1. Slurry mass tests using `abm_variable()`

Sasha D. Hafner

24 April, 2025 05:41

Overview

Point of these tests is to check behavior of `abm()` when slurry mass is variable. Development of the relevant code has been challenging, so it is important to check behavior of multiple scenarios after an update.

Prep

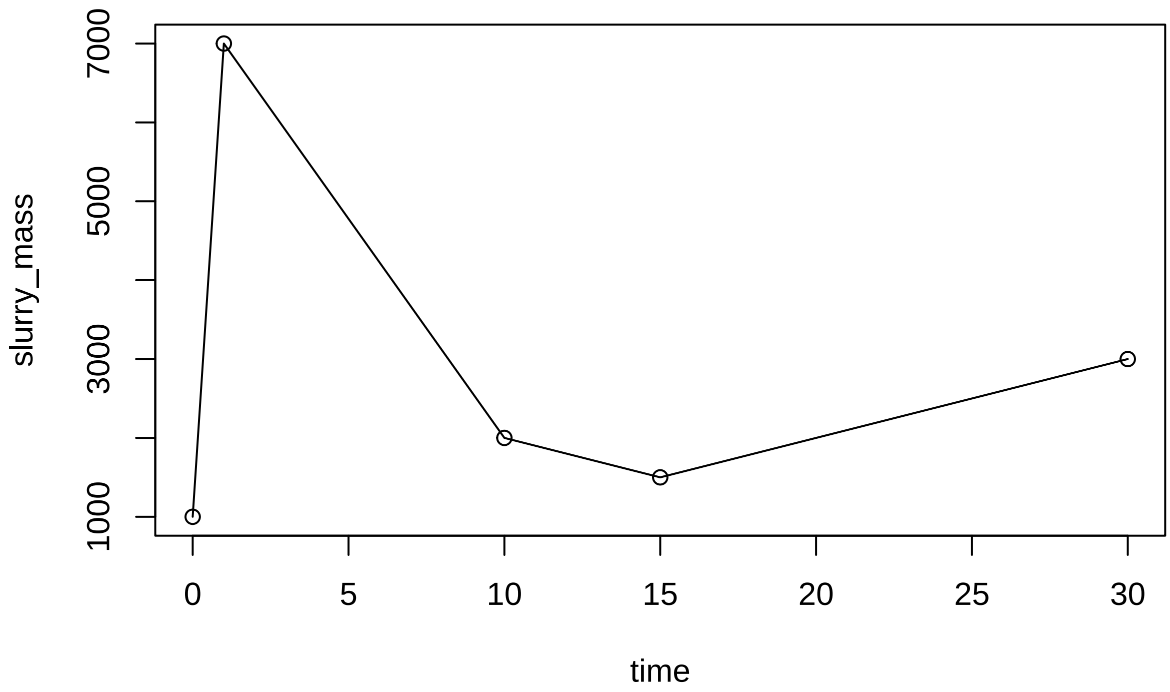
```
devtools::load_all()
```

```
## i Loading ABM
## Loading required package: deSolve
```

Case 1, consecutive removals

Slurry mass data.

```
slurry_mass_dat <- data.frame(time = c(0, 1, 10, 15, 30), slurry_mass = c(1000, 7000, 2000, 1500, 3000)
plot(slurry_mass ~ time, data = slurry_mass_dat, type = 'o')
```



Default, with “early” behavior.

```
out1 <- abm(30, delta_t = 0.1, man_pars = man_pars1.0, grp_pars = grp_pars1.0,
            add_pars = list(storage_depth = 4, area = 1000, floor_area = 0,
                            slurry_mass = slurry_mass_dat, evap = 0, rain = 0))
```

```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.

## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int

## rain = 0 kg/m2/day

## evaporation = 0 kg/m2/day
```

Late

```
out2 <- abm(30, delta_t = 0.1, man_pars = man_pars1.0, grp_pars = grp_pars1.0,
            add_pars = list(storage_depth = 4, area = 1000, floor_area = 0,
                            slurry_mass = slurry_mass_dat, evap = 0, rain = 0),
            approx_method = c(temp = 'linear', pH = 'linear', slurry_mass = 'late'))
```

```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.

## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int

## rain = 0 kg/m2/day

## evaporation = 0 kg/m2/day
```

