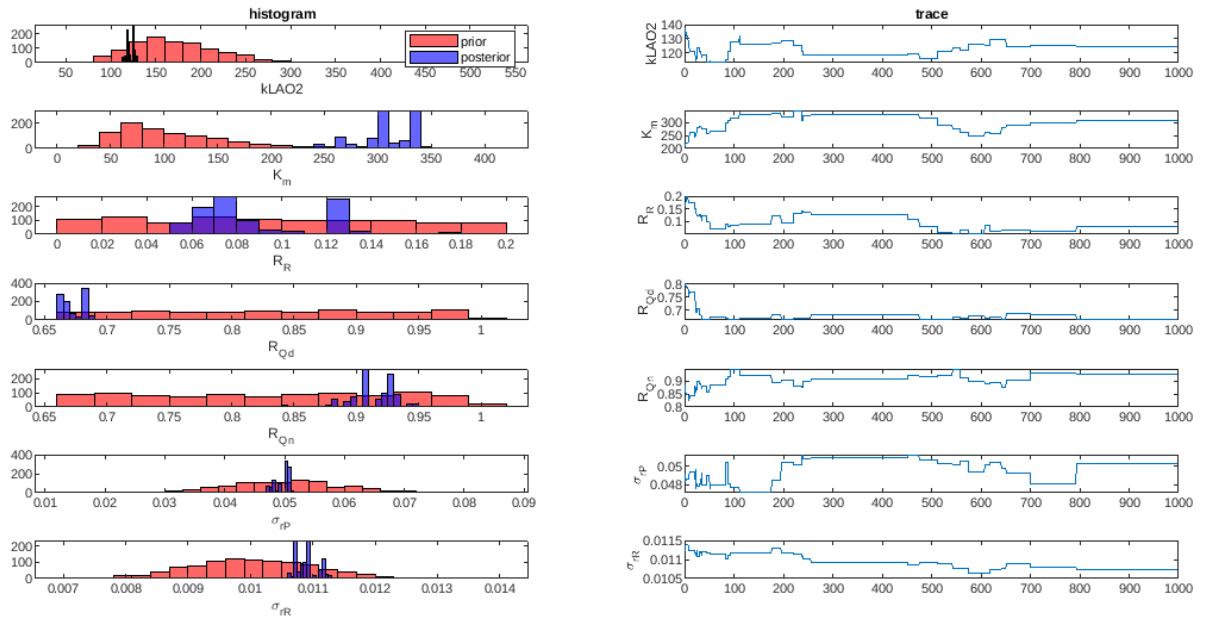
Figure 4.7 : Posterior for  $C_p$ .

Figure 4.8 : Priors, posteriors and traces for model parameters.

