

# Response of Systems to a Perturbation from an Equilibrium — Part II

Reader Accompanying the Course Reaction Transport Modelling in the Hydrosphere

Lubos Polerecky and Karline Soetaert, Utrecht University

April 2021

## Abstract

Here we illustrate how the theoretical analysis explained in Part I helps us gain more insights into the response of the global C cycle to a perturbation. First, we rewrite the differential equations describing the C cycle model using a vector-matrix formalism. Then, we calculate the Jacobian matrix and its eigen-values and eigen-vectors. From these, we then estimate the characteristic *response times* of the system to a perturbation, and the corresponding amounts of C that are *exchanged* within the system on these characteristic time scales. Because the global C cycle model is linear in the state variables, the results of the approach presented here are valid for any magnitude of the perturbation. For non-linear models, the approach can be applied as well, but the results will only be valid for small perturbations around an equilibrium state.

## Global carbon cycle model — the 7-box model

It is recommended that you become familiar with the model of the Earth's global C cycle discussed in class before reading this document. The model considers 7 compartments: atmosphere ( $M_1$ ), surface ocean ( $M_2$ ), marine biota ( $M_3$ ), deep ocean ( $M_4$ ), marine sediments ( $M_5$ ), terrestrial biota ( $M_6$ ), and rocks and fossils ( $M_7$ ).

### Model formulation

First, we reproduce the differential equations describing the model:

$$\begin{aligned}\frac{dM_1}{dt} &= -(r_{12} + r_{16}) \cdot M_1 + r_{21} \cdot M_2 + r_{61} \cdot M_6 \\ \frac{dM_2}{dt} &= r_{12} \cdot M_1 - (r_{21} + r_{23} + r_{24}) \cdot M_2 + r_{32} \cdot M_3 + r_{42} \cdot M_4 + r_{62} \cdot M_6 + r_{72} \cdot M_7 \\ \frac{dM_3}{dt} &= r_{23} \cdot M_2 - (r_{32} + r_{34}) \cdot M_3 \\ \frac{dM_4}{dt} &= r_{24} \cdot M_2 + r_{34} \cdot M_3 - (r_{42} + r_{45}) \cdot M_4 \\ \frac{dM_5}{dt} &= r_{45} \cdot M_4 - r_{57} \cdot M_5 \\ \frac{dM_6}{dt} &= r_{16} \cdot M_1 - (r_{61} + r_{62}) \cdot M_6 \\ \frac{dM_7}{dt} &= r_{57} \cdot M_5 - r_{72} \cdot M_7\end{aligned}\tag{1}$$