And mid

```
out3 <- abm(30, delta_t = 0.1, man_pars = man_pars1.0, grp_pars = grp_pars1.0,
            add_pars = list(storage_depth = 4, area = 1000, floor_area = 0,
                            slurry_mass = slurry_mass_dat, evap = 0, rain = 0),
            approx_method = c(temp = 'linear', pH = 'linear', slurry_mass = 'mid'))
```

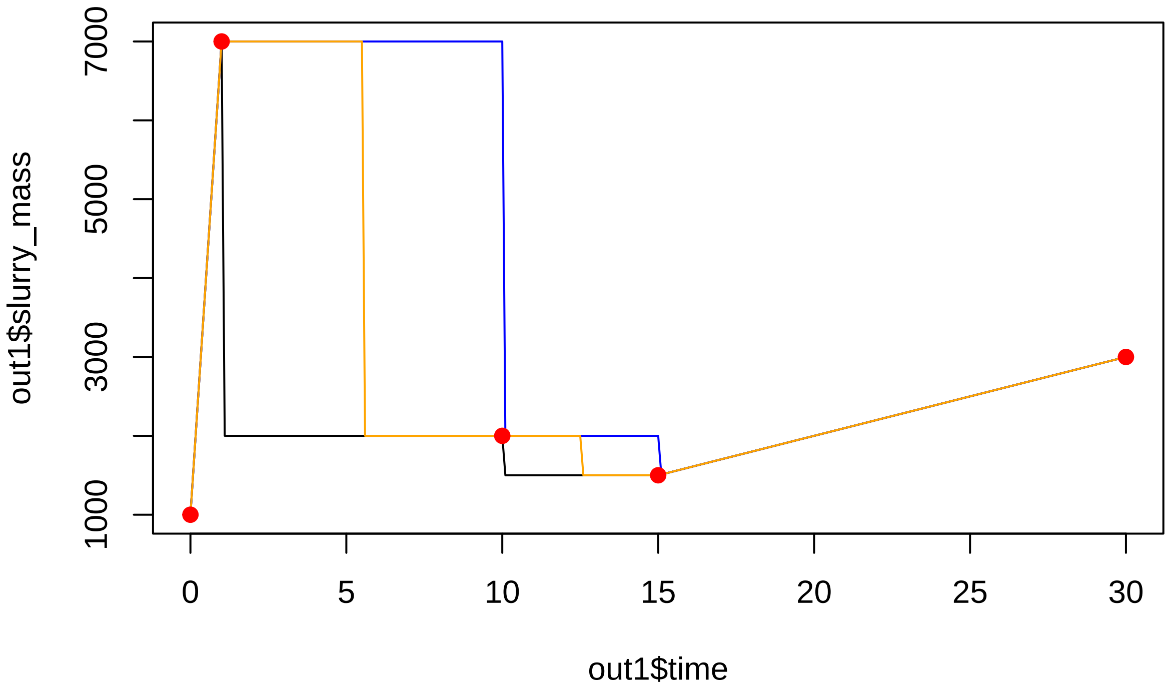
```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.

## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int

## rain = 0 kg/m2/day

## evaporation = 0 kg/m2/day
```