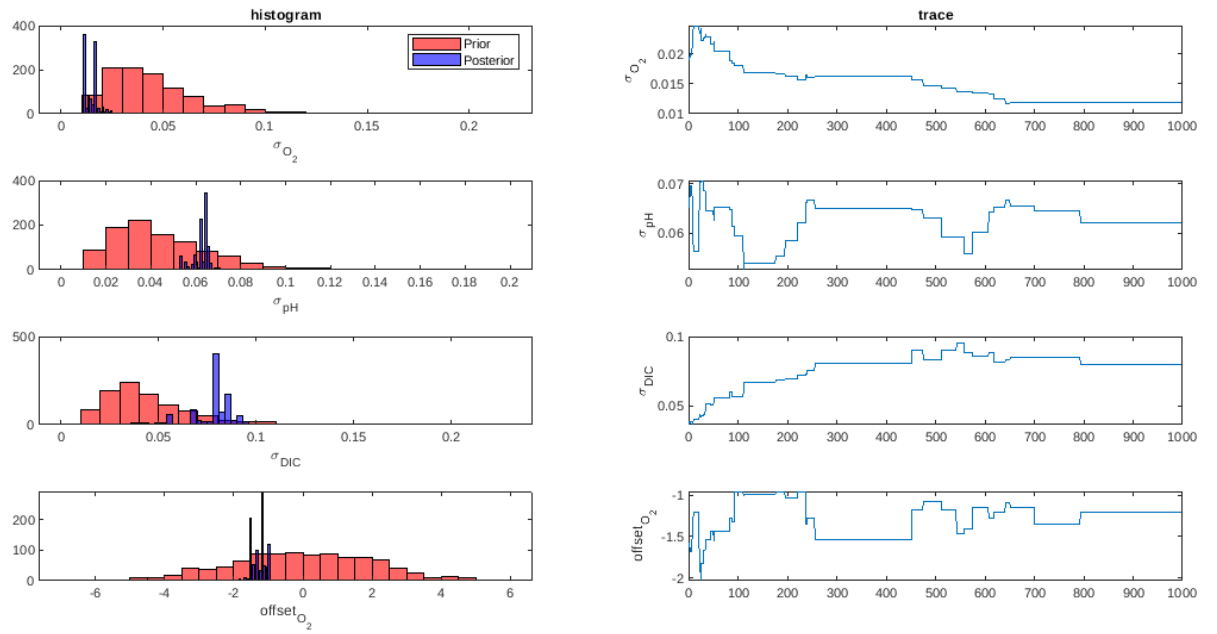


Figure 4.9 : Priors, posteriors and traces for obs. error parameters.

## Appendix A

### LiBbi model code

LiBbi model file: micro\_iterative.bi

```

1 model micro_iterative {
2
3   const F02          = 0.2094
4   const FC02         = 397e-6
5   const S            = 34.0
6   const V            = 500.0           // volume of the
      reactor
7   const DIC_M        = 1724.20         // calculated with
      CO2SYS[ DIC_M = 1724.20, Alk = 1797.90, T = 27, S = 34]
8   const O_2_M        = 226.65
9   const alk_M         = 1797.90
10  const tau           = 6.0
11  const kLA02_m       = log(2.0)*24.0*60.0/tau
12
13  param kLA02
14  param Km
15  param RR
16  param RQ_d
17  param RQ_n
18  param sigma_O_2
19  param sigma_pH
20  param sigma_DIC

```

```
21 param offset_0_2
22
23 input I           // light intensity
24 input T           // temperature (C)
25 input gas         // gas on/off
26 input dil         // dilution rate
27
28 state DIC // state variables
29 state O_2
30 state pH
31 state Cp
32 state mich_ment
33 state O2H_pr
34 state CO2H_pr
35 state R
36 state R1
37 state P
38 state P1
39 state alk
40 state CO2
41 state HCO3
42 state CO3
43 state O_2H
44 state CO2H
45 state h_3
46 state h_free_3
47
48 noise r_R
49 noise r_P
```

```

50
51 /* random walk parameter */
52 param sigma_r_R
53 param sigma_r_P
54
55 obs 02_obs
56 obs pH_obs
57 obs DIC_obs
58 obs alk_obs
59
60 sub parameter {/* prior distribution over parameters */
61 Km      ~ log_normal(log(100.0), 0.5)
62 kLA02   ~ log_normal(log(kLA02_m), 0.3)
63 RR      ~ uniform(0.0001, 0.2)
64 RQ_d    ~ uniform(0.66, 1.0)
65 RQ_n    ~ uniform(0.66, 1.0)
66
67 sigma_0_2 ~ log_normal(log(0.03), 0.5)
68 sigma_pH  ~ log_normal(log(0.03), 0.5)
69 sigma_DIC ~ log_normal(log(0.03), 0.5)
70
71 offset_0_2 ~ normal(0, 2.0)
72
73 sigma_r_R      ~ normal(0.01, 0.001)
74 sigma_r_P      ~ normal(0.05, 0.01)
75 }
76
77 const prop_std = 0.1;
78 sub proposal_parameter {

```

```

79 Km      ~ log_normal(log(Km), 0.5*prop_std)
80 kLA02    ~ log_normal(log(kLA02), 0.3*prop_std)
81 RR      ~ truncated_normal(RR, 0.2*prop_std, lower = 0.0001,
      upper = 0.2)
82 RQ_d     ~ truncated_normal(RQ_d, 0.2*prop_std, lower = 0.66,
      upper = 1.0)
83 RQ_n     ~ truncated_normal(RQ_n, 0.2*prop_std, lower = 0.66,
      upper = 1.0)
84
85
86 sigma_0_2 ~ log_normal(log(sigma_0_2), 0.5*prop_std)
87 sigma_pH  ~ log_normal(log(sigma_pH), 0.5*prop_std)
88 sigma_DIC ~ log_normal(log(sigma_DIC), 0.5*prop_std)
89
90 offset_0_2 ~ normal(offset_0_2, 2.0*prop_std)
91
92 sigma_r_R  ~ normal(sigma_r_R, 0.001*prop_std)
93 sigma_r_P  ~ normal(sigma_r_P, 0.01*prop_std)
94 }
95
96 sub initial {/* prior distribution over initial conditions,
      given parameters */
97 // specify the initial condition model
98 R      ~ normal(log(20.0), 0.4)
99 R1     ~ log_normal(log(20.0), 0.4)
100 P      ~ normal(log(200.0), 0.4)
101 P1     ~ log_normal(log(200.0), 0.4)
102
103 Cp     ~ log_normal(log(300.0), 0.2)