## Model implementation in R

Based on these equations, we implement the model in R. The code below also shows the values of the model parameters and the equilibrium C contents of the modelled compartments. We simplify the analysis by omitting fuel emissions.

```
state.eq <- c(                                     # equilibrium carbon contents, [Pg C]
  M1.ATMOSPH   = 590,                             # atmosphere
  M2.SURFOCE   = 900,                             # surface ocean
  M3.MARBIOT    = 3,                               # marine biota
  M4.DEEPOCE   = 37100,                           # deep ocean
  M5.SEDIM     = 150,                              # ocean sediment
  M6.LANDBIO   = 2300,                             # land biosphere
  M7.ROCKFOS   = 60e6+15e6+11e3 # sediments & rocks & fossil
)

# Model parameters:
parms <- with (as.list(state.eq), c(

  # rate constants, yr-1 (first-order kinetics for all fluxes)
  r1.2 = 70.2 /M1.ATMOSPH, # atm    -> surfoc   (Uptake+weathering)
  r2.1 = 70.6 /M2.SURFOCE, # surfoc -> atm
  r2.3 = 50   /M2.SURFOCE, # surfoc -> marbio
  r3.2 = 39   /M3.MARBIOT, # marbio -> surfoc
  r3.4 = 11   /M3.MARBIOT, # marbio -> deepoc
  r4.2 = 101  /M4.DEEPOCE, # deepoc -> surfoc
  r2.4 = 90.2 /M2.SURFOCE, # surfoc -> deepoc
  r4.5 = 0.2  /M4.DEEPOCE, # deepoc -> sed
  r5.7 = 0.2  /M5.SEDIM,   # sed    -> rocks
  r6.2 = 0.4  /M6.LANDBIO, # landbio -> surfoc
  r1.6 = 60   /M1.ATMOSPH, # atm    -> landbio
  r6.1 = 59.6 /M6.LANDBIO, # landbio -> atm
  r7.2 = 0.2  /M7.ROCKFOS, # rocks_fossil -> surface ocean
  r7.4 = 0.0  /M7.ROCKFOS # rocks_fossil -> deep ocean
)
)

# Model function:
Cmodel <- function(t, C, p) {

  with(as.list (c(C, p)),{ # all fluxes in [PgC/yr]
    F1.2 <- r1.2*M1.ATMOSPH # atm    -> surfoc
    F1.6 <- r1.6*M1.ATMOSPH # atm    -> landbio
    F2.1 <- r2.1*M2.SURFOCE # surfoc -> atm
    F2.3 <- r2.3*M2.SURFOCE # surfoc -> marbio
    F2.4 <- r2.4*M2.SURFOCE # surfoc -> deepoc
    F3.2 <- r3.2*M3.MARBIOT # marbio -> surfoc
    F3.4 <- r3.4*M3.MARBIOT # marbio -> deepoc
    F4.2 <- r4.2*M4.DEEPOCE # deepoc -> surfoc
    F4.5 <- r4.5*M4.DEEPOCE # deepoc -> sed
    F5.7 <- r5.7*M5.SEDIM   # sed    -> rocks
    F6.1 <- r6.1*M6.LANDBIO # landbio -> atm
    F6.2 <- r6.2*M6.LANDBIO # landbio -> surfoc
    F7.2 <- r7.2*M7.ROCKFOS # rocks    -> surface ocean
```

```

# mass balances:
dM1.ATMOSPHERE <- F2.1 + F6.1 - F1.2 - F1.6
dM2.SURFACEOCEAN <- F1.2 + F3.2 + F4.2 + F6.2 + F7.2 - F2.1 - F2.3 - F2.4
dM3.MARINEBIOTA <- F2.3 - F3.2 - F3.4
dM4.DEEPOCEAN <- F2.4 + F3.4 - F4.2 - F4.5
dM5.SEDIMENTOCEAN <- F4.5 - F5.7
dM6.LANDBIOTA <- F1.6 - F6.1 - F6.2
dM7.ROCKS_FOSSIL <- F5.7 - F7.2

# output time derivatives [PgC/yr]
list(c(dM1.ATMOSPHERE, dM2.SURFACEOCEAN, dM3.MARINEBIOTA, dM4.DEEPOCEAN,
      dM5.SEDIMENTOCEAN, dM6.LANDBIOTA, dM7.ROCKS_FOSSIL))
})
} # end of the model function

```

## The Jacobian matrix

The elements of the Jacobian matrix can rather easily be calculated based on the differential equations 1. For example, the differential equation for the state variable  $M_1$  is

$$\frac{dM_1}{dt} = f_1(M_1, \dots, M_7), \quad \text{where} \quad f_1(M_1, \dots, M_7) = -(r_{12} + r_{16}) \cdot M_1 + r_{21} \cdot M_2 + r_{61} \cdot M_6.$$

Thus, by applying the definition of the Jacobian matrix (see Eq. 7c in part I), we can calculate the *first* row as follows:

$$\begin{aligned}
J_{1,\cdot} &= \left[ \frac{\partial f_1}{\partial M_1}, \frac{\partial f_1}{\partial M_2}, \frac{\partial f_1}{\partial M_3}, \frac{\partial f_1}{\partial M_4}, \frac{\partial f_1}{\partial M_5}, \frac{\partial f_1}{\partial M_6}, \frac{\partial f_1}{\partial M_7} \right] \\
&= [-(r_{12} + r_{16}), r_{21}, 0, 0, 0, r_{61}, 0].
\end{aligned}$$

When we perform similar derivations for the remaining rows (state variables), we obtain the following expression for the Jacobian matrix:

$$J = \begin{bmatrix}
-(r_{12} + r_{16}) & +r_{21} & 0 & 0 & 0 & +r_{61} & 0 \\
r_{12} & -(r_{21} + r_{23} + r_{24}) & +r_{32} & +r_{42} & 0 & +r_{62} & +r_{72} \\
0 & r_{23} & -(r_{32} + r_{34}) & 0 & 0 & 0 & 0 \\
0 & r_{24} & +r_{34} & -(r_{42} + r_{45}) & 0 & 0 & 0 \\
0 & 0 & 0 & +r_{45} & -r_{57} & 0 & 0 \\
r_{16} & 0 & 0 & 0 & 0 & -(r_{61} + r_{62}) & 0 \\
0 & 0 & 0 & 0 & r_{57} & 0 & -r_{72}
\end{bmatrix}. \quad (2)$$

The structure of the Jacobian matrix gives us an impression about the *interactions* among the system compartments. Specifically, the off-diagonal elements  $J_{i,j}$ ,  $i \neq j$ , describe the flux of carbon from the compartment  $j$  to the compartment  $i$ , whereas the diagonal elements  $J_{i,i}$  describe the carbon efflux from the compartment  $i$ . Note that, in this particular model, most of the non-diagonal elements are zero, and the sums of the elements in each column are zero. The latter condition is related to the fact that our model describes a *closed* system.