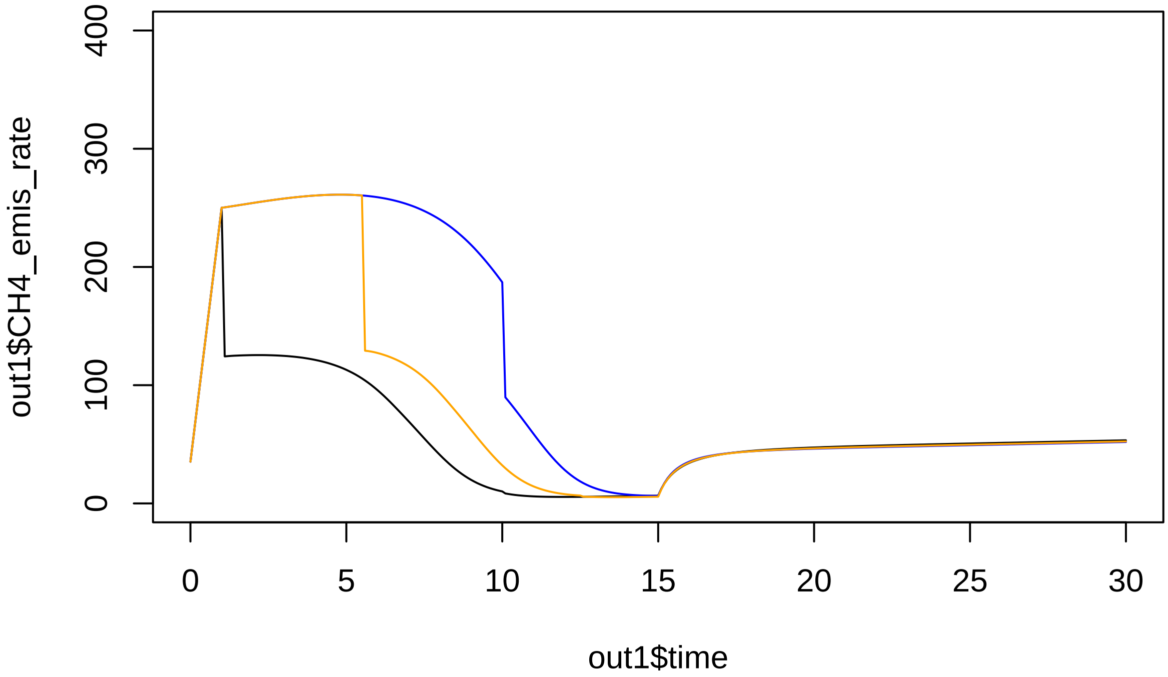
```
plot(out1$time, out1$slurry_mass, type = 'l')
lines(out2$time, out2$slurry_mass, type = 'l', col = 'blue')
lines(out3$time, out3$slurry_mass, type = 'l', col = 'orange')
points(slurry_mass_dat$time, slurry_mass_dat$slurry_mass, col = 'red', pch = 19)
```



Expect that all three approaches hit the red points (input level) exactly.

Methane production rate:

```
plot(out1$time, out1$CH4_emis_rate, type = 'l', ylim = c(0, 400))
lines(out2$time, out2$CH4_emis_rate, type = 'l', col = 'blue')
lines(out3$time, out3$CH4_emis_rate, type = 'l', col = 'orange')
```



Case 2, slurry removal at the beginning

```
slurry_mass_dat <- data.frame(time = c(0, 1, 10, 15, 30),
                              slurry_mass = c(1000, 500, 2000, 1500, 3000))

out4 <- abm(30, delta_t = 0.1, man_pars = man_pars1.0, grp_pars = grp_pars1.0,
            add_pars = list(storage_depth = 4, area = 1000, floor_area = 0,
                            slurry_mass = slurry_mass_dat, evap = 0, rain = 0))
```

```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.

## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int

## rain = 0 kg/m2/day

## evaporation = 0 kg/m2/day
```

```
out5 <- abm(30, delta_t = 0.1, man_pars = man_pars1.0, grp_pars = grp_pars1.0,
            add_pars = list(storage_depth = 4, area = 1000, floor_area = 0,
                            slurry_mass = slurry_mass_dat, evap = 0, rain = 0),
            approx_method = c(temp = 'linear', pH = 'linear', slurry_mass = 'late'))
```

```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.

## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int
## rain = 0 kg/m2/day
```

```
## evaporation = 0 kg/m2/day
```

```
out6 <- abm(30, delta_t = 0.1, man_pars = man_pars1.0, grp_pars = grp_pars1.0,
            add_pars = list(storage_depth = 4, area = 1000, floor_area = 0,
                            slurry_mass = slurry_mass_dat, evap = 0, rain = 0),
            approx_method = c(temp = 'linear', pH = 'linear', slurry_mass = 'mid'))
```

