Implementation of CLM Below-Ground Biogeochemistry in PFLOTRAN

August 20, 2013

Abstract

1 CLM-CN

1.1 Reactions

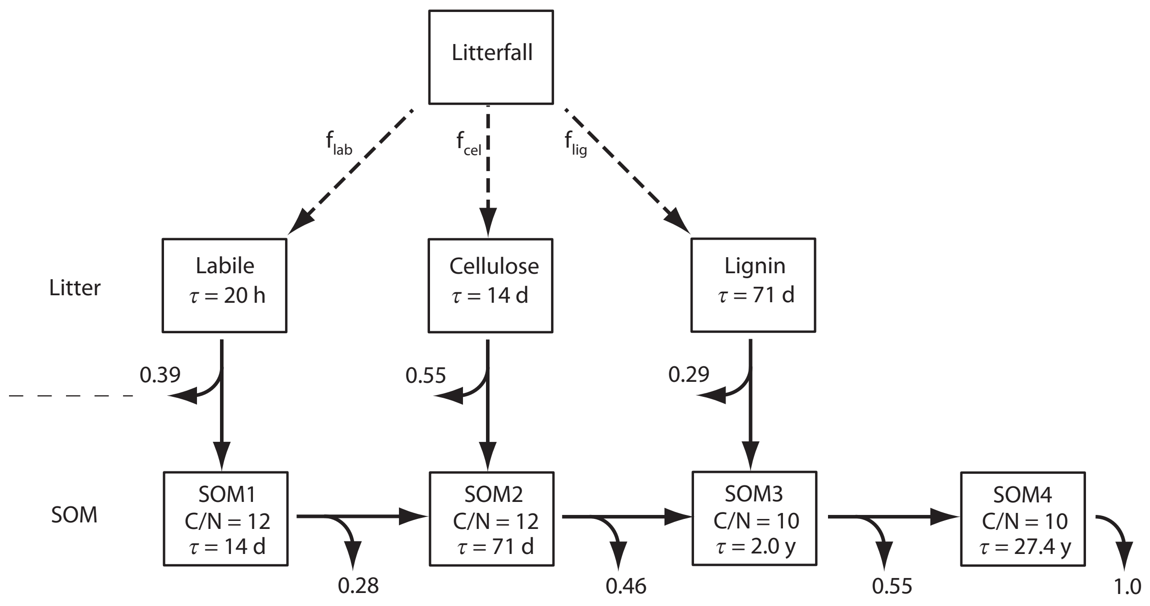

Figure 1: CLM-CN litter and soil organic pools and C and N flows Bonan et al. [2012]

1.1.1 General Reaction

The general decomposition reaction is

$$\mathrm{CN}_u = (1-f)\mathrm{CN}_d + f\mathrm{CO}_2 + n\mathrm{N}_{\mathrm{mineral}} \tag{1}$$

CN_u	=	upstream pool [mol/m^3]
CN_d	=	downstream pool [mol/m^3]
CO_2	=	[mol/m^3]
$\mathrm{N}_{\mathrm{mineral}}$	=	mineral nitrogen [mol/m^3]
u	=	molecular weight ratio of C and N divided by upstream pool C/N [-]
d	=	molecular weight ratio of C and N divided by downstream pool C/N [-]
f	=	respiration fraction [-]
n	=	$[u - (1 - f)d]$

1.1.2 Soil Organic Matter Pools

The C/N ratio is fixed in soil organic matter pools. The reactions are

Table 1: Reactions for the soil organic matter pools

SOM1	=	0.72 SOM2 + 0.28 CO_2 + 0.020000 $\mathrm{N}_{\mathrm{mineral}}$
SOM2	=	0.54 SOM3 + 0.46 CO_2 + 0.025143 $\mathrm{N}_{\mathrm{mineral}}$
SOM3	=	0.45 SOM4 + 0.55 CO_2 + 0.047143 $\mathrm{N}_{\mathrm{mineral}}$
SOM4	=	CO_2 + 0.085714 $\mathrm{N}_{\mathrm{mineral}}$

1.1.3 Litter Pools

The C/N ratio is dependent on the input from plant function groups. As the C/N ratio is generally greater in the litter pools than in the soil organic pools Adair et al. [2008], mineral N is needed to decompose the litter pools. Namely, litter decomposition involves N immobilization through microbial mass synthesis. For example, one observation indicates a C/N ratio of 31.19 for yellow birch Adair et al. [2008]. The reactions are

Table 2: Reactions for the litter pools

Lit1C + 0.027481 Lit1N + 0.016090 $\mathrm{N}_{\mathrm{mineral}}$	=	0.61 SOM1 + 0.39 CO_2
Lit2C + 0.027481 Lit2N + 0.004662 $\mathrm{N}_{\mathrm{mineral}}$	=	0.45 SOM2 + 0.55 CO_2
Lit3C + 0.027481 Lit3N + 0.033376 $\mathrm{N}_{\mathrm{mineral}}$	=	0.71 SOM3 + 0.29 CO_2

1.1.4 Summary

$$\begin{aligned}
\mathrm{Lit1C} + u_1\mathrm{Lit1N} &= (1 - f_1)\,\mathrm{SOM1} + f_1\mathrm{CO_2} + n_1\mathrm{N_{mineral}} \\
\mathrm{Lit2C} + u_2\mathrm{Lit2N} &= (1 - f_2)\,\mathrm{SOM2} + f_2\mathrm{CO_2} + n_2\mathrm{N_{mineral}} \\
\mathrm{Lit3C} + u_3\mathrm{Lit3N} &= (1 - f_3)\,\mathrm{SOM3} + f_3\mathrm{CO_2} + n_3\mathrm{N_{mineral}} \\
\mathrm{SOM1} &= (1 - f_4)\,\mathrm{SOM2} + f_4\mathrm{CO_2} + n_4\mathrm{N_{mineral}} \\
\mathrm{SOM2} &= (1 - f_5)\,\mathrm{SOM3} + f_5\mathrm{CO_2} + n_5\mathrm{N_{mineral}} \\
\mathrm{SOM3} &= (1 - f_6)\,\mathrm{SOM4} + f_6\mathrm{CO_2} + n_6\mathrm{N_{mineral}} \\
\mathrm{SOM4} &= f_7\mathrm{CO_2} + n_7\mathrm{N_{mineral}}
\end{aligned}$$

$$u_i = \mathrm{LitiN/LitiC}$$

1.2 Rate

1.2.1 General

$$R = f_T f_\Psi f_\mathrm{N} k \mathrm{CN}_u \tag{2}$$

$$\begin{aligned}
R &= \text{rate } [\mathrm{mol/(m^3 s)}] \\
f_T &= \exp\left[308.56\left(\frac{1}{71.02} - \frac{1}{T - 227.13}\right)\right] \\
f_\Psi &= \frac{\log\left(\Psi_\mathrm{min}/\Psi\right)}{\log\left(\Psi_\mathrm{min}/\Psi_\mathrm{max}\right)} \\
f_\mathrm{N} &= \frac{\mathrm{N_{mineral}}}{\mathrm{N_{mineral}} + k_{\mathrm{N_{mineral}}}}(\text{if u} < 0) \\
k &= \text{kinetic rate constant}[\mathrm{s^{-1}}] \\
T &= \text{temperature } [\mathrm{K}] \\
\Psi &= \text{soil water potential}[\mathrm{Pa}] \\
\mathrm{CN}_u &= \text{upstream carbon pool } [\mathrm{mol/m^3}] \\
\mathrm{N_{mineral}} &= \text{nitrogen concentration } [\mathrm{mol/m^3}] \\
k_\mathrm{N} &= \text{Mineral N half saturation constant } [\mathrm{mol/m^3}]
\end{aligned}$$

For the general reaction 1,

$$\begin{aligned}
\frac{\partial \mathrm{CN}_u}{\partial t} &= -R \\
\frac{\partial \mathrm{CN}_d}{\partial t} &= (1-f)\,R \\
\frac{\partial \mathrm{CO}_2}{\partial t} &= fR \\
\frac{\partial \mathrm{N}_{\mathrm{mineral}}}{\partial t} &= nR
\end{aligned}$$

1.2.2 Rates

$$\begin{aligned}
R_1 &= f_T f_\theta f_\mathrm{N} k_1 \mathrm{Lit1C} \\
R_2 &= f_T f_\theta f_\mathrm{N} k_2 \mathrm{Lit2C} \\
R_3 &= f_T f_\theta f_\mathrm{N} k_3 \mathrm{Lit3C} \\
R_4 &= f_T f_\theta f_\mathrm{N} k_4 \mathrm{SOM1} \\
R_5 &= f_T f_\theta f_\mathrm{N} k_5 \mathrm{SOM2} \\
R_6 &= f_T f_\theta f_\mathrm{N} k_6 \mathrm{SOM3} \\
R_7 &= f_T f_\theta f_\mathrm{N} k_7 \mathrm{SOM4}
\end{aligned}$$

1.2.3 Mass Conservation

$$
\begin{aligned}
\frac{\partial}{\partial t}(\mathrm{Lit1C}) &= -R_1 \\
\frac{\partial}{\partial t}(\mathrm{Lit1N}) &= -u_1 R_1 \\
\frac{\partial}{\partial t}(\mathrm{Lit2C}) &= -R_2 \\
\frac{\partial}{\partial t}(\mathrm{Lit3N}) &= -u_2 R_2 \\
\frac{\partial}{\partial t}(\mathrm{Lit3C}) &= -R_3 \\
\frac{\partial}{\partial t}(\mathrm{Lit3N}) &= -u_3 R_3 \\
\frac{\partial}{\partial t}(\mathrm{SOM1}) &= (1-f_1)R_1 - R_4 \\
\frac{\partial}{\partial t}(\mathrm{SOM2}) &= (1-f_2)R_2 + (1-f_4)R_4 - R_5 \\
\frac{\partial}{\partial t}(\mathrm{SOM3}) &= (1-f_3)R_3 + (1-f_5)R_5 - R_6 \\
\frac{\partial}{\partial t}(\mathrm{SOM4}) &= (1-f_6)R_6 - R_7 \\
\frac{\partial}{\partial t}(\mathrm{CO_2}) &= f_1R_1 + f_2R_2 + f_3R_3 + f_4R_4 + f_5R_5 + f_6R_6 + f_7R_7 \\
\frac{\partial}{\partial t}(\mathrm{N_{mineral}}) &= n_1R_1 + n_2R_2 + n_3R_3 + n_4R_4 + n_5R_5 + n_6R_6 + n_7R_7
\end{aligned}
$$