Based on this result, we can now implement the Jacobian matrix in R. Note that we use the vector of model parameters `parms` when filling the elements of the matrix.

```
J <- with (as.list(parms),
  matrix(nrow = 7, ncol = 7, byrow = TRUE, data = c(
    -r1.2-r1.6, r2.1, 0, 0, 0, r6.1, 0,
    r1.2, -r2.1-r2.3-r2.4, r3.2, r4.2, 0, r6.2, r7.2,
    0, r2.3, -r3.2-r3.4, 0, 0, 0, 0,
    0, r2.4, r3.4, -r4.2-r4.5, 0, 0, 0,
    0, 0, 0, r4.5, -r5.7, 0, 0,
    r1.6, 0, 0, 0, 0, -r6.1-r6.2, 0,
    0, 0, 0, 0, r5.7, 0, -r7.2
  ))
)
```

## Characteristic time scales

### Residence time

An important parameter characterising a specific *compartment* of the system is the so-called *residence time*. Residence time describes a characteristic time scale in which a substance would leave the compartment if it was not replenished. If the total *efflux* of the substance from the compartment is  $F$  (a value in  $\text{mol s}^{-1}$ ) and the size of the compartment is  $S$  (a value in  $\text{mol}$ ), then the residence time is calculated as  $\tau_{\text{resid}} = S/F$  (a value in  $\text{s}$ ). For linear models such as the global C cycle model analysed here, the efflux is described by the first-order kinetics with respect to the size of the source compartment,  $F = r \cdot S$ , where  $r$  is the rate constant. Thus, for such a simple rate expression, the residence time is calculated as the *inverse* of the rate constant:  $\tau_{\text{resid}} = S/F = 1/r$ . If the total efflux comprises several components, e.g.,  $F = r_1 \cdot S + r_2 \cdot S + \dots$ , then the residence time is equal to the inverse of the *sum* of the rate constants,  $\tau_{\text{resid}} = 1/(r_1 + r_2 + \dots)$ .

The residence time of carbon in each compartment of the global C cycle model can easily be calculated from the Jacobian matrix. As mentioned above, the efflux from the compartment  $i$  is described by the *diagonal* element  $J_{i,i}$ . Thus, the corresponding residence time is calculated as  $\tau_{\text{resid},i} = -1/J_{i,i}$ .<sup>1</sup>

```
# rate constants in 1/yr, residence times in yr
t.resid <- -1/diag(J) # "-1" because effluxes are negative!
names(t.resid) <- names(state.eq)
sort(t.resid)

## M3.MARBIOT M2.SURFOCE M1.ATMOSPH M6.LANDBIO M4.DEEPOCE M5.SEDIM
## 6.000000e-02 4.269450e+00 4.531490e+00 3.833333e+01 3.666008e+02 7.500000e+02
## M7.ROCKFOS
## 3.750550e+08
```

The results above show that the residence time of carbon is only about 0.06 years (22 days) for the marine biota, while it is about 4 years for the surface ocean and the atmosphere, 38 years for the terrestrial biota, 370 years for the deep ocean, 750 years for the marine sediments, and 375 million years for the rocks and fossils. These values illustrate how short-lived some compartments would be if they were not continuously replenished by carbon influx from other compartments.

### Response time

Another interesting property characterising systems is the so-called *response time*. In contrast to the residence time, which characterises each individual compartment, the response time characterises the system as a whole, i.e., it accounts for the fact that the compartments interact with each other via influxes and effluxes. Specifically, the response time is a characteristic time scale in which the system responds to a perturbation.

The analysis in Part I of this Reader provides the theoretical basis for calculating the response time.

<sup>1</sup>The minus sign is used to ensure that the residence time is a positive number if  $J_{i,i} < 0$ .

As concluded there, the time evolution of a perturbed system can be described as a *sum* of components, each varying in time *independently* from the other components. Specifically, the time evolution of the  $i$ th component is described by an exponential function  $e^{\lambda_i t}$ , where  $\lambda_i$  is the  $i$ th eigen value of the Jacobian matrix. Therefore, the response times of the system are calculated from the eigen values of the Jacobian matrix:  $\tau_{resp,i} = -1/\lambda_i$ .

