Air-water isotope equilibration model

How fast does the isotopic composition of vapour pressure in the atmosphere equilibrate with the isotopic composition of liquid water?

Lubos Polerecky

25-07-2021

Contents

Introduction

Empty

Model definition

```
# parameters describing the experimental setup
V3 <- 0.005 # air volume (m3)
P <- 101325 # air pressure (Pa)
Rgas <- 8.314 # gas constant (SI)
TK <- 273.15+20 # temperature (K)
fH2O <- 0.01 # molar fraction of water in air at 100% humidity
kf <- 1.4e3 # rate constant of water evaporation (mol m-2 d-1)
A1 <- pi*(6e-3)*12 # area of the water body 1 (m2); 3*4 subcores, each with inner diameter 1.2 cm
A2 <- 0.25*0.25 # area of the water body 2 (m2); square of 25x25 cm
h1 <- 0.002 # height of the water body 1 (m); height of the overlying water column 2 mm
h2 <- 0.003 # height of the water body 2 (m); water height in the sealed container 3 mm
xini <- 0.002 # natural abundance of 18O
x1ini <- 0.002 # initial 18O atom fraction in water body 1
x2ini <- xini # initial 18O atom fraction in water body 2
x3ini <- xini # initial 18O atom fraction of water vapor (water reservoir 3)
rho <- 1e3 # water density (kg m-3)
MW <- 18e-3 # molar weight of water (kg mol-1)

# calculated parameters
N3max <- V3*P/(Rgas*TK) * fH2O # max amount of H2O molecules in air at 100% humidity (mol)
kb <- kf/N3max # rate constant of water condensation (m-2 d-1)
```

```r
V1 <- A1*h1 # volume of water in reservoir 1 (m3)
V2 <- A2*h2 # volume of water in reservoir 2 (m3)
# the density of particles (N/V=rho/MW) is assumed to be the same for 16O-H2O and 18O-H2O!
N1ini <- rho*V1/MW # initial amount of H2O molecules in reservoir 1 (mol)
N2ini <- rho*V2/MW # initial amount of H2O molecules in reservoir 2 (mol)
N3ini <- N3max # initial amount of H2O molecules in reservoir 3 (mol)

# model parameters
pars <- c(A1 = A1,
          A2 = A2,
          kf_16 = kf*0.995,
          kf_18 = kf,
          kb_16 = kb*0.995,
          kb_18 = kb)

# Initial conditions of the state variables
SVini <- c(N1_16 = N1ini*(1-x1ini), # all in moles
           N1_18 = N1ini*   x1ini,
           N2_16 = N2ini*(1-x2ini),
           N2_18 = N2ini*   x2ini,
           N3_16 = N3ini*(1-x3ini),
           N3_18 = N3ini*   x3ini)