```

```

104 alk      ~ log_normal(log(1750.0), 0.1)
105 DIC      ~ log_normal(log(1300.0), 0.2)
106 O_2      ~ log_normal(log(225.0), 0.2)
107 pH       ~ log_normal(log(8.5), 0.2)
108 CO2      ~ log_normal(log(3.0), 0.4)
109 HCO3     ~ log_normal(log(1000.0), 0.3)
110 CO3      ~ log_normal(log(300.0), 0.4)
111 O_2H     ~ log_normal(log(200.0), 0.2)
112 CO2H     ~ log_normal(log(10.0), 0.2)
113 }
114
115
116 //sub transition(delta = 0.0023) { // obs are in days ie
      delta=1.0 for daily solving. delta=0.00069 for solving
      every minute, 0.0014 for every 2 mins, 0.0021 for 3 mins,
      0.0028 for 4mins, delta=0.000011574 for solving every
      second
117 sub transition(delta = 0.0021) {
118
119 /* processes */
120
121 inline TK      = T + 273.15           // temp in kelvin
122 inline K0_CO2 = exp(-60.2409 + 93.4517*(100.0/TK) + 23.3585*
      log(TK/100.0)+ S*(0.023517 - 0.023656*(TK/100) +
      0.0047036*(TK/100.0)*(TK/100.0)))
123 CO2H          <- K0_CO2*FCO2*1.0220*1e6
124
125 inline K0_O2   = (exp(-1282.8704 + 36619.96/TK + 223.1396*log
      (TK) -0.354707*TK + S*(5.957e-3 -3.7353/TK) + 3.68e-6*S*S)

```



```

    )/(0.2094e-06)

126 O_2H          <- K0_02*F02*1.0220*1e-6
127
128 inline PAC      = HC03              //PAC=photosynthetically
    active carbon. if the phyto are just using CO2 to
    photosynthesise then PAC=CO2
129 inline mm       = PAC/(Km + PAC)
130
131 // CO2SYS iterative solution
132 // set up all the constants
133
134 inline logTK     = log(TK)
135 inline S2        = S*S
136 inline sqrtS     = sqrt(S)
137
138 // total sulphur
139
140 inline TS        = (0.14/96.062)*(S/1.80655)
141 inline IS        = 19.924*S/(1000.0 - 1.005*S)
142
143 inline KS_int    = -4276.1/TK + 141.328 - 23.093*logTK +
    (-13856.0/TK + 324.57 - 47.986*logTK)*sqrt(IS) + (
    35474.0/TK - 771.54 + 114.723*logTK)*IS - 2698.0/TK*IS
    **1.5 + 1776.0/TK*IS**2
144 inline KS       = exp(KS_int)*(1 - 0.001005*S)
145
146 // Fluorine
147
148 inline TF        = 0.000067*S/18.9984/1.80655

```

```

149 inline KF          = exp(-(-874.0/TK - 0.111*sqrtS + 9.68))
150 inline SWS_2_T      = (1.0 + TS/KS)/(1.0 + TS/KS + TF/KF)
151 inline Free_2_T     = 1.0 + TS/KS
152
153 // H2O dissoci
154
155 inline KW = exp(148.9802 - 13847.26/TK - 23.6521*logTK +
    (118.67/TK - 5.977 + 1.0495*logTK)*sqrtS - 0.01615*S)
156
157 // Boron
158
159 inline KB = exp((-8966.90 - 2890.53*sqrtS - 77.942*S + 1.728*
    S*sqrtS - 0.0996*S2)/TK + 148.0248 + 137.1942*sqrtS +
    1.62142*S - (24.4344 + 25.085*sqrtS + 0.2474*S)*logTK +
    0.053105*sqrtS*TK)
160 inline TB = 0.0004326*S/35.0
161
162 // Carbon eq constants
163
164 inline K1 = 10**(-(3633.86/TK - 61.2172 + 9.6777 *logTK -
    0.011555*S + 0.0001152*S**2))*1.23 //1.23 experiment
    specific and measured
165 inline K2 = 10**(-( 471.8/TK + 25.9290 - 3.16967*logTK -
    0.01781*S + 0.0001122*S**2))*0.53 //0.53 experiment
    specific and measured
166
167 // end all the constants
168
169 // intial guess at the pH (use the approximating equation)