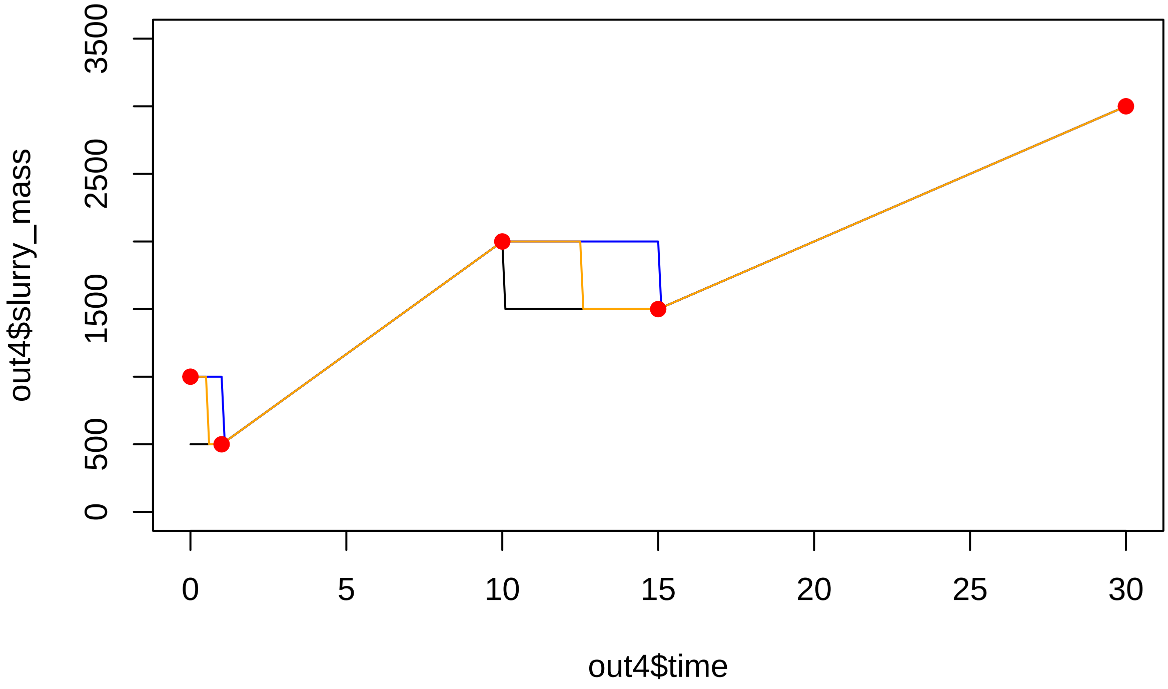
```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.
```

```
## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int
```

```
## rain = 0 kg/m2/day
```

```
## evaporation = 0 kg/m2/day
```

```
plot(out4$time, out4$slurry_mass, type = 'l', ylim = c(0, 3500))
lines(out5$time, out5$slurry_mass, type = 'l', col = 'blue')
lines(out6$time, out6$slurry_mass, type = 'l', col = 'orange')
points(slurry_mass_dat$time, slurry_mass_dat$slurry_mass, col = 'red', pch = 19)
```



Case 3, slurry removal at end

```
slurry_mass_dat <- data.frame(time = c(0, 1, 10, 15, 30),
                              slurry_mass = c(1000, 2000, 4000, 4500, 1000))

out7 <- abm(30, delta_t = 0.1, man_pars = man_pars1.0, grp_pars = grp_pars1.0,
            add_pars = list(storage_depth = 4, area = 1000, floor_area = 0,
                            slurry_mass = slurry_mass_dat, evap = 0, rain = 0))
```

```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.
```

```
## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int
```

```
## rain = 0 kg/m2/day
```

```
## evaporation = 0 kg/m2/day
```

```
out8 <- abm(30, delta_t = 0.1, man_pars = man_pars1.0, grp_pars = grp_pars1.0,
            add_pars = list(storage_depth = 4, area = 1000, floor_area = 0,
                            slurry_mass = slurry_mass_dat, evap = 0, rain = 0),
            approx_method = c(temp = 'linear', pH = 'linear', slurry_mass = 'late'))
```

```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.
```

```
## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int
```

```
## rain = 0 kg/m2/day
```

```
## evaporation = 0 kg/m2/day
```

```
out9 <- abm(30, delta_t = 0.1, man_pars = man_pars1.0, grp_pars = grp_pars1.0,
            add_pars = list(storage_depth = 4, area = 1000, floor_area = 0,
                            slurry_mass = slurry_mass_dat, evap = 0, rain = 0),
            approx_method = c(temp = 'linear', pH = 'linear', slurry_mass = 'mid'))
```