To calculate the response times of the global C cycle, we use the function `eigen` available in R, which returns both the eigen values and eigen vectors.

```
eig      <- eigen(J)
lambda   <- eig$values           # eigen values, 1/yr
V        <- eig$vectors          # eigen-vectors are columns of V!
t.resp   <- 1/abs(lambda)        # response time, yr
t.resp[t.resp>1e10] <- Inf       # replace the large value (1/0) by Inf
knitr::kable(data.frame(t.resp=t(round(t.resp,2))))
```

t.resp.1	t.resp.2	t.resp.3	t.resp.4	t.resp.5	t.resp.6	t.resp.7
0.06	3.24	8.3	96.49	750	204324.2	Inf

The output shows that the system responds on *multiple* time scales. They can be roughly grouped into (i) a very rapid response (0.06 years, corresponding to 22 days), (ii) rapid response (about 3 and 8 years), (iii) long response (about 100 and 750 years), and (iv) very long response (about 200,000 years). The last value is very large because it corresponds to the eigen value of 0. It was therefore set to infinity (1/0). The fact that one of the eigen values is 0 shows that the system will eventually reach a steady state, i.e., its “response” time will be infinite.

## Amounts of C exchanged on different time scales

In addition to the response time scales, the eigen vector / eigen value analysis additionally allows us to estimate the *amount* of C that is *exchanged* within the system on the particular time scale. This follows from the mathematical analysis presented in Part I of this Reader: the amounts of exchanged C are given by the *elements* of the vector  $\vec{\gamma}_{ini}$ . This vector is calculated by multiplying the perturbation vector  $\vec{\delta}_{ini}$  with the inverse of the matrix  $V$  constructed from the eigen vectors of the Jacobian matrix ( $\vec{\gamma}_{ini} = V^{-1} \cdot \vec{\delta}_{ini}$ ; see Eq. 15c in Part I).

To illustrate this, we consider a perturbation by deforestation where 100 Pg C was transferred from the terrestrial biota ( $M_6$ ) to the atmosphere ( $M_1$ ). This corresponds to the following value of the initial perturbation vector (see Eq. 13b in Part I):

$$\vec{\delta}_{ini} = [+100, 0, 0, 0, 0, -100, 0]^T.$$

Here, the notation  $[\cdot]^T$  indicates that  $\vec{\delta}_{ini}$  is a *column* vector. By multiplying this vector with the matrix  $V^{-1}$ , we obtain the following result for the vector  $\vec{\gamma}_{ini}$ :

```
# initial perturbation vector
delta.ini <- matrix(c(100, 0, 0, 0, 0, -100, 0), ncol=1) # column vector
gamma.ini <- solve(V) %*% delta.ini                    # matrix-vector multiplication

# display delta.ini, t.resp and gamma.ini together
knitr::kable(data.frame(delta.ini = delta.ini,
                        responseTime = round(t.resp,2),
                        exchangedC = round(gamma.ini,2)))
```

delta.ini	responseTime	exchangedC
100	0.06	0.00
0	3.24	-96.58
0	8.30	110.88
0	96.49	-52.40
0	750.00	-0.03
-100	204324.22	0.03
0	Inf	0.00

This result shows that although the response time scales range from 0.06 yr to about 200,000 yr, *significant* amounts of C are only exchanged on *three* time scales: 3.2 years (about 97 *Pg C* is exchanged), 8.2 years (about 111 *Pg C* is exchanged), and 96.5 years (about 52 *Pg C* is exchanged). The amounts of C exchanged on the other time scales are negligible ( $< 0.03$  *Pg C*). This is one example of a new insight that can be gained from the analysis based on the Jacobian matrix.

## Response of the system to a perturbation

The eigen value / eigen vector analysis can go one step further: it can reveal *which compartments* are involved in the exchange of a particular amount of carbon on a given time scale. Or, taken from a different angle, it can reveal the *amounts of carbon* that *leave or enter* a specific compartment on different time scales.

This information is contained in the matrix of amplitudes  $a_{j,i}$  (see Eq. 15b in Part I). Specifically,  $a_{j,i}$  represents the amount of carbon that *leaves*<sup>2</sup> the compartment  $j$  on the characteristic time scale  $\tau_{resp,i}$ . We store these values in a matrix **a** ( $a[j,i]$  corresponds to  $a_{j,i}$ ):

```
a <- matrix(0, nrow=7, ncol=7) # initialize 7x7 matrix with zeros
for(j in 1:7)
  for(i in 1:7) a[j,i] <- V[j,i]*gamma.ini[i]
```

Now, we display the second column of the matrix **a** to identify the compartments involved in the exchange of 97 *Pg C*, which occurs on the time scale of 3.2 years:

```
tab <- data.frame(t(a[,2])) # 2nd column, to be displayed as a row
# replace column names by shorter versions for better display
names(tab) <- names.short <- strtrim(names(state.eq),7)
knitr::kable(round(tab,2))
```

M1.ATMO	M2.SURF	M3.MARB	M4.DEEP	M5.SEDI	M6.LAND	M7.ROCK
63.93	-64.43	-0.22	23.7	0	-22.97	0

This result shows that the exchange on the time scale of 3.2 years mostly occurs between the atmosphere, the deep ocean, the surface ocean, and the land biota. More specifically, while the first two are carbon *sources*, the latter two are carbon *sinks*: about 64 *Pg C* and 24 *Pg C* are transferred *from* the atmosphere and the deep ocean, whereas about 64 *Pg C* and 23 *Pg C* are transferred *into* the surface ocean and the land biota, respectively.

