Spatially-explicit model of cosmogenic nuclide production in sediments

Nicolas Gauthier and Nari Miller

October 20, 2016

```
library(magrittr)
library(raster)
library(ggplot2)
library(reshape2)
```

Model specification

We use the following model for P_z, the production rate of ^{10}Be at depth z:

$$P_z = P_0 e^{-z\frac{L}{p}} - N\lambda$$

with parameters:

P_0 is the production rate of Beryllium-10 in quartz at sea level

```
P_0 <- 4.49     # [at/g/yr] after Stone, 1999.
```

λ is the decay constant for ^{10}Be

```
ltlambda <- log(2) / 1.5e6
```

L is the absorption mean-free path (attenuation length)

```
L <- 160    # [g/cm2]
```

p is the density of overburden

```
p <- 2.6    # [ g/cm3]
```

and variables:

N is the concentration of nuclides in the sample
z is the depth to that packet of sediment at time t
t is some length of time.

Assumptions

For the sake of simplicity, we assume there is no topographic shielding (topographic shielding factor = 1) and a constant location in the Mediterranean at 40N 0E.

Sample data

First we need base set of raster maps to initialize the variables.
We create a multi-layer raster brick where each cell represents a 1 cm^3 packet of sediment, the **values()** of the cells correspond to N (the concentration of nuclides in that packet), and the index of each layer corresponds to $1 + z$, the depth of that packet of sediment.

Lets create a sample raster brick with 5 layers of 1x1 cells, with an initial value of $N = 1000$ for each cell.

```
rast <- matrix(1000) %>% raster
N0 <- brick(c(rast, rast, rast, rast, rast, rast, rast, rast, rast, rast))
```

Simulation

First translate the above formula for P_z into an R function that calculates the ^{10}Be production rate given values of N and z.

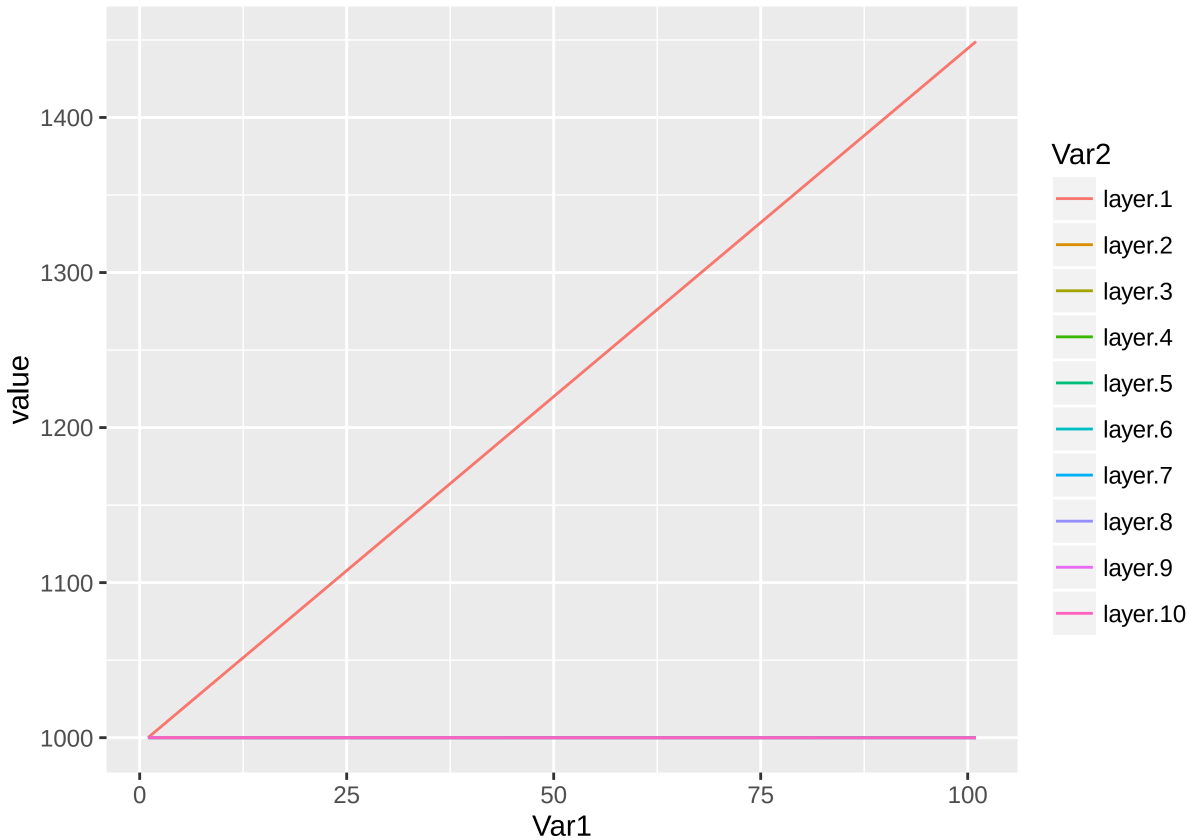
```
P_z <- function(N, z){
  P_0 * exp(-z*L/p) - N*ltlambda
}
```