1.3 Implementation in PFLOTRAN

1.3.1 Numerical Methods

Applying finite-volume spatial discretization:

$$\int \frac{\partial x}{\partial t} dV = \int -\sum R_j dV \tag{3}$$

$$\frac{\partial x}{\partial t}\Delta V = -\sum R_j \Delta V \tag{4}$$

Implicit time discretization:

$$\frac{\Delta V}{\Delta t}\left(x^{k+1} - x^k\right) = -\sum R_j^{k+1}\Delta V \tag{5}$$

Residual:

$$\mathcal{R} = \frac{\Delta V}{\Delta t}\left(x^{k+1} - x^k\right) + \sum R_i^{k+1}\Delta V \tag{6}$$

Jacobian:

$$\mathcal{J} = \frac{\partial \mathcal{R}}{\partial x} \tag{7}$$

Newton-Raphson Method:

$$\mathcal{J}\delta x = -\mathcal{R} \tag{8}$$

$$x^{k+1,i+1} = x^{k+1,i} + \delta x \tag{9}$$

Table 3: Units for residuals and Jacobian

	aqueous species	immobile species	mixed
x	mol/L	mol/m^3	
ΔV	L	m^3	L
R	mol/Ls	mol/m^{3}s	mol/Ls
$\mathcal{R}$	mol/s	mol/s	mol/s
$\mathcal{J}$	L/s	m^3/s	

1.3.2 Implementation

The source code reaction_sandbox_clm_cn.F90 implements CLM-CN with input file like the following:

```
CHEMISTRY
  ...
  IMMOBILE_SPECIES
    N
    C
    SOM1
    SOM2
    SOM3
    SOM4
    LabileC
    CelluloseC
    LigninC
    LabileN
    CelluloseN
    LigninN
  /
  ...
  REACTION_SANDBOX
    CLM-CN
      POOLS   ! CN ratio
        SOM1  12.d0
        SOM2  12.d0
        SOM3  10.d0
        SOM4  10.d0
        Labile
        Cellulose
        Lignin
      /
      REACTION
        UPSTREAM_POOL Labile
        DOWNSTREAM_POOL SOM1
        TURNOVER_TIME 20. h
        RESPIRATION_FRACTION 0.39d0
        N_INHIBITION 1.d-10
      /
      REACTION
        UPSTREAM_POOL Cellulose
        DOWNSTREAM_POOL SOM2
        TURNOVER_TIME 14. d
        RESPIRATION_FRACTION 0.55
        N_INHIBITION 1.d-10
      /
      REACTION
        UPSTREAM_POOL Lignin
        DOWNSTREAM_POOL SOM3
        TURNOVER_TIME 71. d
        RESPIRATION_FRACTION 0.29d0
        N_INHIBITION 1.d-10
      /
      REACTION
        UPSTREAM_POOL SOM1
        DOWNSTREAM_POOL SOM2
        TURNOVER_TIME 14. d
        RESPIRATION_FRACTION 0.28d0
        N_INHIBITION 1.d-10
      /
      REACTION
        UPSTREAM_POOL SOM2
        DOWNSTREAM_POOL SOM3
        TURNOVER_TIME 71. d
        RESPIRATION_FRACTION 0.46d0
        N_INHIBITION 1.d-10
      /
      REACTION
        UPSTREAM_POOL SOM3
        DOWNSTREAM_POOL SOM4
        TURNOVER_TIME 2. y
        RESPIRATION_FRACTION 0.55d0
        N_INHIBITION 1.d-10
      /
      REACTION
```

```
        UPSTREAM_POOL SOM4
        TURNOVER_TIME 27.4 y
        RESPIRATION_FRACTION 1.d0
        N_INHIBITION 1.d-10
      /
    /
  /
```

In the source code, the key is to specify the residual and Jacobian. The residuals are:

$$
\begin{aligned}
\mathcal{R}_{\text{Lit1C}} &= \frac{\Delta V}{\Delta t}\left(\text{Lit1C}^{k+1} - \text{Lit1C}^{k}\right) + R_1^{k+1}\Delta V \\
\mathcal{R}_{\text{Lit1N}} &= \frac{\Delta V}{\Delta t}\left(\text{Lit1N}^{k+1} - \text{Lit1N}^{k}\right) + u_1 R_1^{k+1}\Delta V \\
\mathcal{R}_{\text{Lit2C}} &= \frac{\Delta V}{\Delta t}\left(\text{Lit2C}^{k+1} - \text{Lit2C}^{k}\right) + R_2^{k+1}\Delta V \\
\mathcal{R}_{\text{Lit2N}} &= \frac{\Delta V}{\Delta t}\left(\text{Lit2N}^{k+1} - \text{Lit2N}^{k}\right) + u_2 R_2^{k+1}\Delta V \\
\mathcal{R}_{\text{Lit3C}} &= \frac{\Delta V}{\Delta t}\left(\text{Lit3C}^{k+1} - \text{Lit3C}^{k}\right) + R_3^{k+1}\Delta V \\
\mathcal{R}_{\text{Lit3N}} &= \frac{\Delta V}{\Delta t}\left(\text{Lit3N}^{k+1} - \text{Lit3N}^{k}\right) + u_3 R_3^{k+1}\Delta V \\
\mathcal{R}_{\text{SOM1}} &= \frac{\Delta V}{\Delta t}\left(\text{SOM1}^{k+1} - \text{SOM1}^{k}\right) - \left[(1-f_1)R_1^{k+1} - R_4^{k+1}\right]\Delta V \\
\mathcal{R}_{\text{SOM2}} &= \frac{\Delta V}{\Delta t}\left(\text{SOM2}^{k+1} - \text{SOM2}^{k}\right) - \left[(1-f_2)R_2^{k+1} + (1-f_4)R_4^{k+1} - R_5^{k+1}\right]\Delta V \\
\mathcal{R}_{\text{SOM3}} &= \frac{\Delta V}{\Delta t}\left(\text{SOM3}^{k+1} - \text{SOM3}^{k}\right) - \left[(1-f_3)R_3^{k+1} + (1-f_5)R_5^{k+1} - R_6^{k+1}\right]\Delta V \\
\mathcal{R}_{\text{SOM4}} &= \frac{\Delta V}{\Delta t}\left(\text{SOM4}^{k+1} - \text{SOM4}^{k}\right) - \left[(1-f_6)R_6^{k+1} - R_7^{k+1}\right]\Delta V \\
\mathcal{R}_{\text{CO}_2} &= \frac{\Delta V}{\Delta t}\left(\text{CO}_2^{k+1} - \text{CO}_2^{k}\right) \\
&\quad - \left[f_1 R_1^{k+1} + f_2 R_2^{k+1} + f_3 R_3^{k+1} + f_4 R_4^{k+1} + f_5 R_5^{k+1} + f_6 R_6^{k+1} + R_7^{k+1}\right]\Delta V \\
\mathcal{R}_{\text{N}_{\text{mineral}}} &= \frac{\Delta V}{\Delta t}\left(\text{N}_{\text{mineral}}^{k+1} - \text{N}_{\text{mineral}}^{k}\right) \\
&\quad - \left[n_1 R_1^{k+1} + n_2 R_2^{k+1} + n_3 R_3^{k+1} + n_4 R_4^{k+1} + n_5 R_5^{k+1} + n_6 R_6^{k+1} + R_7^{k+1}\right]\Delta V
\end{aligned}
$$

For Lit1 decomposition, the rate is

$$R_1 = f_T f_\Psi f_{\text{N}} k_1 \text{Lit1C} \tag{10}$$

the derivatives are:

$$\frac{\partial R_1}{\partial \text{Lit1C}} = f_T f_\Psi f_{\text{N}} k_1 = R'_{1,\text{Lit1C}} \tag{11}$$

$$\frac{\partial R_1}{\partial \text{N}_{\text{mineral}}} = f_T f_\Psi k_1 \text{Lit1C} \frac{k_N}{(k_N + \text{N}_{\text{mineral}})^2} = R'_{1,\text{N}} \tag{12}$$