Now, we display the entire matrix **a** to reveal the complete picture about the amounts of carbon exchanged between the compartments on the different time scales. We display the transpose of the matrix so that each row in the table corresponds to a different characteristic time scale.

```
tab <- data.frame(t.resp, gamma.ini, t(a)) # include transpose of "a"
names(tab) <- c("timescale", "Cexchange", names.short)
tab <- rbind(tab, c(NA, NA, rowSums(a)))
rownames(tab) <- c(paste("t", 1:7, sep=""), "total")
```

<sup>2</sup>Or *enters* the compartment, if  $a_{j,i} < 0$ .

```
knitr::kable(round(tab,2))
```

	timescale	Cexchange	M1.ATMO	M2.SURF	M3.MARB	M4.DEEP	M5.SEDI	M6.LAND	M7.ROCK
t1	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
t2	3.24	-96.58	63.93	-64.43	-0.22	23.70	0.00	-22.97	0.00
t3	8.30	110.88	41.13	67.19	0.23	-64.22	0.00	-44.32	0.00
t4	96.49	-52.40	-5.06	-2.75	-0.01	40.54	-0.02	-32.70	0.00
t5	750.00	-0.03	0.00	0.00	0.00	0.00	0.02	0.00	-0.02
t6	204324.22	0.03	0.00	0.00	0.00	-0.02	0.00	0.00	0.02
t7	Inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
total	NA	NA	100.00	0.00	0.00	0.00	0.00	-100.00	0.00

The interpretation of the second row in this table is given above. The rest of the table is interpreted in a similar way:

- About 111 *Pg C* is exchanged within the system on the characteristic time scale of 8.3 yr; this includes a transfer of about 41 *Pg C* and 67 *Pg C* from the atmosphere and the surface ocean, and a transfer of about 64 *Pg C* and 44 *Pg C* into the deep ocean and the land biota, respectively (row 3).
- About 52 *Pg C* is exchanged within the system on the characteristic time scale of 96.5 yr; this time, the deep ocean is the only *source* (about 40.5 *Pg C*), while the terrestrial biota (32.7 *Pg C*), the atmosphere (5 *Pg C*), and the surface ocean (2.75 *Pg C*) are *sinks* (row 4).
- The amounts of C exchanged on the time scales 0.06, 750 and 200,000 years are insignificant (row 1, 5 and 6).

Overall, this analysis reveals that, with respect to the amounts of the exchanged C, the perturbation by deforestation mostly involves four compartments: the atmosphere, the surface ocean, the deep ocean and the terrestrial biota. Some of them can switch the role between a sink and a source depending on the time scale. Moreover, the effects of the perturbation “decay away” during the first 300 years,<sup>3</sup> and they become negligible afterwards.<sup>4</sup> This is another example of a new insight that can be gained from the analysis based on the Jacobian matrix.

## Graphical illustration

### Characteristic timescales of total C exchanged

To illustrate the above conclusions graphically, we show how the *amounts of exchanged C* vary in time. Additionally, we show how the *perturbation* of each compartment varies in time, and how this variation can be *decomposed* into components varying on different characteristic time scales.

First, we perform the calculations based on equations 15a–c in Part I. Note that, for each compartment, the decomposition of the perturbation into components varying on different time scales is stored as a *list* of matrices, where the first and second dimension of each matrix represents the total time (`tseq`) and the characteristic time scale of the variation (`TSsel`), respectively.

```
TSsel <- 1:7 # consider all characteristic time scales, or
#TSsel <- c(2,3,4) # only those on which signif. amounts of C are exchanged
NTS <- length(TSsel) # number of time scales considered
NC <- length(state.eq) # total number of compartments

# use exponentially increasing time steps to capture all time scales
Nt <- 1000 # number of time points considered
tseq <- c(0, 0.01*exp(seq(0, 18, length.out = Nt-1)))
```

<sup>3</sup>Here we applied a useful “rule of thumb”: for a response time of  $\tau_{resp}$ , the perturbation decreases to less than 5% of the initial value at times above  $3 \cdot \tau_{resp}$ . This is because the perturbation varies in time as  $e^{-t/\tau_{resp}}$ , and because  $e^{-3} \approx 0.05$ .