Numerically integrate the differential equation with Euler's method. Using this function and the sample raster brick, iterate over a period of 100 years.

```
nsim <- 100 # simulation length

N <- N0 # initial conditions

record <- values(N) # vector to store the outputs of the simulation

for(i in 1:nsim){
  delta <- P_z(N, 1:nlayers(N) - 1)
  N <- N + delta
  record <- rbind(record, values(N))
}
```

Plot the resulting solution.

```
ggplot(melt(record), aes(x=Var1,y=value, color = Var2)) + geom_line()
```



Note that the sediment packets at depth are decaying with time, just at a rate small enough not to be visible when compared with the rate of change of the surface level.

Numerical integration

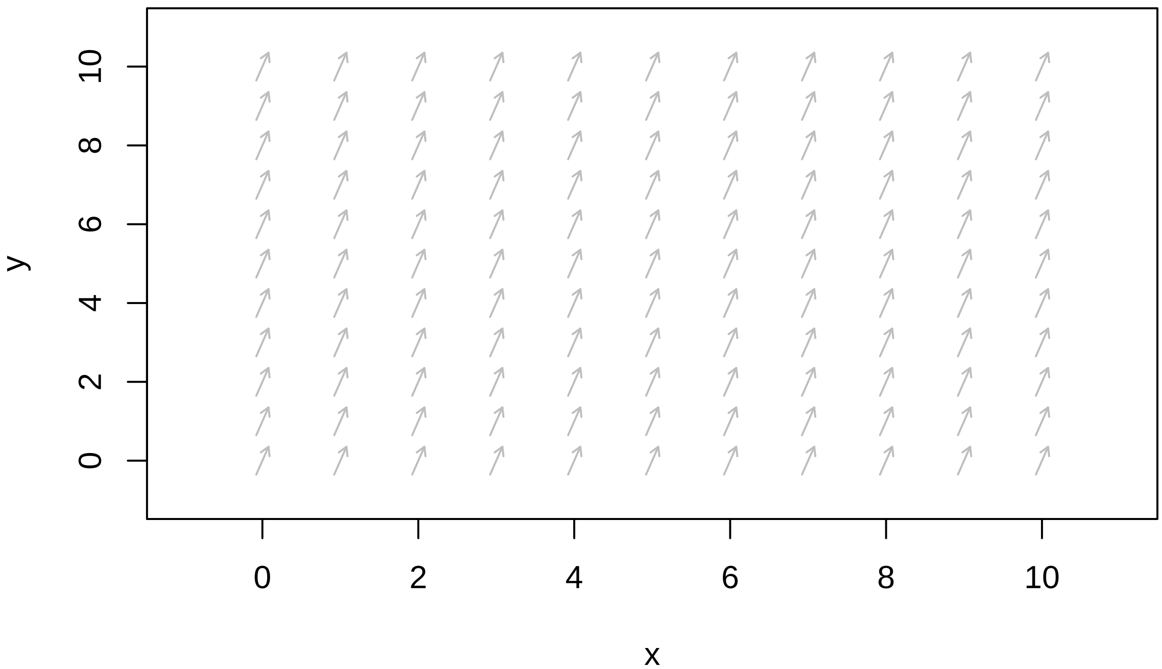
Maybe euler's method is introducing some errors, try a more advanced ODE solver? Still in progress . . .

Redifine the function to be deSolve compataible.

```
library(deSolve)
library(phaseR)

be10 <- function(t, y, parameters){
    z <- parameters
    dy <- P_0 * exp(-z*L/p) - y*ltlambda
    list(dy)
}
```

Phase plot at surface

```
#t <- seq(0,40,.5)
nuclide.flow <- flowField(be10, x.lim = c(0,10), y.lim = c(0,10),
                          system = 'one.dim', parameters = 0, add = F)
nuclide.null <- nullclines(be10, x.lim = c(0,10), y.lim = c(0,10),
                           system = 'one.dim', points = 200, parameters = 0)
```



Phase plot at depth

```
nuclide.flow <- flowField(be10, x.lim = c(0,10), y.lim = c(0,10),
                          system = 'one.dim', parameters = 5, add = F)
nuclide.null <- nullclines(be10, x.lim = c(0,10), y.lim = c(0,10),
                           system = 'one.dim', points = 200, parameters = 5)
```