$$\begin{aligned}
\frac{\partial R_{\mathrm{Lit1N}}}{\partial \mathrm{Lit1C}} &= \frac{\partial(u_1R_1)}{\partial \mathrm{Lit1C}} &&= R_1\frac{\partial u_1}{\partial \mathrm{Lit1C}} + u_1R_{1,\mathrm{Lit1C}} &&= -R_1\frac{\mathrm{Lit1N}}{\mathrm{Lit1C}^2} + u_1R'_{1,\mathrm{Lit1C}}\\
\frac{\partial R_{\mathrm{Lit1N}}}{\partial \mathrm{Lit1N}} &= \frac{\partial(u_1R_1)}{\partial \mathrm{Lit1N}} &&= R_1\frac{\partial u_1}{\partial \mathrm{Lit1N}} &&= R_1\frac{1}{\mathrm{Lit1C}}\\
\frac{\partial R_{\mathrm{Lit1N}}}{\partial \mathrm{Nmineral}} &= \frac{\partial(u_1R_1)}{\partial \mathrm{Nmineral}} &&= R_1\frac{\partial u_1}{\partial \mathrm{Nmineral}} + u_1R'_{1,\mathrm{N}} &&= u_1R'_{1,\mathrm{N}}\\
\frac{\partial R_{\mathrm{Nmineral}}}{\partial \mathrm{Lit1C}} &= -\frac{\partial(n_1R_1)}{\partial \mathrm{Lit1C}} &&= -R_1\frac{\partial n_1}{\partial \mathrm{Lit1C}} - n_1R'_{1,\mathrm{Lit1C}} &&= R_1\frac{\mathrm{Lit1N}}{\mathrm{Lit1C}^2} - n_1R'_{1,\mathrm{Lit1C}}\\
\frac{\partial R_{\mathrm{Nmineral}}}{\partial \mathrm{Lit1N}} &= -\frac{\partial(n_1R_1)}{\partial \mathrm{Lit1N}} &&= -R_1\frac{\partial n_1}{\partial \mathrm{Lit1N}} &&= -R_1\frac{1}{\mathrm{Lit1C}}\\
\frac{\partial R_{\mathrm{Nmineral}}}{\partial \mathrm{N_{mineral}}} &= -\frac{\partial(n_1R_1)}{\partial \mathrm{N_{mineral}}} &&= -R_1\frac{\partial n_1}{\partial \mathrm{N_{mineral}}} &&= -n_1R'_{1,N}
\end{aligned}$$

Table 4: Jacobian for Litter Pools

	LitiC	LitiN	SOMi	CO_2	N_{mineral}
LitiC	$R'_{\mathrm{i,LitiC}}$	0	0	0	$R'_{i,N}$
LitiN	$-R_i\frac{\mathrm{LitiN}}{\mathrm{LitiC}^2} + u_iR'_{\mathrm{i,LitiC}}$	$R_i\frac{1}{\mathrm{LitiC}}$	0	0	$u_iR'_{i,N}$
SOMi	$-(1-f_i)R'_{\mathrm{i,LitiC}}$	0	0	0	$-(1-f_i)R'_{i,N}$
CO_2	$-f_iR'_{\mathrm{i,LitiC}}$	0	0	0	$-f_iR'_{i,N}$
N_{mineral}	$R_i\frac{\mathrm{LitiN}}{\mathrm{LitiC}^2} - n_iR'_{\mathrm{i,LitiC}}$	$-R_1\frac{1}{\mathrm{LitiC}}$	0	0	$-n_iR'_{i,N}$

Table 5: Jacobian for SOM Pools

	SOM1	SOM2	SOM3	SOM4	CO_2	N_{mineral}
SOM1	R'_4	0	0	0	0	0
SOM2	$-(1-f_4)R'_4$	R'_5	0	0	0	0
SOM3	0	$-(1-f_5)R'_5$	R'_6	0	0	0
SOM4	0	0	$-(1-f_6)R'_6$	R'_7	0	0
CO_2	$-f_4R'_4$	$-f_5R'_5$	$-f_6R'_6$	$-R'_7$	0	0
N_{mineral}	$-n_4R'_4$	$-n_5R'_5$	$-n_6R'_6$	$-n_7R'_7$	0	0

1.4 Applications

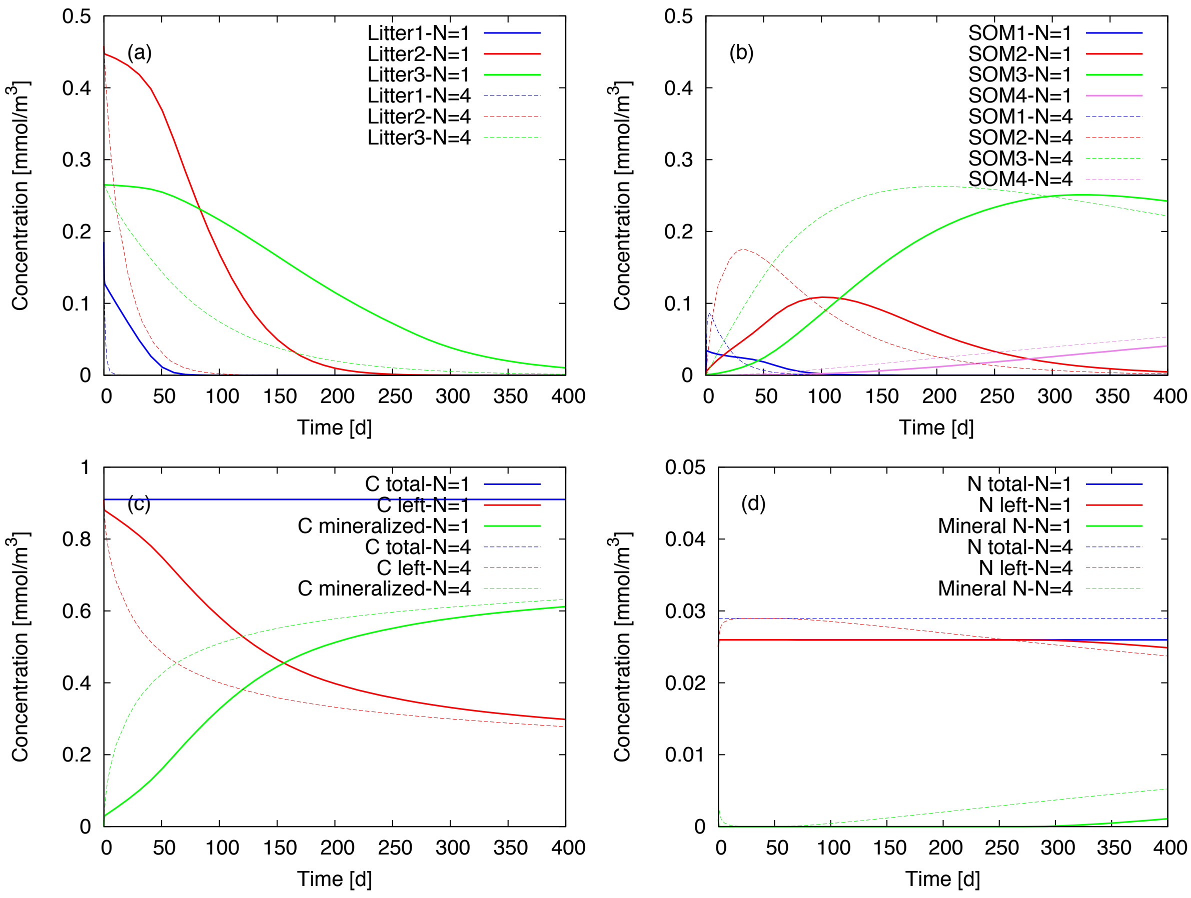

Figure 2: Demonstrating N limiting on C decomposition (initial mineral N=1 and 4 μ mol/m^3)

2 CLM4.5 CH_4 Oxidation

2.1 Reaction

$$\mathrm{CH_4 + 2O_2 = CO_2 + 2H_2O} \tag{13}$$

2.2 Rate

$$R = k\frac{\mathrm{CH_4}}{k_{\mathrm{CH4}} + \mathrm{CH_4}}\frac{\mathrm{O_2}}{k_{\mathrm{O2}} + \mathrm{O_2}} f_T f_\Psi \tag{14}$$

2.3 Residuals

$$\begin{aligned}
\mathcal{R}_{\mathrm{CH4}} &= \frac{\Delta V}{\Delta t}\left(\mathrm{CH}_4^{k+1} - \mathrm{CH}_4^{k}\right) + R^{k+1}\Delta V \\
\mathcal{R}_{\mathrm{O2}} &= \frac{\Delta V}{\Delta t}\left(\mathrm{O}_2^{k+1} - \mathrm{O}_2^{k}\right) + 2R^{k+1}\Delta V \\
\mathcal{R}_{\mathrm{CO2}} &= \frac{\Delta V}{\Delta t}\left(\mathrm{CO}_2^{k+1} - \mathrm{CO}_2^{k}\right) - R^{k+1}\Delta V
\end{aligned}$$

2.4 Jacobian

$$\begin{aligned}
\frac{\partial R}{\partial \mathrm{CH_4}} &= k\frac{k_{\mathrm{CH4}}}{(k_{\mathrm{CH4}} + \mathrm{CH_4})^2}\frac{\mathrm{O_2}}{k_{\mathrm{O2}} + \mathrm{O_2}} f_T f_\Psi = R'_{\mathrm{CH4}} \\
\frac{\partial R}{\partial \mathrm{O_2}} &= k\frac{\mathrm{CH_4}}{k_{\mathrm{CH4}} + \mathrm{CH_4}}\frac{k_{\mathrm{O2}}}{(k_{\mathrm{O4}} + \mathrm{O_2})^2} f_T f_\Psi = R'_{\mathrm{O2}}
\end{aligned}$$

Table 6: Jacobian for methane oxidation

	CH_4	O_2	CO_2
CH_4	R'_{CH4}	R'_{O2}	0
O_2	$2R'_{\mathrm{CH4}}$	$2R'_{\mathrm{O2}}$	0
CO_2	$-R'_{\mathrm{CH4}}$	$-R'_{\mathrm{O2}}$	0

2.5 Application

Input file

```
CHEMISTRY
  PRIMARY_SPECIES
    O2(aq)
    Methane(aq)
    CO2(aq)
  /
  REDOX_SPECIES
    CO2(aq)
    Methane(aq)
    O2(aq)
  /
  REACTION_SANDBOX
    CH4O
      RATE_CONSTANT 1.25d-10 ! mol/m3 s
      HALFSATURATIONCH4 5.0d-6
      HALFSATURATIONO2  2.0d-5
    /
  /
  DATABASE ../../pflotran-clm4me/database/hanford.dati
......
CONSTRAINT initial
  CONCENTRATIONS
    O2(aq)      0.001 T
    Methane(aq) 0.001 T
    CO2(aq)   1.0d-10 T
  /
END
```