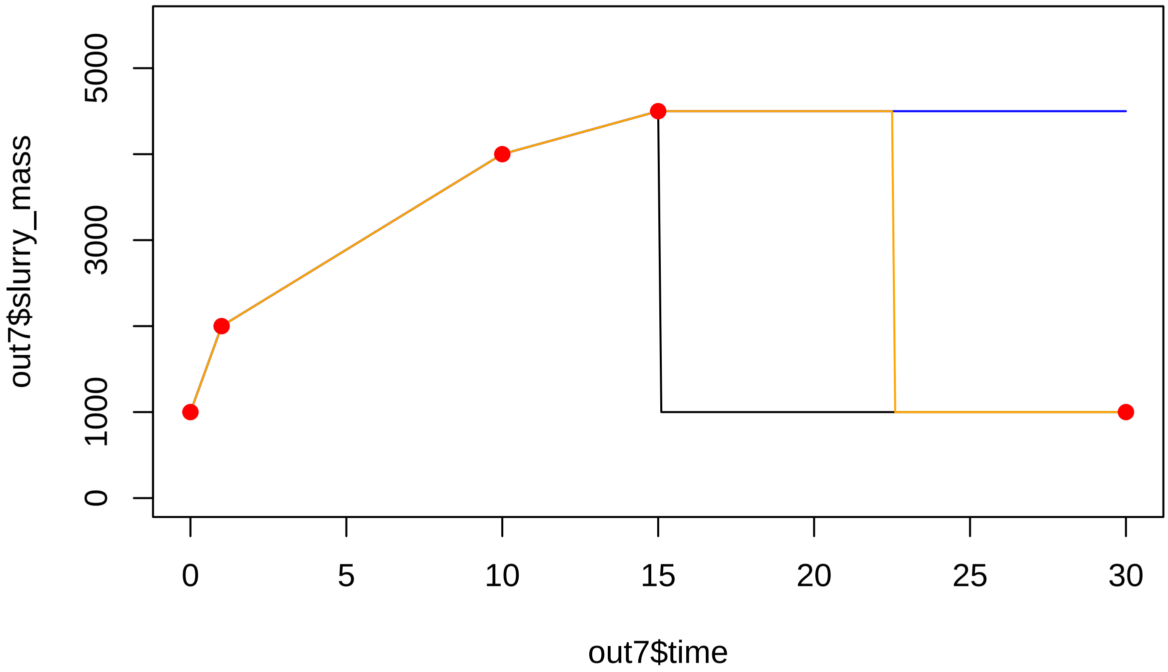
```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.
```

```
## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int
```

```
## rain = 0 kg/m2/day
```

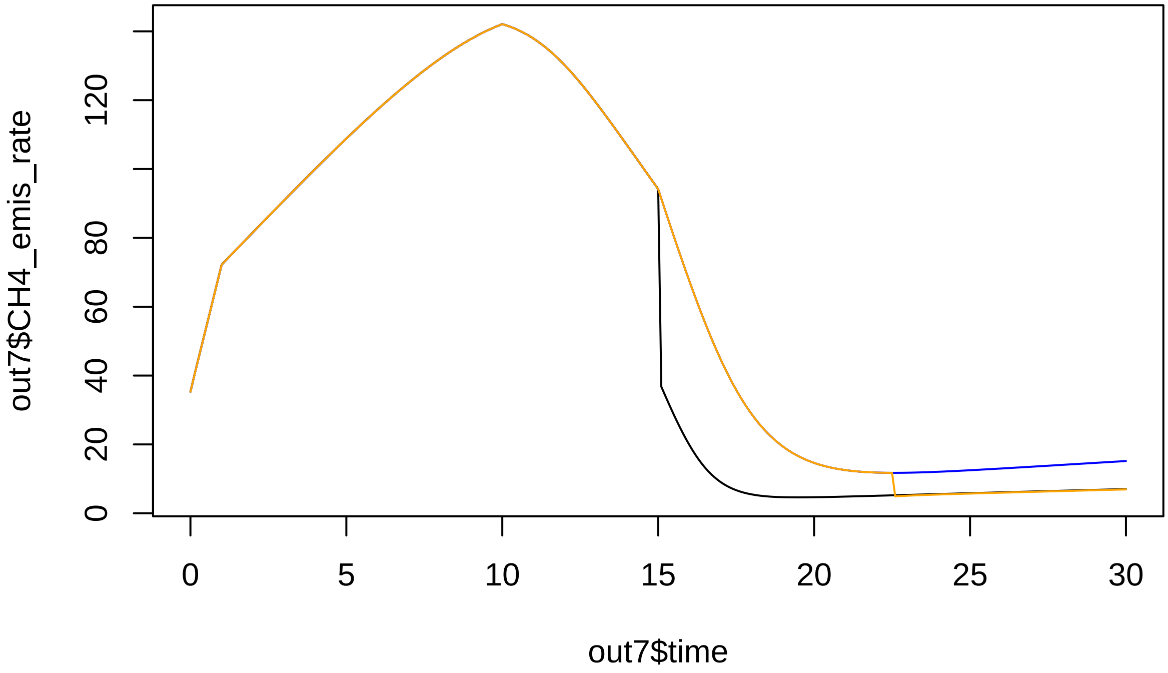
```
## evaporation = 0 kg/m2/day
```

```
plot(out7$time, out7$slurry_mass, type = 'l', ylim = c(0, 5500))
lines(out8$time, out8$slurry_mass, type = 'l', col = 'blue')
lines(out9$time, out9$slurry_mass, type = 'l', col = 'orange')
points(slurry_mass_dat$time, slurry_mass_dat$slurry_mass, col = 'red', pch = 19)
```