```

```

170
171 inline pH_init = 12.26 -0.0030605*DIC -0.043752*T -0.013625*S
      + 0.00011315*alk + 1.3463e-05*DIC*T + 5.2215e-07*DIC*alk
172
173 // iteration 1
174
175 inline h_1      = 10.0**(-pH_init)
176 inline h_free_1 = h_1/Free_2_T
177 inline f0_1      = (DIC*1e-6*(K1*h_1 + 2.0*K1*K2)/(h_1*h_1 +
      K1*h_1 + K1*K2) - h_free_1 + KW/h_1 - alk*1e-6 + TB/(1.0 +
      h_1/KB))*1e6
178 inline df0_1     = (DIC*1e-6*(K1 + 2.0*K1*K2)/(h_1**2.0 + K1*
      h_1 + K1*K2) - DIC*1e-6*(K1*h_1 + 2.0*K1*K2)/(h_1**2.0 +
      K1*h_1 + K1*K2)**2.0*(2.0*h_1 + K1) - TB*1.0/(1.0 + h_1/KB
      )**2.0/KB - KW/h_1**2.0 - 1.0/Free_2_T)*1e6*(-log(10.0)
      *10.0**(-pH_init))
179 inline pH_1      = pH_init - f0_1/df0_1
180
181 // iteration 2
182
183 inline h_2      = 10.0**(-pH_1)
184 inline h_free_2 = h_2/Free_2_T
185 inline f0_2      = (DIC*1e-6*(K1*h_2 + 2.0*K1*K2)/(h_2*h_2 +
      K1*h_2 + K1*K2) - h_free_2 + KW/h_2 - alk*1e-6 + TB/(1.0 +
      h_2/KB))*1e6
186 inline df0_2     = (DIC*1e-6*(K1 + 2.0*K1*K2)/(h_2**2.0 + K1*
      h_2 + K1*K2) - DIC*1e-6*(K1*h_2 + 2.0*K1*K2)/(h_2**2.0 +
      K1*h_2 + K1*K2)**2.0*(2.0*h_2 + K1) - TB*1.0/(1.0 + h_2/KB
      )**2.0/KB - KW/h_2**2.0 - 1.0/Free_2_T)*1e6*(-log(10.0)

```

```

      *10.0**(-pH_1))
187 inline pH_2      = pH_1 - f0_2/df0_2
188
189 // iteration 3
190
191 h_3                <- 10.0**(-pH_2)
192 h_free_3           <- h_3/Free_2_T
193 inline f0_3        = (DIC*1e-6*(K1*h_3 + 2.0*K1*K2)/(h_3*h_3 +
      K1*h_3 + K1*K2) - h_free_3 + KW/h_3 - alk*1e-6 + TB/(1.0 +
      h_3/KB))*1e6
194 inline df0_3       = (DIC*1e-6*(K1 + 2.0*K1*K2)/(h_3**2.0 + K1*
      h_3 + K1*K2) - DIC*1e-6*(K1*h_3 + 2.0*K1*K2)/(h_3**2.0 +
      K1*h_3 + K1*K2)**2.0*(2.0*h_3 + K1) - TB*1.0/(1.0 + h_3/KB
      )**2.0/KB - KW/h_3**2.0 - 1.0/Free_2_T)*1e6*(-log(10.0)
      *10.0**(-pH_2))
195 pH                <- pH_2 - f0_3/df0_3
196
197 // iteration 4
198
199 //      inline h_4      = 10.0**(-pH_3)
200 //      inline h_free_4 = h_4/Free_2_T
201 //      inline f0_4      = (DIC*1e-6*(K1*h_4 + 2.0*K1*K2)/(h_4
      *h_4 + K1*h_4 + K1*K2) - h_free_4 + KW/h_4 - alk*1e-6 + TB
      /(1.0 + h_4/KB))*1e6
202 //      inline df0_4     = (DIC*1e-6*(K1 + 2.0*K1*K2)/(h_4
      **2.0 + K1*h_4 + K1*K2) - DIC*1e-6*(K1*h_4 + 2.0*K1*K2)/(
      h_4**2.0 + K1*h_4 + K1*K2)**2.0*(2.0*h_4 + K1) - TB
      *1.0/(1.0 + h_4/KB)**2.0/KB - KW/h_4**2.0 - 1.0/Free_2_T)
      *1e6*(-log(10.0)*10.0**(-pH_3))