Code

```
subroutine CH4OReact(this,Residual,Jacobian,compute_derivative, &
                         rt_auxvar,global_auxvar,porosity,volume,reaction, &
                         option)
  word = "Methane(aq)"
  is_ch4 = GetPrimarySpeciesIDFromName(word,reaction,option)

  word = "CO2(aq)"
  is_co2 = GetPrimarySpeciesIDFromName(word,reaction,option)

  word = "O2(aq)"
  is_o2 = GetPrimarySpeciesIDFromName(word,reaction,option)

  temp_K = global_auxvar%temp(1) + 273.15d0
  F_t = exp(308.56d0*(one_over_71_02 - 1.d0/(temp_K - 227.13d0)))

  F_theta = log(theta_min/global_auxvar%sat(1)) * one_over_log_theta_min

  L_water = porosity*global_auxvar%sat(iphase)*volume*1.d3
  c_ch4 = rt_auxvar%total(is_ch4,iphase)
  c_o2 = rt_auxvar%total(is_o2,iphase)

  rate = this%rate_constant * L_water * &  ! mole/(L sec)
```

```
    c_ch4/(this%kmch4 + c_ch4) * c_o2/(this%kmo2 + c_o2) * F_t * F_theta

  Residual(is_ch4) = Residual(is_ch4) + rate
  Residual(is_o2) = Residual(is_o2) + 2.0 * rate
  Residual(is_co2) = Residual(is_co2) - rate

  if (compute_derivative) then

    ! always add contribution to Jacobian
    ! units = (mol/sec)*(kg water/mol) = kg water/sec

    !dx/(k+x) = k/(k+x)^2

    drate_dch4 = rate * this%kmch4 / c_ch4 / (this%kmch4 + c_ch4)
    drate_do2  = rate * this%kmo2 / c_o2 / (this%kmo2 + c_o2)

    Jacobian(is_ch4,is_ch4) = Jacobian(is_ch4,is_ch4) - drate_dch4
    Jacobian(is_ch4,is_o2) = Jacobian(is_ch4,is_o2) - drate_do2
    Jacobian(is_o2,is_ch4) = Jacobian(is_o2,is_ch4) - 2.0 * drate_dch4
    Jacobian(is_o2,is_o2) = Jacobian(is_o2,is_o2) - 2.0 * drate_do2
    Jacobian(is_co2,is_ch4) = Jacobian(is_co2,is_ch4) + drate_dch4
    Jacobian(is_co2,is_o2) = Jacobian(is_co2,is_o2) + drate_do2

  endif

end subroutine CH4OReact
```

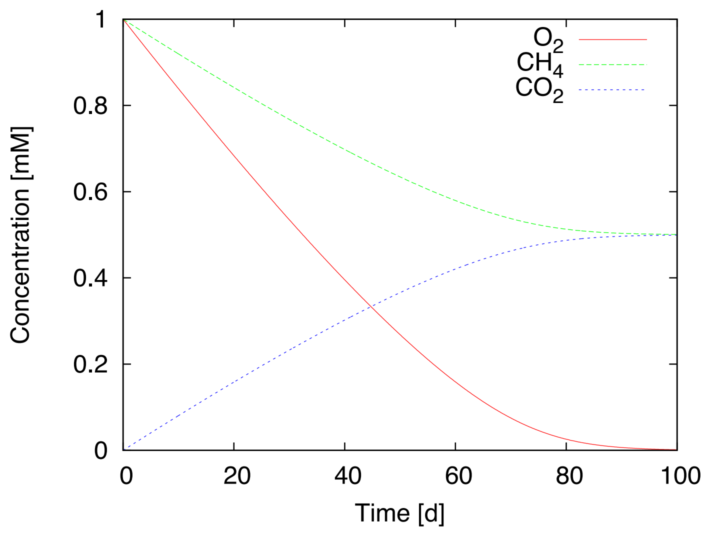

Figure 3: Example calculation for methane oxidation)

3 Acetoclastic Methanogenesis

3.1 Reaction

$$\mathrm{Ac^- + H_2O = CH_4 + HCO_3^-} \tag{15}$$

$$(1 + y/2)\mathrm{Ac^-} + 0.5y\mathrm{H^+} + \mathrm{H_2O} = \mathrm{CH_4} + \mathrm{HCO_3^-} + y\mathrm{C_{bio}} \tag{16}$$

3.2 Rate

$$R = k\mathrm{C_{bio}}\frac{\mathrm{Ac^-}}{k_{\mathrm{Ac}} + \mathrm{Ac^-}} f_T f_\Psi \tag{17}$$

3.3 Mass Conservation

$$\begin{aligned}
\frac{\partial \mathrm{Ac^-}}{\partial t} &= -(1 + \frac{y}{2})R \\
\frac{\partial \mathrm{H^+}}{\partial t} &= -\frac{y}{2}R \\
\frac{\partial \mathrm{CH_4}}{\partial t} &= R \\
\frac{\partial \mathrm{HCO_3^-}}{\partial t} &= R \\
\frac{\partial \mathrm{C_{bio}}}{\partial t} &= 1000\theta y R
\end{aligned}$$

Note: for the last equation, PFLOTRAN accounts for the 1000θ internally.

3.4 Residuals

$$\begin{aligned}
\mathcal{R}_{\text{Ac-}} &= \frac{\Delta V}{\Delta t}\left(\mathrm{Ac}^{-k+1} - \mathrm{Ac}^{-k}\right) + (1 + y/2)R^{k+1}\Delta V \\
\mathcal{R}_{\text{CH4}} &= \frac{\Delta V}{\Delta t}\left(\mathrm{CH}_4^{k+1} - \mathrm{CH}_4^{k}\right) - R^{k+1}\Delta V \\
\mathcal{R}_{\text{Cbio}} &= \frac{\Delta V}{\Delta t}\left(\mathrm{C}_{\text{bio}}^{k+1} - \mathrm{C}_{\text{bio}}^{k}\right) - yR^{k+1}\Delta V \\
\mathcal{R}_{\text{HCO3-}} &= \frac{\Delta V}{\Delta t}\left(\mathrm{HCO}_3^{-k+1} - \mathrm{HCO}_3^{-k}\right) - R^{k+1}\Delta V \\
\mathcal{R}_{\text{H+}} &= \frac{\Delta V}{\Delta t}\left(\mathrm{H}^{+k+1} - \mathrm{H}^{+k}\right) + y/2R^{k+1}\Delta V
\end{aligned}$$

3.5 Jacobian

$$\frac{\partial R}{\partial \mathrm{Ac}} = k\mathrm{C}_{\mathrm{bio}}\frac{k_{\mathrm{Ac}}}{(k_{\mathrm{Ac}}+\mathrm{Ac}^-)^2}f_T f_\Psi = R'_a$$
$$\frac{\partial R}{\partial \mathrm{C}_{\mathrm{bio}}} = k\frac{\mathrm{Ac}^-}{k_{\mathrm{Ac}}+\mathrm{Ac}^-}f_T f_\Psi = R'_b$$

Table 7: Jacobian for methane oxidation

	Ac^-	CH_4	$\mathrm{C}_{\mathrm{bio}}$	HCO_3^-	H^+
Ac^-	$(1+y/2)R'_a$	0	$(1+y/2)R'_b$	0	0
CH_4	$-R'_a$	0	$-R'_b$	0	0
$\mathrm{C}_{\mathrm{bio}}$	$-yR'_a$	0	$-yR'_b$	0	0
HCO_3^-	$-R'_a$	0	$-R'_b$	0	0
H^+	$0.5yR'_a$	0	$0.5yR'_b$	0	0

3.6 Application

Input

```
CHEMISTRY
  PRIMARY_SPECIES
    Acetate-
    Methane(aq)
    H+
    HCO3-
  /
  SECONDARY_SPECIES
    OH-
    CO3--
    CO2(aq)
:     Acetic_acid(aq)
  /
  REDOX_SPECIES
    Acetate-
    Methane(aq)
  /
  IMMOBILE_SPECIES
    Acemeg
  /
  REACTION_SANDBOX
     AceMeg
      RATE_CONSTANT      1.0d-6
      HALFSATURATIONAC   1.0d-5
      YIELDCOEFFICIENT   0.02
     /
  /
  DATABASE ../../pflotran-clm4me/database/hanford.dat
/end{verbatim}

\noindent Code
\begin{verbatim}
  L_water = porosity*global_auxvar%sat(iphase)*volume*1.d3
  c_ac = rt_auxvar%pri_molal(this%is_ac)
  c_bio = rt_auxvar%immobile(this%ispec_id_cbio)

  rate = this%rate_constant * L_water * &
    c_bio * c_ac/(this%kmac + c_ac) * F_t * F_theta

  ! alway subtract contribution from residual (mole/sec)
  Residual(this%is_ac) = Residual(this%is_ac) + (1.0 + 0.5 * this%yield) * rate
  Residual(this%is_ch4) = Residual(this%is_ch4) - rate
  Residual(this%is_cbio) = Residual(this%is_cbio) - this%yield * rate
  Residual(this%is_hco3) = Residual(this%is_hco3) - rate
  Residual(this%is_h) = Residual(this%is_h) + 0.5 * this%yield * rate

  if (compute_derivative) then

! 11. If using an analytical Jacobian, add code for Jacobian evaluation

    ! always add contribution to Jacobian
    ! units = (mol/sec)*(kg water/mol) = kg water/sec

    drate_dac = rate * this%kmac / c_ac / (this%kmac + c_ac)
    drate_dcb = rate / c_bio

    Jacobian(this%is_ac,  this%is_ac)   = Jacobian(this%is_ac,this%is_ac)   &
                                        - (1.0 + 0.5 * this%yield) * drate_dac
    Jacobian(this%is_ch4, this%is_ac)   = Jacobian(this%is_ch4,this%is_ac)  &
                                        + drate_dac
    Jacobian(this%is_cbio,this%is_ac)   = Jacobian(this%is_cbio,this%is_ac) &
                                        + this%yield * drate_dac
    Jacobian(this%is_hco3,this%is_ac)   = Jacobian(this%is_hco3,this%is_ac) &
                                        + 0.5 * this%yield * drate_dac
    Jacobian(this%is_ac,  this%is_cbio) = Jacobian(this%is_ac,this%is_cbio) &
                                        - (1.0 + 0.5 * this%yield) * drate_dcb
    Jacobian(this%is_ch4, this%is_cbio) = Jacobian(this%is_ch4,this%is_cbio) &
                                        + drate_dcb
    Jacobian(this%is_cbio,this%is_cbio) = Jacobian(this%is_cbio,this%is_cbio) &
                                        + this%yield * drate_dcb
    Jacobian(this%is_hco3,this%is_cbio) = Jacobian(this%is_hco3,this%is_cbio) &
                                          + 0.5 * this%yield * drate_dcb

  endif
```