Note that last input point is ignored for 'late' method.

```
plot(out7$time, out7$CH4_emis_rate, type = 'l')
lines(out8$time, out8$CH4_emis_rate, type = 'l', col = 'blue')
lines(out9$time, out9$CH4_emis_rate, type = 'l', col = 'orange')
```



Case 4, rain/evaporation correction

```r
slurry_mass_dat <- read.csv('slurry_mass.csv')
```

Should get an error if adjusted slurry level is negative

```r
out_a0 <- abm(days = 4*365, add_pars = list(slurry_mass = slurry_mass_dat))
```

```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.

## Error in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Negative slurry mass va
##   Check parameters and try again.
```

```r
out_a1 <- abm(days = 4*365, add_pars = list(slurry_mass = slurry_mass_dat, area = 100))
```

```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.

## Warning in abm(days = 4 * 365, add_pars = list(slurry_mass = slurry_mass_dat, : Maximum slurry mass
## Check output.

## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int

## rain = 1.9 kg/m2/day

## evaporation = 0.25 kg/m2/day
```

Does it also work with different emptying alignment?

```r
out_a2 <- abm(days = 4*365, add_pars = list(slurry_mass = slurry_mass_dat, area = 100),
              approx_method = c(temp = 'linear', pH = 'linear', slurry_mass = 'mid'))
```

```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.

## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, :
## Setting rain and evap to 0 because "mid" method was selected.

## Warning in abm(days = 4 * 365, add_pars = list(slurry_mass = slurry_mass_dat, : Maximum slurry mass
## Check output.

## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int

## rain = 1.9 kg/m2/day

## evaporation = 0.25 kg/m2/day
```

```r
out_a3 <- abm(days = 4*365, add_pars = list(slurry_mass = slurry_mass_dat, area = 100),
              approx_method = c(temp = 'linear', pH = 'linear', slurry_mass = 'late'))
```

```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.
```

```
## Warning in abm(days = 4 * 365, add_pars = list(slurry_mass = slurry_mass_dat, : Maximum slurry mass e
## Check output.
```

```
## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int
```

```
## rain = 1.9 kg/m2/day
```

```
## evaporation = 0.25 kg/m2/day
```

```r
out_b <- abm(days = 4*365, add_pars = list(slurry_mass = slurry_mass_dat, rain = 0, evap = 0))
```

```
## Warning in abm_variable(days = days, delta_t = delta_t, times = times, y = y, : Fixed wash_water valu
## will be ignored because variable slurry input is used.
```

```
## Warning in abm(days = 4 * 365, add_pars = list(slurry_mass = slurry_mass_dat, : Maximum slurry mass e
## Check output.
```