```

```

203 //      inline pH_4      = pH_3 - f0_4/df0_4
204
205 //      pH                <- pH_4
206
207 // calculate the final concentrations
208
209 inline H      = 10.0**(-pH)
210 inline H2     = H*H
211 inline denom  = (H2 + K1*H + K1*K2)
212 CO2          <- DIC*H2/denom
213 HCO3         <- DIC*H*K1/denom
214 CO3          <- DIC*K1*K2/denom
215
216 // end CO2SYS iterative solution
217
218
219 /* R and P as random walks */
220
221 r_R      ~ normal(0.0, sigma_r_R)
222 R        <- R + r_R
223 R1       <- exp(R)
224
225 r_P      ~ normal(0.0, sigma_r_P)
226 P        <- P + r_P
227 P1       <- exp(P)
228
229 ode(h = 0.1, atoler = 1.0e-6, rtoler = 1.0e-6, alg = 'RK4(3)
      '){

```

```

230 dDIC/dt = -P1*24.0*I*mm + R1*24.0
          + gas*0.893*kLA02*(C02H - C02)          + dil/V*(
          DIC_M - DIC)
231 dO_2/dt = (P1*24.0*I*mm - R1*24.0)/(RQ_d*I + RQ_n*(1.0-I))
          + gas*kLA02*(O_2H - O_2)          + dil/V*(O_2_M
          - O_2) + offset_O_2
232 dalk/dt = RR*P1*24.0*I*mm
          + dil/V*(
          alk_M - alk)
233 dCp/dt = (P1*24.0*I*mm - R1*24.0)
          + dil/V*(
          Cp)
234
235 }
236
237 mich_ment <- mm
238 O2H_pr    <- O_2H
239 C02H_pr   <- C02H
240
241 }
242
243
244 sub observation {
245
246 O2_obs ~ log_normal(log(O_2), sigma_O_2)
247 pH_obs ~ log_normal(log(pH), sigma_pH)
248 DIC_obs ~ log_normal(log(DIC), sigma_DIC)
249 alk_obs ~ log_normal(log(alk), sigma_DIC)
250 }