For $k_{Ac} = 0$,

$$\frac{\partial \mathrm{Ac}}{\partial t} = -(1 + 0.5y)k\mathrm{C}_{\mathrm{bio}}$$
$$\frac{\partial \mathrm{C}_{\mathrm{bio}}}{\partial t} = 1000\theta yk\mathrm{C}_{\mathrm{bio}}$$

$$\mathrm{C}_{\mathrm{bio}} = \mathrm{C}_{\mathrm{bio,0}}\mathrm{EXP}(1000\theta ykt)$$
$$\mathrm{Ac} = \mathrm{Ac}_0 - \frac{1+0.5y}{1000\theta y}\mathrm{C}_{\mathrm{bio,0}}\mathrm{EXP}(1000\theta ykt)$$

This analytical solution is used in the following figure to check the numerical solution. Note $\mathrm{C}_{\mathrm{bio}}$ is supposed to stop increasing when acetate is exhausted.

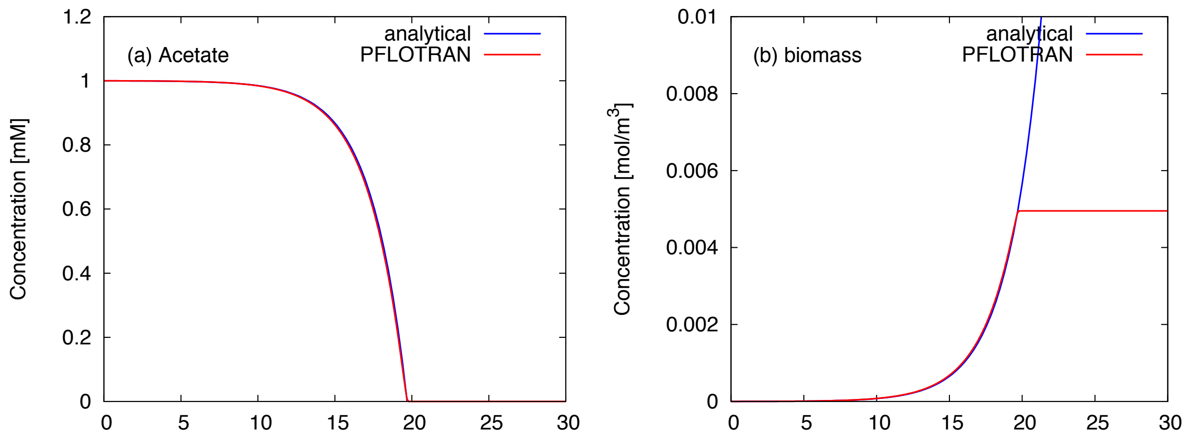

Figure 4: Example calculation for acetoclastic methanogenesis:

4 A General Example

If we implement a number of general reactions and rate formulae in PFLOTRAN, we can add as many specific reactions with specific parameter values in the input file. By doing this, we do not have to change the source code or develop a new reaction_sandbox for each specific case. For example, if we consider decomposition of LabileC as a first order decay to produce acetate and H_2, which are used by methanogens to produce methane using the following reactions:

$\text{LabileC} = \frac{1}{3}\ \text{Acetate}^- + \frac{1}{3}\ HCO_3^- + \frac{1}{9}\ H_2 + \frac{2}{3}\ H^+$

$\text{Acetate}^- + H_2O = \text{Methane} + HCO_3^-$

$H_2 + \frac{1}{4}\ HCO_3^- + \frac{1}{4}\ H^+ = \frac{1}{4}\ \text{Methane} + \frac{3}{4}\ H_2O$

We can use the GENERAL_REACTION and MICROBIAL_REACTION functions in PFLOTRAN to specify the reactions and parameter values as follow:

```
CHEMISTRY
  PRIMARY_SPECIES
    A(aq)
    Acetate-
    H2(aq)
    H+
    HCO3-
    Methane(aq)
    Na+
  /
  SECONDARY_SPECIES
    OH-
    CO3--
    CO2(aq)
:     Acetic_acid(aq)
  /
  REDOX_SPECIES
    Acetate-
    Methane(aq)
    H2(aq)
    H+
  /
  IMMOBILE_SPECIES
    Acmeg
    H2meg
  /

  GENERAL_REACTION
    REACTION A(aq) <-> 0.3333 Acetate- + 0.3333 HCO3- + 0.1111 H2(aq) + 0.6666 H+
    FORWARD_RATE 1.3889d-5  ! 1/s
    BACKWARD_RATE 0.d0
  /
  MICROBIAL_REACTION
    REACTION Acetate- + H2O <-> Methane(aq) + HCO3-
    RATE_CONSTANT     1.0d-6
    MONOD Acetate-    1.0d-5
    BIOMASS           Acmeg 0.01
  /

  MICROBIAL_REACTION
    REACTION H2(aq) + 0.25 HCO3- + 0.25 H+ <-> 0.25 Methane(aq) + 0.75 H2O
    RATE_CONSTANT     1.0d-5
    MONOD H2(aq)      1.0d-7
    BIOMASS           H2meg 0.02
  /

  DATABASE ../../pflotran-clm4me/database/hanford.dat
```

With initial conditions as follow,

```
CONSTRAINT initial
  CONCENTRATIONS
    A(aq)         0.001 T
    Acetate-    1.0d-10 T
    H2(aq)      1.0d-10 T
    H+              7.0 pH
    HCO3-        5.0d-3 T
    Methane(aq) 1.0d-10 T
    Na+          5.0d-3 Z
  /
  IMMOBILE
    Acmeg       1.0d-5
    H2meg        1.0d-7
  /
END
```

PFLOTRAN will give results like in the following figure. The point is that we can add many reactions in the input file.

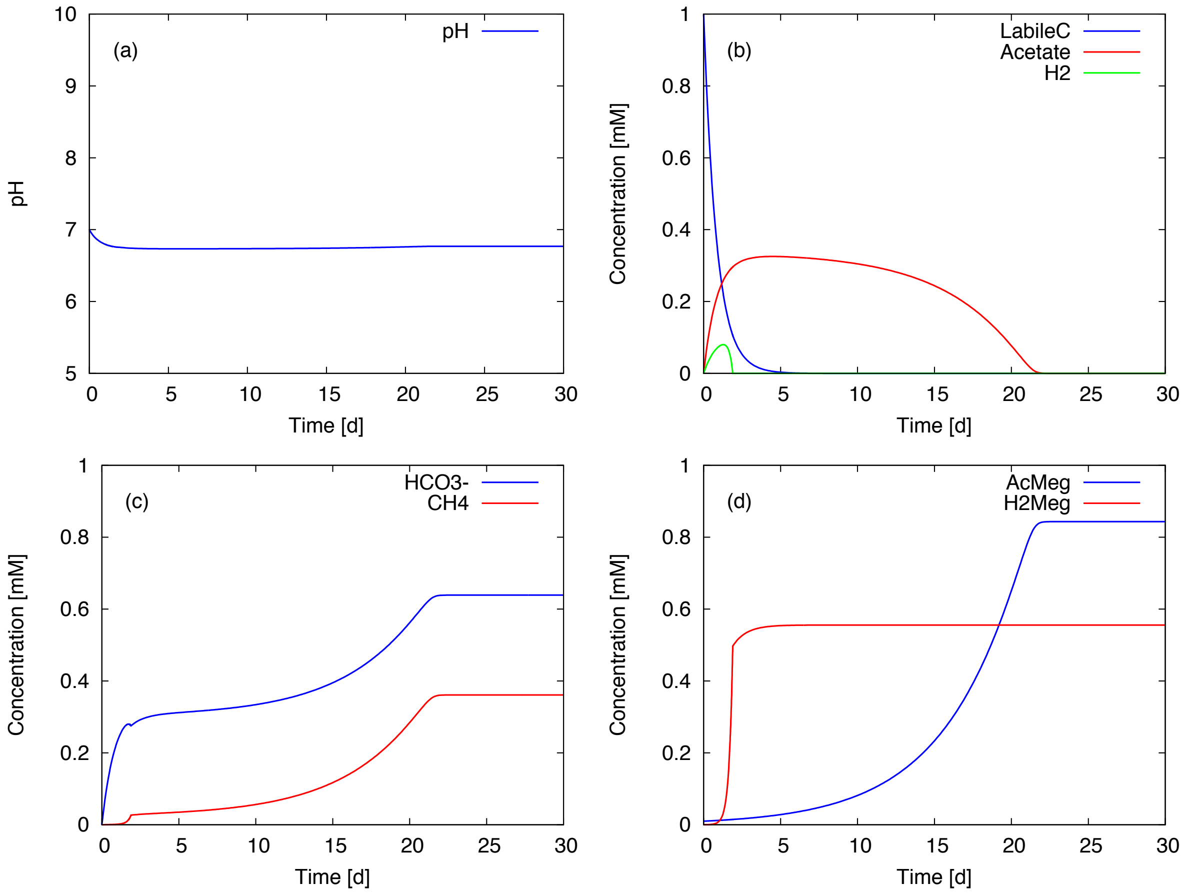

Figure 5: Example calculation for multiple microbial reactions

5 CLM-CNP

Starting from CLM-CN, CLM-CNP provides options to

1. add microbial, enzyme, and DOC pools
2. allow variable C:N:P ratios in all of the pools
3. use Monod, Machielis-Menten and other rate models
4. specify immobile or mobile pools/species

through an input file. It reduces to CLM-CN if none of the additional features is specified in the input file. These added features can be added incrementally. Sorption, and other geochemical processes can be added separately (outside of the CLM-CNP reaction_sandbox).