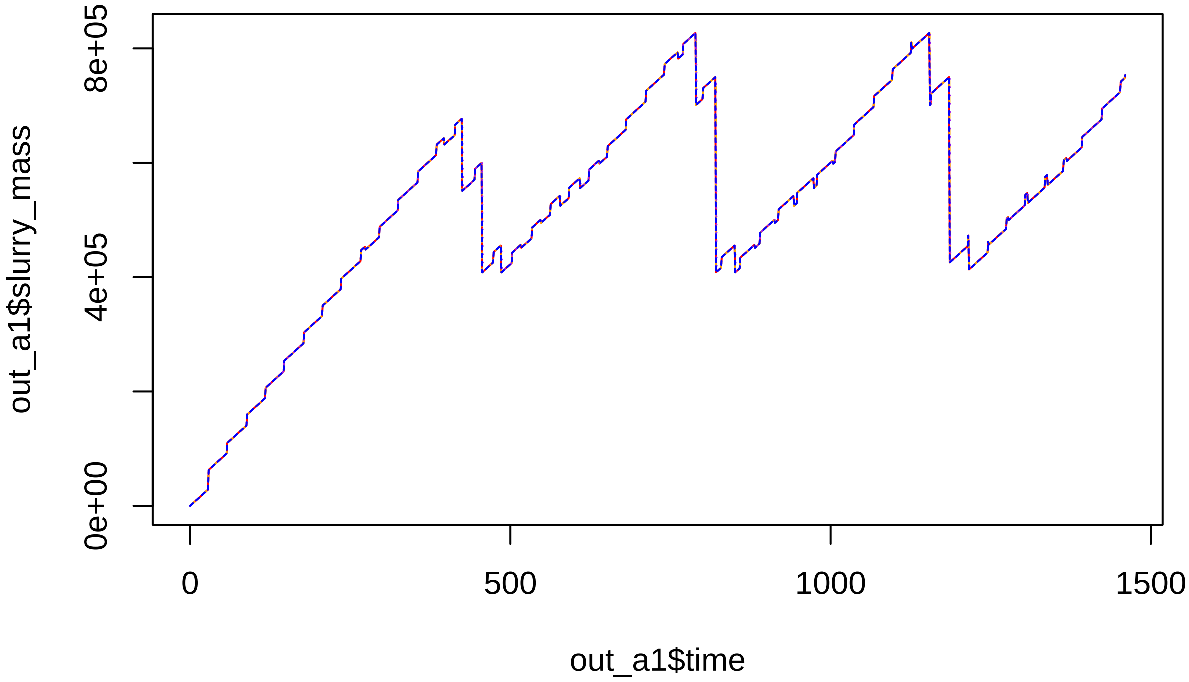
```
## arguments overwritten by slurry_mass: slurry_prod_rate, empty_int, resid_depth, wash_water, wash_int
```

```
## rain = 0 kg/m2/day
```

```
## evaporation = 0 kg/m2/day
```

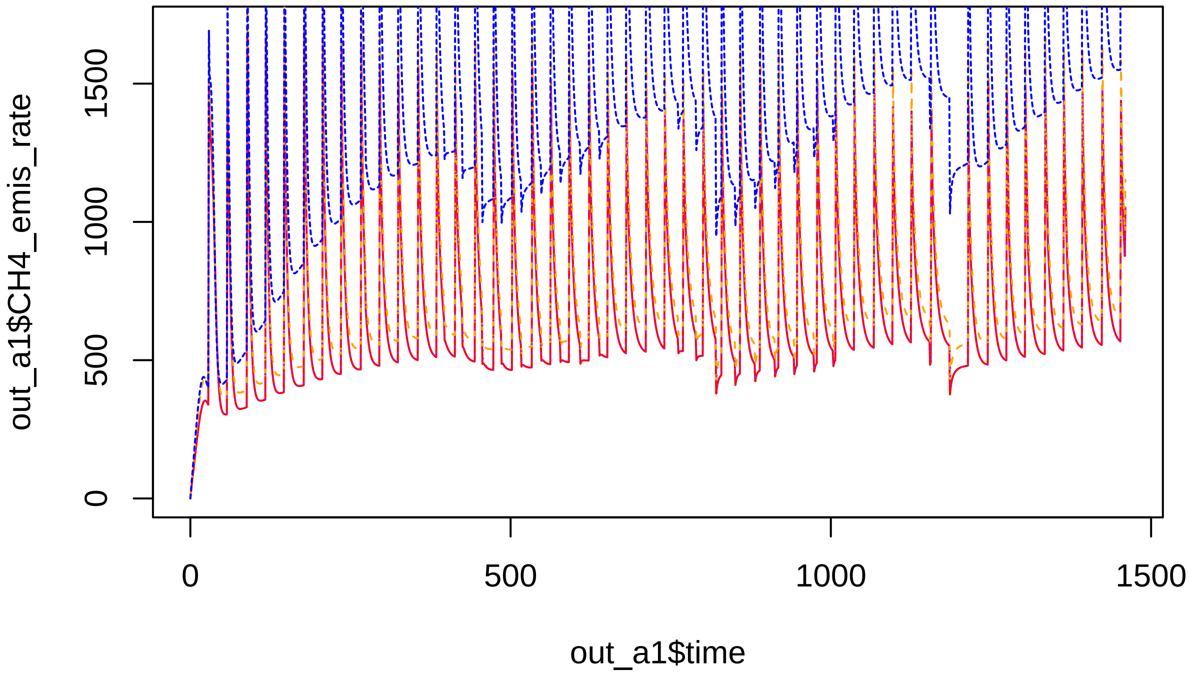
Check for mismatch between input slurry_mass and output slurry_mass

```r
plot(out_a1$time, out_a1$slurry_mass, col = 'red', type = 'l')
lines(out_a2$time, out_a2$slurry_mass, col = 'orange', lty = 2)
lines(out_a3$time, out_a3$slurry_mass, col = 'purple', lty = 3)
lines(out_b$time, out_b$slurry_mass, col = 'blue', lty = '3131')
```

