Advection-diffusion test

Francis Kurian

10/6/2021

Unit testing R code with testthat

Two functions are given for using testthat library and functions: **advect** ,**diffuse**

```
setwd("C:/Users/jeoku/OneDrive/Documents/chapman/CS510/docs/AdvDiff/AdvDiff")
source("src/advect.R")
source("src/diffuse.R")

#### Defines parameters ####
D <- 1e-8        # Diffusion coefficient in m^2/s
#D <- 0        # Diffusion coefficient in m^2/s
delta.t <- 1e-5  # Time step size in s
end.time <- 0.01    # End simulation time in s
start.x <- 5.6
start.y <- -0.97

# Creates points to follow
num.dots <- 10
dotsx <- rep(start.x, num.dots)
dotsy <- rep(start.y, num.dots)
dots.start <- matrix(c(dotsx, dotsy), num.dots, 2)

# Reads in position (x,y) and velocity (Ux,Uy) data
x <- as.matrix(read.table("data/x.csv", header = FALSE, sep = ","))
y <- as.matrix(read.table("data/y.csv", header = FALSE, sep = ","))
Ux <- as.matrix(read.table("data/Ux.csv", header = FALSE, sep = ","))
Uy <- as.matrix(read.table("data/Uy.csv", header = FALSE, sep = ","))
Ux[is.na(Ux)] <- 0
Uy[is.na(Uy)] <- 0

t <- 0
dots <- dots.start  # re-assigning the start matrix to dots
```

Initial dots dataset:

```
##       [,1]  [,2]
##  [1,]  5.6 -0.97
##  [2,]  5.6 -0.97
##  [3,]  5.6 -0.97
##  [4,]  5.6 -0.97
##  [5,]  5.6 -0.97
##  [6,]  5.6 -0.97
##  [7,]  5.6 -0.97
##  [8,]  5.6 -0.97
##  [9,]  5.6 -0.97
## [10,]  5.6 -0.97
```

1. Ensures that points that have no diffusion move together.

set the Diffusion Coefficient to 0. Results should exactly match.

Argument 1: sum(advect(dots, x, y, Ux, Uy, 0.5*delta.t)) # advect with initial dots dataset Argument 2: sum(diffuse((advect(dots, x, y, Ux, Uy, 0.5*delta.t)), 0, delta.t)) #advect with diffusion coeficient D=0 Pass test with D=0

```
library("testthat")
sum(advect(dots, x, y, Ux, Uy, 0.5*delta.t)) # advect with initial dots dataset
```

```
## [1] 46.3
```

```
sum(diffuse((advect(dots, x, y, Ux, Uy, 0.5*delta.t)), 0, delta.t)) #advect with diffusion coeficient D
```

```
## [1] 46.3
```

```
test_that("No Diffusion", {expect_equal(sum(advect(dots, x, y, Ux, Uy, 0.5*delta.t)),
                                   sum(diffuse((advect(dots, x, y, Ux, Uy, 0.5*delta.t)), 0, delta.t)))
```

```
## Test passed
```

2.The square of the mean displacement of molecules diffusing without advection is close to 4*D*delta.t.

Use diffuse function on initial dots dataset. Do not advect. The square of the mean displacement of molecules diffusing without advection will be close to 4*D*delta.t

```
dots <- dots.start
dots4 <- diffuse(dots, D, delta.t)
4*D*delta.t #benchmark to compare
```

```
## [1] 4e-13
```

```
mean(dots4[,2]-dots4[1,2])**2 # expected to match with 4*D*Delta.t
```

```
## [1] 5.765486e-14
```

```
test_that("Mean Square Displacement", {expect_equal(4*D*delta.t,(mean(dots4[,2]-dots4[1,2])**2))})
```

```
## Test passed
```