5.1 C

5.1.1 Reaction

$$\mathrm{C}_u \rightarrow c_m\mathrm{C}_m + c_b\mathrm{C}_b + c_e\mathrm{C}_e + c_d\mathrm{C}_d \tag{18}$$

or

$$\mathrm{C}_u \rightarrow \sum c_i\mathrm{C}_i \tag{19}$$

The subscript u, m, b, e, and d denote upstream, mineral, bacterial, enzyme, and downstream pools. To balance the reaction, $c_m + c_b + c_e + c_d = 1$. c_b and c_e can be variable in the future.

For CLM-CN, c_m is the respiration fraction f, $c_d = 1 - f$, and $c_b = c_e = 0$.

To incorporate microbial pools with Monod rate, $c_b \neq 0$.

To use Michaelis-Menten rate, $c_e \neq 0$

5.1.2 Rate

$$\frac{d[\mathrm{C}_u]}{dt} = -R = -k\prod f([\mathrm{C}_i])f(pH)f(\psi)f(T) \tag{20}$$

$[\mathrm{C}_i]$ is the concentration of C_i. k is the rate coefficient. function $f([\mathrm{C}_i])$, $f(pH)$, $f(\psi)$, and $f(T)$ account for influence of C_i, pH, moisture, and temperature on the reaction rate. C_i can be any component in or not in Eq. (18). We consider four options for $f([\mathrm{C}_i])$:

$f([\mathrm{C}_i]) = 1$	default
$f([\mathrm{C}_i]) = [\mathrm{C}_i]$	first order
$f([\mathrm{C}_i]) = [\mathrm{C}_i]/(K_{Ci} + [\mathrm{C}_i])$	substrate/electron donor or electron acceptor limitation
$f([\mathrm{C}_i]) = I_{Ci}/(I_{Ci} + [\mathrm{C}_i])$	inhibition

For CLM-CN, $f([\mathrm{C}_u]) = [\mathrm{C}_u]$, $f([\mathrm{C}_m]) = f([\mathrm{C}_b]) = f([\mathrm{C}_d]) = 1$.

To incorporate microbial pools without Monod rate, $f([\mathrm{C}_u]) = [\mathrm{C}_u]/(K_{C_u} + [\mathrm{C}_u])$, $f([\mathrm{C}_b]) = [\mathrm{C}_b]$, $f([\mathrm{C}_m]) = f([\mathrm{C}_d]) = 1$.

To use Michaelis-Menten rate, $f([\mathrm{C}_u]) = [\mathrm{C}_u]/(K_{C_u} + [\mathrm{C}_u])$, $f([\mathrm{C}_e]) = [\mathrm{C}_e]$, $f([\mathrm{C}_m]) = f([\mathrm{C}_d]) = f([\mathrm{C}_b]) = 1$.

5.2 N

5.2.1 Reaction

$$n_u\mathrm{N}_u \rightarrow n_m\mathrm{N}_m + n_b\mathrm{N}_b + n_e\mathrm{N}_e + n_d\mathrm{N}_d \tag{21}$$

or

$$n_u\mathrm{N}_u \rightarrow \sum n_i\mathrm{N}_i \tag{22}$$

$$n_m + n_b + n_e + n_d = n_u \tag{23}$$

$n_u = [\mathrm{N}_u]/[\mathrm{C}_u]$, $n_b = [\mathrm{N}_b]/[\mathrm{C}_b]$, $n_e = [\mathrm{N}_e]/[\mathrm{C}_e]$, and $n_d = [\mathrm{N}_d]/[\mathrm{C}_d]$. These stoichiometric coefficient can be fixed or variable. If $n_m > 0$, this decomposition reaction produces mineral N (mineralization). Otherwise, the reaction takes up mineral N (immobilization). In the late case, a $f([\mathrm{N}_m]) = [\mathrm{N}_m]/(K_{Nm} + [\mathrm{N}_m])$ term is added tothe decomposition rate in Eq. (20).

5.2.2 Rate

$$-\frac{1}{n_u}\frac{\partial[\mathrm{N}_u]}{\partial t} = \frac{1}{n_m}\frac{\partial[\mathrm{N}_m]}{\partial t} = \frac{1}{n_b}\frac{\partial[\mathrm{N}_b]}{\partial t} = \frac{1}{n_e}\frac{\partial[\mathrm{N}_e]}{\partial t} = \frac{1}{n_d}\frac{\partial[\mathrm{N}_d]}{\partial t} \tag{24}$$

5.3 P

5.3.1 Reaction

$$p_u\mathrm{P}_u \rightarrow p_m\mathrm{P}_m + p_b\mathrm{P}_b + p_e\mathrm{P}_e + p_d\mathrm{P}_d \tag{25}$$

$$p_u\mathrm{P}_u \rightarrow \sum p_i\mathrm{P}_i \tag{26}$$

$$p_m + p_b + p_e + p_d = p_u \tag{27}$$

$p_u = [\mathrm{P}_u]/[\mathrm{C}_u]$, $p_b = [\mathrm{P}_b]/[\mathrm{C}_b]$, $p_e = [\mathrm{P}_e]/[\mathrm{C}_e]$, and $p_d = [\mathrm{P}_d]/[\mathrm{C}_d]$. These stoichiometric coefficient can be fixed or variable. If $p_m > 0$, this decomposition reaction produces mineral P (mineralization). Otherwise, the reaction takes up mineral P (immobilization). In the late case, a $f([\mathrm{P}_m]) = [\mathrm{P}_m]/(K_{Pm} + [\mathrm{P}_m])$ term is added tothe decomposition rate in Eq. (20).

5.3.2 Rate

$$-\frac{1}{p_u}\frac{\partial[\mathrm{P}_u]}{\partial t} = \frac{1}{p_m}\frac{\partial[\mathrm{P}_m]}{\partial t} = \frac{1}{p_b}\frac{\partial[\mathrm{P}_b]}{\partial t} = \frac{1}{p_e}\frac{\partial[\mathrm{P}_e]}{\partial t} = \frac{1}{p_d}\frac{\partial[\mathrm{P}_d]}{\partial t} \tag{28}$$

5.4 Residuals

$$\begin{aligned}
R_{Cu} &= -R \\
R_{Cdi} &= d_i R \\
R_{Cm} &= (1-d_i)R \\
R_{Nu} &= -\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]}R \\
R_{Ndi} &= \frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}R \\
R_{Nm} &= \left(\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]} - \frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\right)R
\end{aligned}$$

5.5 Jacobians

Table 8: Jacobian for general decomposition

	C_u	C_{di}	C_m	N_u	N_{di}	N_m
C_u	$-\frac{\partial R}{\partial \mathrm{C}_u}$	$-\frac{\partial R}{\partial \mathrm{C}_{di}}$	0	0	0	$-\frac{\partial R}{\partial \mathrm{N}_m}$
C_{di}	$d_i\frac{\partial R}{\partial \mathrm{C}_u}$	$d_i\frac{\partial R}{\partial \mathrm{C}_{di}}$	0	0	0	$d_i\frac{\partial R}{\partial \mathrm{N}_m}$
C_m	$(1-d_i)\frac{\partial R}{\partial \mathrm{C}_u}$	$(1-d_i)\frac{\partial R}{\partial \mathrm{C}_{di}}$	0	0	0	$(1-d_i)\frac{\partial R}{\partial \mathrm{N}_m}$
N_u	$-\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]}\frac{\partial R}{\partial \mathrm{C}_u} + \frac{[\mathrm{N}_u]}{[\mathrm{C}_u]^2}R$	$-\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]}\frac{\partial R}{\partial \mathrm{C}_{di}}$	0	$-\frac{1}{[\mathrm{C}_u]}R$	0	$-\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]}\frac{\partial R}{\partial \mathrm{N}_m}$
N_{di}	$\frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\frac{\partial R}{\partial \mathrm{C}_u}$	$\frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\frac{\partial R}{\partial \mathrm{C}_{di}} - \frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]^2}R$	0	0	$\frac{1}{[\mathrm{C}_{di}]}R$	$\frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\frac{\partial R}{\partial \mathrm{N}_m}$
N_m	$\left(\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]} - \frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\right)\frac{\partial R}{\partial \mathrm{C}_u} - \frac{[\mathrm{N}_u]}{[\mathrm{C}_u]^2}R$	$\left(\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]} - \frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\right)\frac{\partial R}{\partial \mathrm{C}_{di}} + \frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]^2}R$	0	$\frac{1}{[\mathrm{C}_u]}R$	$-\frac{1}{[\mathrm{C}_{di}]}R$	$\left(\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]} - \frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\right)\frac{\partial R}{\partial \mathrm{N}_m}$