# Model function: calculates time-derivatives and other output
WaterExchangeModel <-function(t, state, pars) {
  # t: time, state: state variables, pars: model parameters
  with (as.list(c(state, pars)),{

  # current amounts of H2O molecules and isotope ratios
    N1 <- N1_16+N1_18
    N2 <- N2_16+N2_18
    N3 <- N3_16+N3_18
    x1 <- N1_18/N1
    x2 <- N2_18/N2
    x3 <- N3_18/N3

  # rate expressions [mol/d]
    Evap1_16 <- kf_16 * (1-x1) * A1 * (N1>0)
    Evap1_18 <- kf_18 *    x1  * A1 * (N1>0)
    Cond1_16 <- kb_16 * (1-x3) * A1 * N3
    Cond1_18 <- kb_18 *    x3  * A1 * N3
    Evap2_16 <- kf_16 * (1-x2) * A2 * (N2>0)
    Evap2_18 <- kf_18 *    x2  * A2 * (N2>0)
    Cond2_16 <- kb_16 * (1-x3) * A2 * N3
    Cond2_18 <- kb_18 *    x3  * A2 * N3

  # Time-derivatives: dN/dt = production - consumption [mol/d]
    f <- 0
    dN1_16.dt <- -Evap1_16 + Cond1_16
    dN1_18.dt <- -Evap1_18 + Cond1_18
    dN2_16.dt <- (-Evap2_16 + Cond2_16)*f
    dN2_18.dt <- (-Evap2_18 + Cond2_18)*f
    dN3_16.dt <-  Evap1_16 - Cond1_16 + (Evap2_16 - Cond2_16)*f
    dN3_18.dt <-  Evap1_18 - Cond1_18 + (Evap2_18 - Cond2_18)*f
```

```
    Ntot_16 <- N1_16+N2_16+N3_16
    Ntot_18 <- N1_18+N2_18+N3_18
    xtot    <- Ntot_18/(Ntot_16+Ntot_18)

  # return time-derivatives and ordinary variables as a list
    list(c(dN1_16.dt, dN1_18.dt,
           dN2_16.dt, dN2_18.dt,
           dN3_16.dt, dN3_18.dt),
      # other output
      Ntot_16 = Ntot_16,
      Ntot_18 = Ntot_18,
      x1 = x1,
      x2 = x2,
      x3 = x3,
      xtot = xtot,
      delta1 = (x1/xini-1)*1e3,
      delta2 = (x2/xini-1)*1e3,
      delta3 = (x3/xini-1)*1e3,
      delta31  = (x3/x1-1)*1e3
    )
  })
}
```
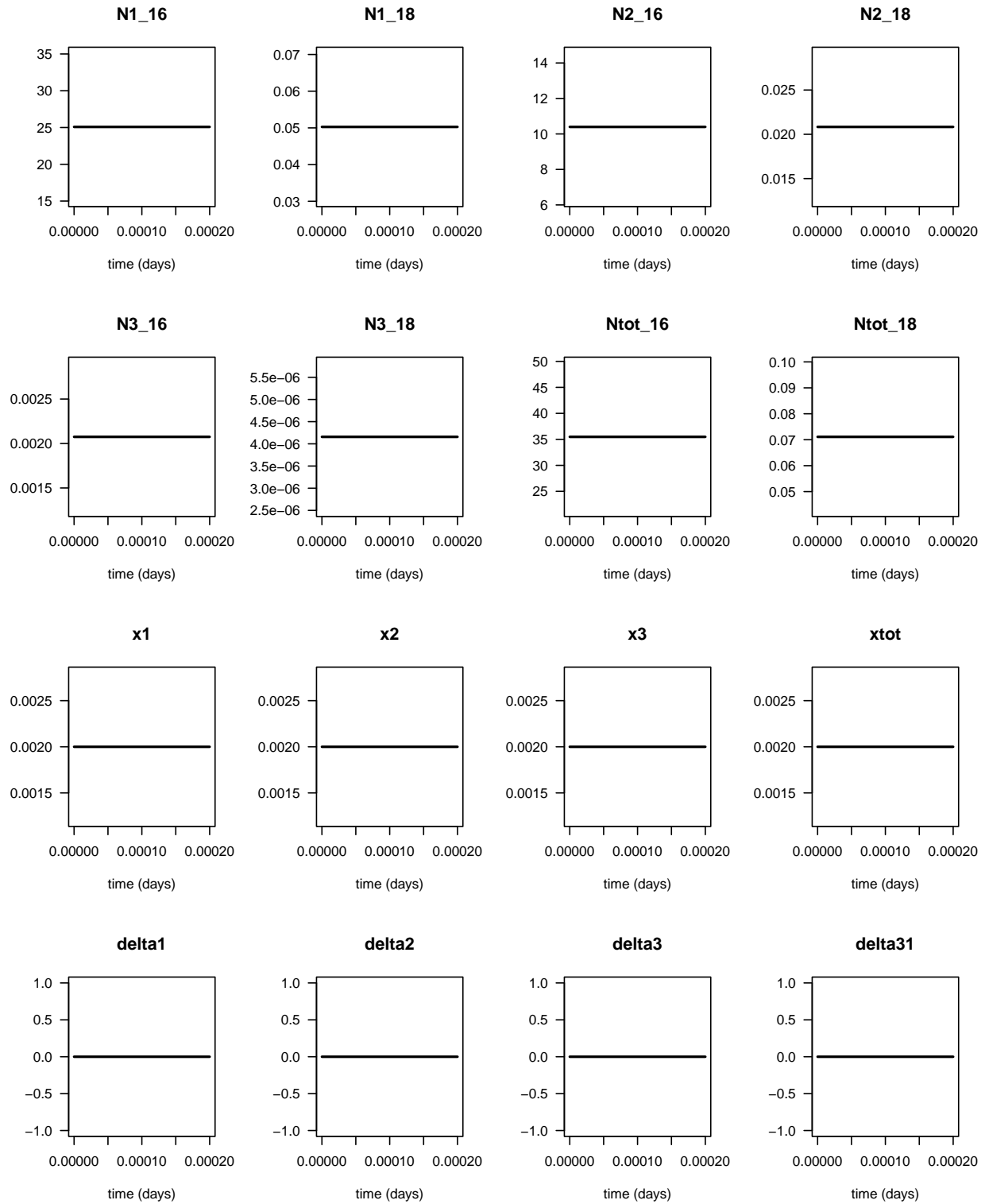
Model solution

Dynamic solution

```
require(deSolve)  # package with integration methods
outtimes <- seq(from = 0, to = 0.0002, length.out = 100)
out <- ode(y = SVini, parms = pars, func = WaterExchangeModel, times = outtimes)

plot(out, xlab="time (days)", las=1, lwd=2, mfrow=c(4,4))
```

N1_16

N1_18

N2_16

N2_18

N3_16

N3_18

Ntot_16

Ntot_18

x1

x2

x3

xtot

delta1

delta2

delta3

delta31


```
#      ylab=list("mol/m3","mol/m3","mol/m3/d","mol/m3/d"))
#legend("topright", legend = c("original", "10-fold lower"),
#      title="aeration rate constant:",
#      col=1:2, lwd=2, lty=1:2, bty="n")
```