<sup>4</sup>This statement is valid for the system as a whole, and may not necessarily be true for an individual compartment, especially for a small compartment such as the marine biota.

```

# calculate gamma.t for all time scales considered (NTS)
gamma.t <- matrix(data=0, nrow=Nt, ncol=NTS)
for (i in 1:NTS)
  gamma.t[,i] <- gamma.ini[TSsel[i]]*exp(lambda[TSsel[i]]*tseq) # exchanged C

# calculate DeltaS.t for all compartments
DeltaS.t <- list() # will have one matrix per compartment
DeltaS.j.t <- matrix(nrow=Nt, ncol=NC, data=0) # total perturbation

pert.t <- matrix(nrow=Nt, ncol=NTS, data=0) # dynamics for each timescale
for (j in 1:NC){
  # loop over compartments
  for (i in 1:NTS)
    # loop over time scale
    pert.t[,i] <- a[j,TSsel[i]]*exp(lambda[TSsel[i]]*tseq) # perturbation

  DeltaS.t[[j]] <- pert.t # keep it for each compartment
  # sum over all time scales to obtain total perturbation for compartment j
  DeltaS.j.t[,j] <- rowSums(pert.t) # Eq. 15a
}

```

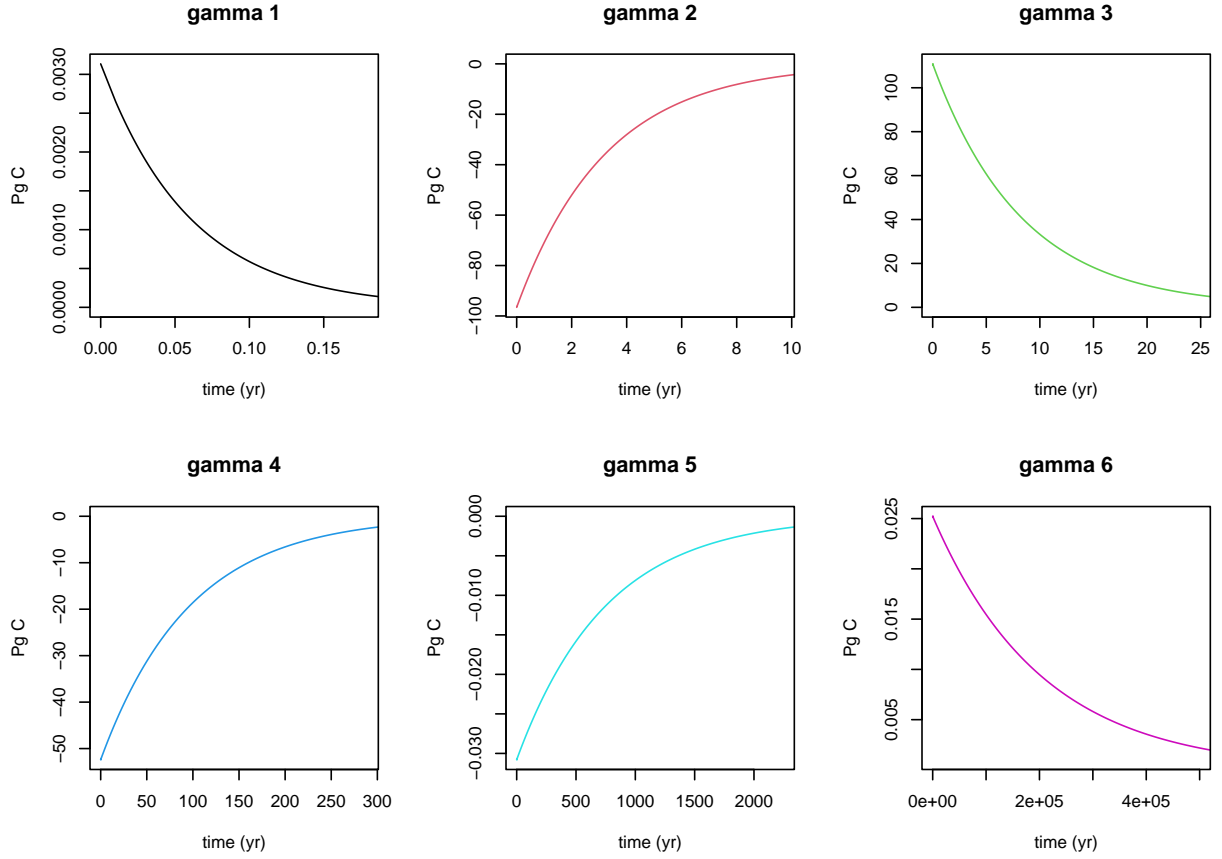
The graphs below show that although the amounts of exchanged carbon decrease in time on 6 characteristic time scales (note differences in the x-scale for each graph), the differences in the y-scale illustrate that most of the carbon is exchanged on characteristic time scales between 3 and 100 years, as mentioned above.

```

nfig <- min(NTS,6)
ifelse(nfig<4, par(mfrow=c(1,3)), par(mfrow=c(2,3)))
for(i in 1:nfig)
  plot(tseq, gamma.t[,i], type="l", lty=1, col=i,
       main=paste("gamma",i), xlab="time (yr)", ylab="Pg C",
       xlim=c( 0, min(3*t.resp[TSsel[i]],5e5) )
  )

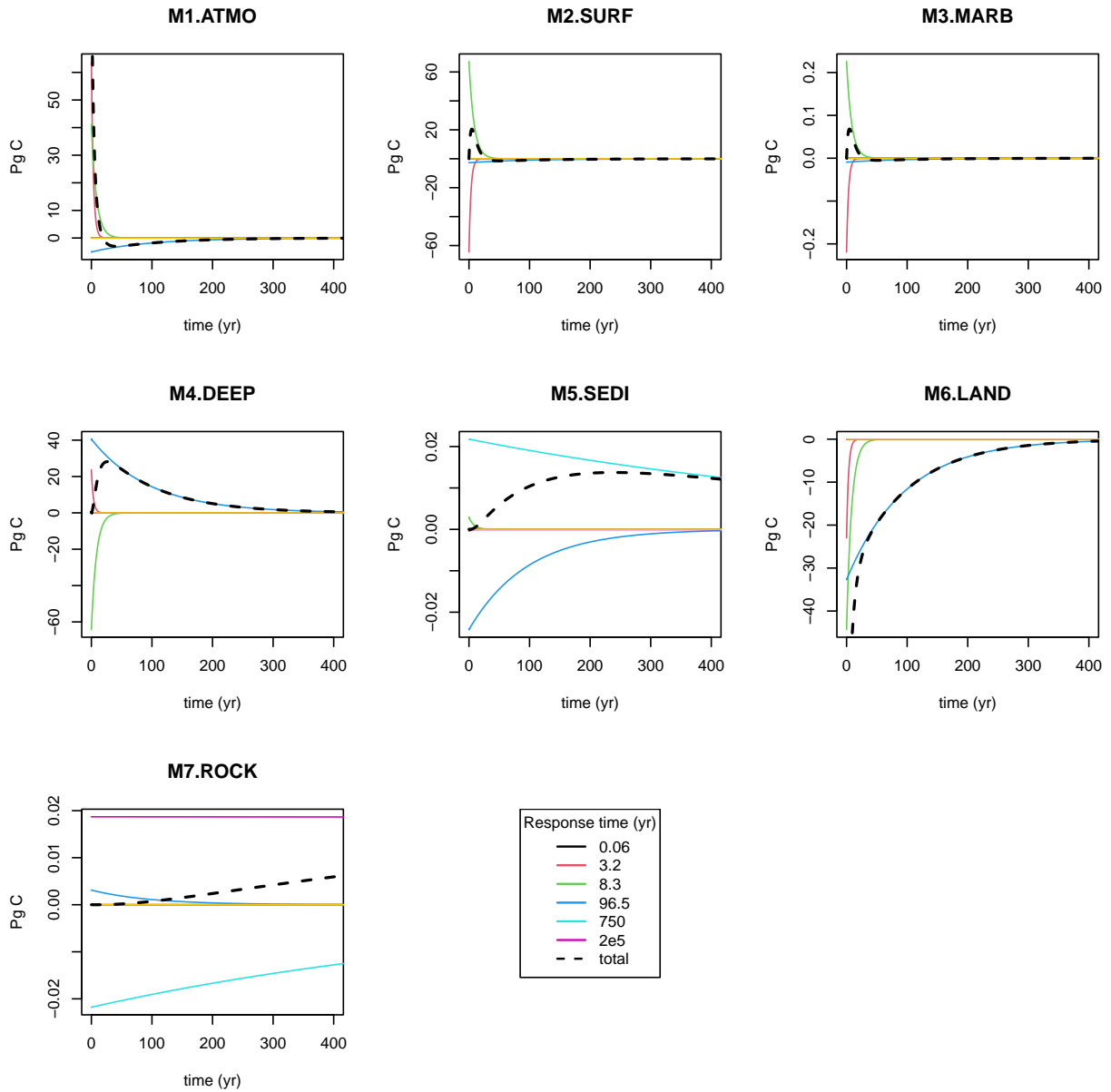
```