To use $\mathrm{N}_m/(\mathrm{K}_N + \mathrm{N}_m)$ term for N-limiting cases ($\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]} - \frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]} < 0$),

$$\frac{\partial R}{\partial \mathrm{N}_m} = R\frac{K_N}{(K_N + \mathrm{N}_m)\mathrm{N}_m} \tag{29}$$

because

$$\frac{\partial \frac{x}{k+x}}{\partial x} = \frac{k}{(k+x)^2} \tag{30}$$

For the first order rate term,

$$\frac{\partial R}{\partial \mathrm{C}_x} = \frac{R}{[\mathrm{C}_x]} \tag{31}$$

For the Monod rate term,

$$\frac{\partial R}{\partial \mathrm{C}_x} = R\frac{K_x}{(K_x + [\mathrm{C}_x])[\mathrm{C}_x]} \tag{32}$$

For the inhibition term,

$$\frac{\partial R}{\partial \mathrm{C}_x} = -R\frac{K_x}{(K_x + [\mathrm{C}_x])[\mathrm{C}_x]} \tag{33}$$

because

$$\frac{\partial \frac{k}{k+x}}{\partial x} = -\frac{k}{(k+x)^2} \tag{34}$$

Table 9: Jacobian for general decomposition

	C_u	C_{di}	C_m	N_u	N_{di}	N_m
C_u	$-\frac{\partial R}{\partial \mathrm{C}_u}$	$-\frac{\partial R}{\partial \mathrm{C}_{di}}$	0	0	0	$-\frac{\partial R}{\partial \mathrm{N}_m}$
C_{di}	$d_i\frac{\partial R}{\partial \mathrm{C}_u}$	$d_i\frac{\partial R}{\partial \mathrm{C}_{di}}$	0	0	0	$d_i\frac{\partial R}{\partial \mathrm{N}_m}$
C_m	$(1-d_i)\frac{\partial R}{\partial \mathrm{C}_u}$	$(1-d_i)\frac{\partial R}{\partial \mathrm{C}_{di}}$	0	0	0	$(1-d_i)\frac{\partial R}{\partial \mathrm{N}_m}$
N_u	$-\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]}\frac{\partial R}{\partial \mathrm{C}_u}$	$-\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]}\frac{\partial R}{\partial \mathrm{C}_{di}}$	0	0	0	$-\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]}\frac{\partial R}{\partial \mathrm{N}_m}$
N_{di}	$\frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\frac{\partial R}{\partial \mathrm{C}_u}$	$\frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\frac{\partial R}{\partial \mathrm{C}_{di}}$	0	0	0	$\frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\frac{\partial R}{\partial \mathrm{N}_m}$
N_m	$\left(\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]}-\frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\right)\frac{\partial R}{\partial \mathrm{C}_u}$	$\left(\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]}-\frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\right)\frac{\partial R}{\partial \mathrm{C}_{di}}$	0	0	0	$\left(\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]}-\frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]}\right)\frac{\partial R}{\partial \mathrm{N}_m}$