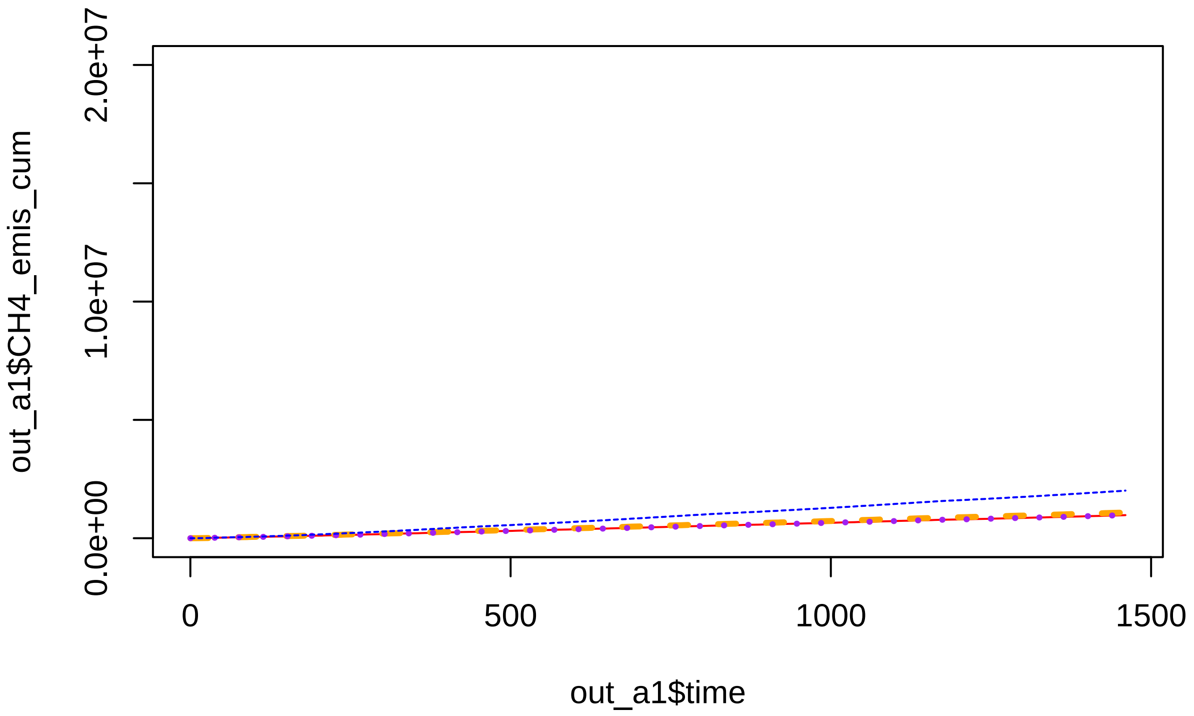


These different simulations should have different CH4 production. With no net rain, `abm()` interprets input as having more loading, so more methane.

```r
plot(out_a1$time, out_a1$CH4_emis_rate, col = 'red', type = 'l')
lines(out_a2$time, out_a2$CH4_emis_rate, col = 'orange', lty = 2)
lines(out_a3$time, out_a3$CH4_emis_rate, col = 'purple', lty = 3)
lines(out_b$time, out_b$CH4_emis_rate, col = 'blue', lty = '3131')
```



```
plot(out_a1$time, out_a1$CH4_emis_cum, col = 'red', type = 'l', ylim = c(0, 2E7))
lines(out_a2$time, out_a2$CH4_emis_cum, col = 'orange', lty = 2, lwd = 3)
lines(out_a3$time, out_a3$CH4_emis_cum, col = 'purple', lty = 3, lwd = 3)
lines(out_b$time, out_b$CH4_emis_cum, col = 'blue', lty = '3131')
```


out_a1$CH4_emis_cum
0.0e+00
1.0e+07
2.0e+07
out_a1$time