```

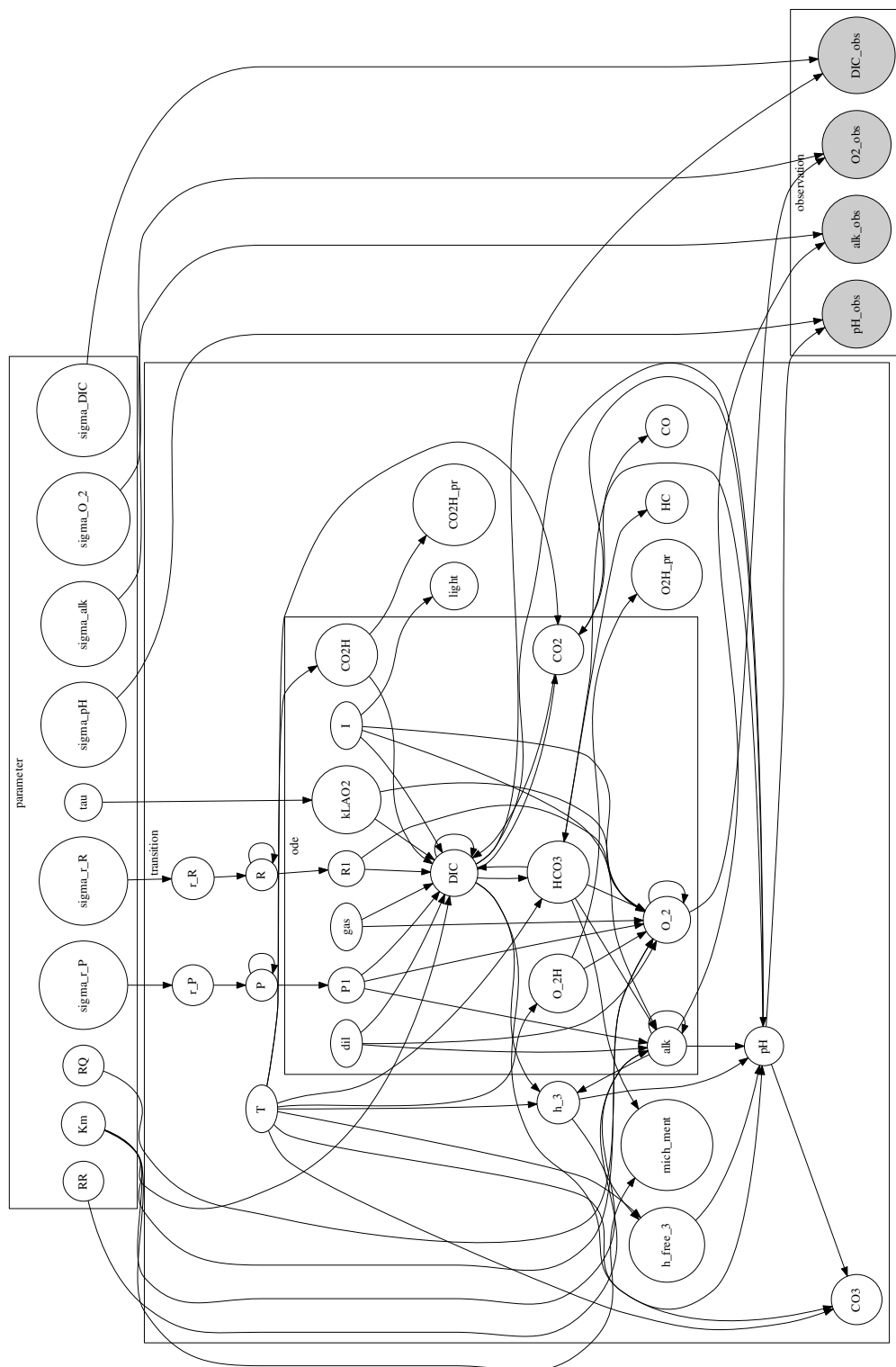
251 }

### LiBbi prior sampling file: prior.conf

```
1 --target prior
2 --model-file micro_iterative.bi
3 --nsamples 500
4 --start-time 0.61304
5 --end-time 4.7866
6 --noutputs 6049
7 --input-file data/input_all_2018_normalised.nc
8 --output-file results/prior_micro_iterative.nc
```

### LiBbi posterior sampling file: posterior.conf

```
1 --target posterior
2 --model-file micro_iterative.bi
3 --input-file data/input_all_2018_normalised.nc
4 --obs-file data/obs_all_2018.nc
5 --nsamples 500
6 --nparticles 1024
7 --start-time 0.61304
8 --end-time 4.7866
9 --noutputs 6049
10 --output-file results/posterior_micro_iterative.nc
11 --with-transform-initial-to-param
```





## Bibliography

- [1] Rubin Battino, Timothy R Rettich, and Toshihiro Tominaga. The solubility of oxygen and ozone in liquids. *Journal of physical and chemical reference data*, 12(2):163–178, 1983.
- [2] Arnold E Greensberg, LS Clesceri, Andrew D Eaton, and MAH Franson. Standard methods for the examination of water and wastewater. *American Public Health Asociation, Whashington, DC*, 1992.
- [3] E Molina Grima, JA Sánchez Pérez, F Ía Garc Camacho, and A Robles Medina. Gas-liquid transfer of atmospheric co<sub>2</sub> in microalgal cultures. *Journal of Chemical Technology & Biotechnology*, 56(4):329–337, 1993.
- [4] Ernie Lewis, Doug Wallace, and Linda J Allison. Program developed for co {sub 2} system calculations. Technical report, Brookhaven National Lab., Dept. of Applied Science, Upton, NY (United States . . . , 1998.
- [5] Niels Ramsing and Jens Gundersen. Seawater and gases. *Limnol. Oceanogr*, 37:1307–1312, 2011.
- [6] SMAC Van Heuven, D Pierrot, JWB Rae, E Lewis, and DWR Wallace. Matlab program developed for co<sub>2</sub> system calculations. *ORNL/CDIAC-105b. Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, US Department of Energy, Oak Ridge, Tennessee*, 530, 2011.
- [7] R.F Weiss. Carbon dioxide in water and seawater: the solubility of a non-ideal gas. *Marine chemistry*, 2(3):203–215, 1974.