Table 10: Jacobian for general decomposition

	C_u	C_{di}	C_m	N_u	N_{di}	N_m
C_u	0	0	0	0	0	0
C_{di}	0	0	0	0	0	0
C_m	$(1-d_i)\frac{\partial R}{\partial \mathrm{C}_u}$	$(1-d_i)\frac{\partial R}{\partial \mathrm{C}_{di}}$	0	0	0	$(1-d_i)\frac{\partial R}{\partial \mathrm{N}_m}$
N_u	$\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]^2}R$	0	0	$-\frac{1}{[\mathrm{C}_u]}R$	0	0
N_{di}	0	$-\frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]^2}R$	0	0	$\frac{1}{[\mathrm{C}_{di}]}R$	0
N_m	$-\frac{[\mathrm{N}_u]}{[\mathrm{C}_u]^2}R$	$\frac{[\mathrm{N}_{di}]}{[\mathrm{C}_{di}]^2}R$	0	$\frac{1}{[\mathrm{C}_u]}R$	$-\frac{1}{[\mathrm{C}_{di}]}R$	0

```
subroutine CLM_CNPReact(this,Residual,Jacobian,compute_derivative, &
                        rt_auxvar,global_auxvar,porosity,volume,reaction, &
                        option)

  use Option_module
  use Reaction_Aux_module, only : reaction_type, GetPrimarySpeciesIDFromName

  implicit none

  class(reaction_sandbox_CLM_CNP_type) :: this
  type(option_type) :: option
  type(reaction_type) :: reaction
  PetscBool :: compute_derivative
  PetscReal :: Residual(reaction%ncomp)
  PetscReal :: Jacobian(reaction%ncomp,reaction%ncomp)
  PetscReal :: porosity
  PetscReal :: volume
  type(reactive_transport_auxvar_type) :: rt_auxvar
  type(global_auxvar_type) :: global_auxvar

  PetscInt, parameter :: iphase = 1
  PetscInt :: offset
  PetscInt :: i, j, ires, ires_j, ires_n
  PetscReal :: conc
  PetscReal :: L_water
  PetscReal :: rate, drate
  PetscReal :: tmp_real

  offset = reaction%offset_immobile

  rate = this%rate_constant * volume

! first order term
  do i = 1, this%nFirstOrder
     conc = rt_auxvar%immobile(this%ispec_1st(i))
     rate = rate * conc
  enddo

! monod term
  do i = 1, this%nMonod
     conc = rt_auxvar%immobile(this%ispec_mnd(i))
     rate = rate * conc/(conc + this%half_saturation(i))
  enddo

! inhibition term
  do i = 1, this%nInhibition
```

```
      conc = rt_auxvar%immobile(this%ispec_inh(i))
      rate = rate * this%inhibition_coef(i)/(conc + this%inhibition_coef(i))
    enddo

  ! N limiting
    if((this%bNEnabled)) then
  ! upstream CN
      write(*, *) this%Upstream%name_c, this%Upstream%name_n, this%Upstream%ratio_cn, t
      if(this%upstream_ispec_n > 0) then
         if(rt_auxvar%immobile(this%upstream_ispec_c) .LE. 1.0d-10) then
            write(option%fid_out,*) 'Upstream C concentration is 0 in CN ratio calculat
         endif
         this%upstream_stoich_n = rt_auxvar%immobile(this%upstream_ispec_n) &
                         / max(rt_auxvar%immobile(this%upstream_ispec_c), 1.0d-10)
      else
        if(this%upstream_cn .LT. 1.0d-10) then
            option%io_buffer = 'CHEMISTRY,REACTION_SANDBOX,CLM_CNP check:' // &
              ' upstream CN ratio is zero.'
            call printErrMsg(option)
         endif
         this%upstream_stoich_n = 1.0d0 / this%upstream_cn
      endif

  ! downstream CN
      this%mineral_n_stoich = this%upstream_stoich_n   ! start from upstream N, substra
      do i = 1, this%nDownstream
         if(this%downstream_ispec_n(i) > 0) then
            if(rt_auxvar%immobile(this%downstream_ispec_c(i)) .LE. 1.0d-10) then
               write(option%fid_out,*) 'Downtream C concentration < 1d-10 in CN ratio c
            endif
            this%downstream_stoich_n(i) = rt_auxvar%immobile(this%downstream_ispec_n(i)
                         / max(rt_auxvar%immobile(this%downstream_ispec_c(i)), 1.0d-1
         else
            this%downstream_stoich_n(i) = 1.0d0 / this%downstream_cn(i)
         endif
         this%mineral_n_stoich = this%mineral_n_stoich - this%downstream_stoich_n(i)
      enddo

      if(this%mineral_n_stoich .LT. 0.0d0) then
         conc = rt_auxvar%immobile(this%mineral_n_ispec)
         rate = rate * conc/(conc + N_inhibition_constant)
      endif
    endif

  ! residuals
    ires = reaction%offset_immobile + this%upstream_ispec_c
    Residual(ires) = Residual(ires) + rate

    ires = reaction%offset_immobile + this%mineral_c_ispec
    Residual(ires) = Residual(ires) - rate * this%mineral_c_stoich

    do i = 1, this%nDownstream
       ires = reaction%offset_immobile + this%downstream_ispec_c(i)
       Residual(ires) = Residual(ires) - rate * this%downstream_stoich_c(i)
    enddo

    if(this%bNEnabled) then
       if(.not.this%bfixed_upstream_cn) then
          ires = reaction%offset_immobile + this%upstream_ispec_n
          Residual(ires) = Residual(ires) + rate * this%upstream_stoich_n
       endif

       ires_n = reaction%offset_immobile + this%mineral_n_ispec
       Residual(ires_n) = Residual(ires_n) - rate * this%mineral_n_stoich

       do i = 1, this%nDownstream
         if(.not.this%bfixed_downstream_cn(i)) then
            ires = reaction%offset_immobile + this%downstream_ispec_n(i)
            Residual(ires) = Residual(ires) - rate * this%downstream_stoich_n(i)
         endif
       enddo
    endif

  ! Jacobian
    if (compute_derivative) then
  !  first order terms
     do j = 1, this%nFirstOrder
       conc = rt_auxvar%immobile(this%ispec_1st(j))
       drate = rate / conc
       ires_j = reaction%offset_immobile + this%ispec_1st(j)

       ires = reaction%offset_immobile + this%upstream_ispec_c
       Jacobian(ires,ires_j) = Jacobian(ires,ires_j) + drate

       ires = reaction%offset_immobile + this%mineral_c_ispec
       Jacobian(ires, ires_j) = Jacobian(ires, ires_j) - drate * this%mineral_c_stoich

       do i = 1, this%nDownstream
          ires = reaction%offset_immobile + this%downstream_ispec_c(i)
          Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - drate * this%downstream_stoic
       enddo
      if(this%bNEnabled) then
        if(.not.this%bfixed_upstream_cn) then
         ires = reaction%offset_immobile + this%upstream_ispec_n
         Jacobian(ires,ires_j) = Jacobian(ires,ires_j) + drate * this%upstream_stoich_n

         if((this%ispec_1st(j) .eq. this%upstream_ispec_c)) then
            Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - rate * this%upstream_stoich
                                  / rt_auxvar%immobile(this%upstream_ispec_c)
         endif
        endif

         ires_n = reaction%offset_immobile + this%mineral_n_ispec
         Jacobian(ires_n,ires_j) = Jacobian(ires_n,ires_j) - drate * this%mineral_n_sto

         if((.not.this%bfixed_upstream_cn) .and. (this%ispec_1st(j) .eq. this%upstream_
            Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - rate * this%mineral_n_stoic
                                  / rt_auxvar%immobile(this%upstream_ispec_c)
         endif

         do i = 1, this%nDownstream
          if(.not.this%bfixed_downstream_cn(i)) then
            ires = reaction%offset_immobile + this%downstream_ispec_n(i)
            Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - drate * this%downstream_sto

            if(this%ispec_1st(j) .eq. this%downstream_ispec_c(i)) then
               Jacobian(ires,ires_j) = Jacobian(ires,ires_j) &
                                     - rate * this%downstream_stoich_n(i) &
                                     / rt_auxvar%immobile(this%downstream_ispec_c(i))

               Jacobian(ires_n,ires_j) = Jacobian(ires_n,ires_j) - rate * this%mineral_
                                  / rt_auxvar%immobile(this%downstream_ispec_c(i))
            endif
          endif
         enddo
      endif
     enddo

  !  monod terms
     do j = 1, this%nMonod
       conc = rt_auxvar%immobile(this%ispec_mnd(j))
       drate = -rate / (this%half_saturation(j) + conc)
       ires_j = reaction%offset_immobile + this%ispec_mnd(j)

       ires = reaction%offset_immobile + this%upstream_ispec_c
       Jacobian(ires, ires_j) = Jacobian(ires, ires_j) + drate
```

```
        ires = reaction%offset_immobile + this%mineral_c_ispec
        Jacobian(ires, ires_j) = Jacobian(ires, ires_j) - drate * this%mineral_c_stoich

        do i = 1, this%nDownstream
           ires = reaction%offset_immobile + this%downstream_ispec_c(i)
           Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - drate * this%downstream_stoic
        enddo

       if(this%bNEnabled) then
         if(.not.this%bfixed_upstream_cn) then
          ires = reaction%offset_immobile + this%upstream_ispec_n
          Jacobian(ires,ires_j) = Jacobian(ires,ires_j) + drate * this%upstream_stoich_n

          if(this%ispec_1st(j) .eq. this%upstream_ispec_c) then
             Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - rate * this%upstream_stoich
                                   / rt_auxvar%immobile(this%upstream_ispec_c)
          endif
         endif

          ires_n = reaction%offset_immobile + this%mineral_n_ispec
          Jacobian(ires_n,ires_j) = Jacobian(ires_n,ires_j) - drate * this%mineral_n_sto

          if((.not.this%bfixed_upstream_cn) .and. (this%ispec_mnd(j) .eq. this%upstream_
             Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - rate * this%mineral_n_stoic
                                   / rt_auxvar%immobile(this%upstream_ispec_c)
          endif

          do i = 1, this%nDownstream
           if(.not.this%bfixed_downstream_cn(i)) then
             ires = reaction%offset_immobile + this%downstream_ispec_n(i)
             Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - drate * this%downstream_sto

             Jacobian(ires_n,ires_j) = Jacobian(ires_n,ires_j) - rate * this%mineral_n_s
                                   / rt_auxvar%immobile(this%downstream_ispec_c(i))

             if(this%ispec_mnd(j) .eq. this%downstream_ispec_c(i)) then
                Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - rate * this%downstream_s
                                   / rt_auxvar%immobile(this%downstream_ispec_c(i))
             endif
           endif
          enddo
       endif
      enddo

!   inhibition terms
    do j = 1, this%nInhibition
```

```
conc = rt_auxvar%immobile(this%ispec_inh(j))
drate = -rate / (this%inhibition_coef(j) + conc)/this%inhibition_coef(j)
ires_j = reaction%offset_immobile + this%ispec_inh(j)

ires = reaction%offset_immobile + this%upstream_ispec_c
Jacobian(ires, ires_j) = Jacobian(ires, ires_j) + drate

ires = reaction%offset_immobile + this%mineral_c_ispec
Jacobian(ires, ires_j) = Jacobian(ires, ires_j) - drate * this%mineral_c_stoich

do i = 1, this%nDownstream
   ires = reaction%offset_immobile + this%downstream_ispec_c(i)
   Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - drate * this%downstream_stoic
enddo

if(this%bNEnabled) then
  if(.not.this%bfixed_upstream_cn) then
   ires = reaction%offset_immobile + this%upstream_ispec_n
   Jacobian(ires,ires_j) = Jacobian(ires,ires_j) + drate * this%upstream_stoich_n

   if(this%ispec_1st(j) .eq. this%upstream_ispec_c) then
      Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - rate * this%upstream_stoich
                            / rt_auxvar%immobile(this%upstream_ispec_c)
   endif
  endif

   ires_n = reaction%offset_immobile + this%mineral_n_ispec
   Jacobian(ires_n,ires_j) = Jacobian(ires_n,ires_j) - drate * this%mineral_n_sto

   if((.not.this%bfixed_upstream_cn) .and. (this%ispec_inh(j) .eq. this%upstream_
      Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - rate * this%mineral_n_stoic
                            / rt_auxvar%immobile(this%upstream_ispec_c)
   endif

   do i = 1, this%nDownstream
      ires = reaction%offset_immobile + this%downstream_ispec_n(i)
      Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - drate * this%downstream_sto

      if((.not.this%bfixed_downstream_cn(i)) .and. (this%ispec_inh(j) .eq. this%d
         Jacobian(ires,ires_j) = Jacobian(ires,ires_j) - rate * this%downstream_s
                            / rt_auxvar%immobile(this%downstream_ispec_c(i))

         Jacobian(ires_n,ires_j) = Jacobian(ires_n,ires_j) - rate * this%mineral_
                            / rt_auxvar%immobile(this%downstream_ispec_c(i))
      endif
   enddo
```

```
    endif
   enddo

! N limiting
   if(this%bNEnabled .and. this%mineral_n_stoich .LT. 0.0d0) then
     conc = rt_auxvar%immobile(this%mineral_n_ispec)
     drate = -rate / (N_inhibition_constant + conc)
     ires_n = reaction%offset_immobile + this%mineral_n_ispec

     ires = reaction%offset_immobile + this%upstream_ispec_c
     Jacobian(ires, ires_n) = Jacobian(ires, ires_n) + drate

     ires = reaction%offset_immobile + this%mineral_c_ispec
     Jacobian(ires, ires_n) = Jacobian(ires, ires_n) - drate * this%mineral_c_stoich

     do i = 1, this%nDownstream
        ires = reaction%offset_immobile + this%downstream_ispec_c(i)
        Jacobian(ires,ires_n) = Jacobian(ires,ires_n) - drate * this%downstream_stoic
     enddo

       ires = reaction%offset_immobile + this%upstream_ispec_n
       Jacobian(ires,ires_n) = Jacobian(ires,ires_n) + drate * this%upstream_stoich_n

       if((.not.this%bfixed_upstream_cn)) then
          Jacobian(ires,ires_n) = Jacobian(ires,ires_n) - rate * this%upstream_stoich
                              / rt_auxvar%immobile(this%upstream_ispec_c)
       endif

       Jacobian(ires_n,ires_n) = Jacobian(ires_n,ires_n) - drate * this%mineral_n_sto

       if((.not.this%bfixed_upstream_cn)) then
          Jacobian(ires,ires_n) = Jacobian(ires,ires_n) - rate * this%mineral_n_stoic
                              / rt_auxvar%immobile(this%upstream_ispec_c)
       endif

       do i = 1, this%nDownstream
          ires = reaction%offset_immobile + this%downstream_ispec_n(i)
          Jacobian(ires,ires_n) = Jacobian(ires,ires_n) - drate * this%downstream_sto

          if(.not.this%bfixed_downstream_cn(i)) then
             Jacobian(ires,ires_n) = Jacobian(ires,ires_n) - rate * this%downstream_s
                              / rt_auxvar%immobile(this%downstream_ispec_c(i))

             Jacobian(ires_n,ires_n) = Jacobian(ires_n,ires_n) - rate * this%mineral_
                              / rt_auxvar%immobile(this%downstream_ispec_c(i))
          endif
       enddo
   endif  ! if N limiting
  endif   ! if (compute_derivative) then
end subroutine CLM_CNPReact
```

References

E. Carol Adair, William J. Parton, Steven J. del Grosso, Whendee L. Silver, Mark E. Harmon, Sonia A. Hall, Ingrid C. Burke, and Stephen C. Hart. Simple three-pool model accurately describes patterns of long-term litter decomposition in diverse climates. *Global Change Biology*, 14(11):2636–2660, 2008. ISSN 1365-2486. doi: 10.1111/j.1365-2486.2008.01674.x. URL `http://dx.doi.org/10.1111/j.1365-2486.2008.01674.x`.

Gordon B. Bonan, Melannie D. Hartman, William J. Parton, and William R. Wieder. Evaluating litter decomposition in earth system models with long-term litterbag experiments: an example using the community land model version 4 (clm4). *Global Change Biology*, pages n/a–n/a, 2012. ISSN 1365-2486. doi: 10.1111/gcb.12031. URL `http://dx.doi.org/10.1111/gcb.12031`.