For the individual compartments, the graphs below show that the time variation of the *total* perturbation (dashed black line) can be decomposed into *independent components* that exponentially decay with time (colored lines). The characteristic time scales of this decay are equal to the response times,  $\tau_{resp,i}$ ,  $i = 1, \dots, 6$ . The graphs illustrate that after about 300 years, the amount of carbon exchanging among the compartments has decreased practically to zero, and thus the system has essentially returned back to the equilibrium. Additionally, a comparison of the y-scales of the individual graphs reveals that the perturbation by deforestation mostly affects only *four* compartments (the atmosphere, the surface ocean, the deep ocean and the terrestrial biota), as mentioned earlier.

```
par(mfrow=c(3,3))
for(j in 1:NC)
{
  matplot(tseq, DeltaS.t[[j]][,1:NTS], type="l", lwd=1, lty=1, xlim=c(0,400),
          col=TSsel, main=names.short[j], ylab="Pg C", xlab="time (yr)")
  lines(tseq, DeltaS.j.t[,j], lty=2, col=1, lwd=2, mfrow=NULL)
}
plot.new()
legend("top", legend=c("0.06", "3.2", "8.3", "96.5", "750", "2e5", "total"),
      col=c(1:6,1), lty=c(rep(1,6),2), lwd=1.5, title="Response time (yr)")
```



## Comparison between the numerical and analytical solutions

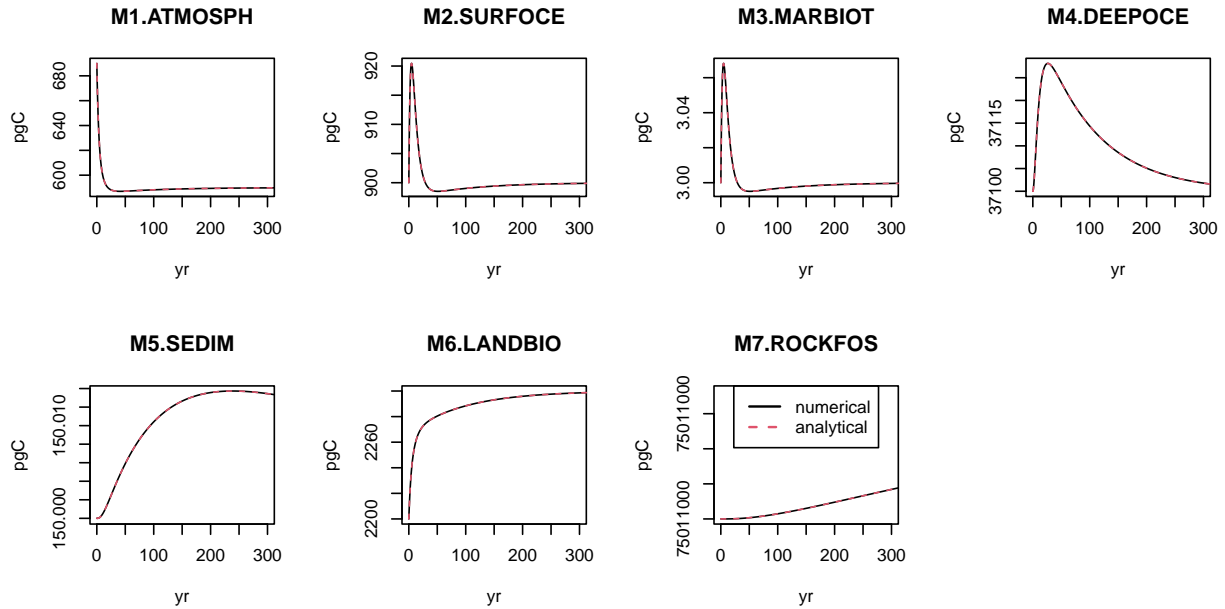
Finally, we use both model implementations to show that they indeed produce similar results:

```
# Numerical solution, determined by deSolve:
require(deSolve)
state <- state.eq + as.vector(delta.ini) # make delta.ini a row vector
out <- ode(y=state, times=tseq, func=Cmodel, parms=parms)

# Analytical perturbation, based on the Jacobian matrix analysis:
# Add total perturbation to the equilibrium state to calculate
# the C content for each state variable as a function of time.
state.t <- matrix(rep(1,Nt)) %*% state.eq + DeltaS.j.t
out.an <- data.frame(tseq, state.t)
```

```
names(out.an) <- c("time", names(state.eq)) # make the names equal

plot(out, ylab= "pgC", xlab="yr", lwd=1, mfrow=c(2,4), xlim=c(0,300),
      col=1, obs=out.an, obspar=list(type="l", col=2, lty=2))
legend("top", legend=c("numerical", "analytical"), col=1:2, lty=1:2, lwd=1.5)
```



## References

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

Karline Soetaert, Thomas Petzoldt, R. Woodrow Setzer (2010). Solving Differential Equations in R: Package deSolve. Journal of Statistical Software, 33(9), 1–25. URL <http://www.jstatsoft.org/v33/i09/> DOI 10.18637/jss.v033.i09