

# Speed of abmVar() in new simple1 version

Sasha D. Hafner

23 July, 2025 14:05

## Prep

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

```
## i Loading ABM
```

## Basic parameters

I am leaving out several elements that would only apply for `abmRegular()`.

```
mng_pars <- list(storage_depth = 4,  
                 area = 100,  
                 resid_enrich = 1)
```

```
sub_pars <- list(subs = c('cellulose', 'protein', 'lipids'),  
                 forms = c(cellulose = 'C6H10O5', protein = 'C4 H6.1 O1.2 N',  
                           lipids = 'C57 H104 O6', urea = 'CO(NH2)2'),  
                 T_opt_hyd = c(default = 60),  
                 T_min_hyd = c(default = 0),  
                 T_max_hyd = c(default = 90),  
                 hydrol_opt = c(lipids = 0.01, protein = 0.05, cellulose = 0.1),  
                 sub_fresh = c(lipids = 3, protein = 20, cellulose = 35),  
                 sub_init = c(lipids = 3, protein = 20, cellulose = 35))
```

```
grp_pars <- list(grps = c('m0', 'm1', 'm2'),  
                 yield = c(default = 0.05),  
                 xa_fresh = c(default = 0.05),  
                 xa_init = c(default = 0.05),  
                 dd_rate = c(default = 0.02),  
                 ksv = c(default = 1),  
                 qhat_opt = c(m0 = 1, m1 = 1, m2 = 2),  
                 T_opt = c(m0 = 18, m1 = 18, m2 = 28),  
                 T_min = c(m0 = 0, m1 = 6.41, m2 = 6.41),  
                 T_max = c(m0 = 25, m1 = 25, m2 = 38))
```

```
man_pars <- list(VFA_fresh = c(CH3COOH = 2), pH = 7, dens = 1000)  
chem_pars <- list(COD_conv = c(CH4 = 1/0.2507))
```

## 1. Short var pars

This will be for a slurry storage tank with a 30 m diameter and 4 m depth. So maximum volume is 2800 t ( $15^2 * \pi * 4$ ). It is 1/4 full at the start of the year, gradually filled, emptied in the autumn, and gradually filled.

```
var_dat <- data.frame(time = c(0, 200, 201, 365),
                      slurry_mass = c(1000, 2800, 10, 1000) * 1000,
                      temp_C = c(10, 12, 15, 12))
var_pars <- list(var = var_dat)
```

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

```
## i Loading ABM
```

```
system.time(
out1 <- abm(365,
            mng_pars = mng_pars,
            man_pars = man_pars,
            grp_pars = grp_pars,
            sub_pars = sub_pars,
            chem_pars = chem_pars,
            var_pars = var_pars,
            startup = 1)
)
```

```
##
```

```
## Startup run 1x -> and final run
```

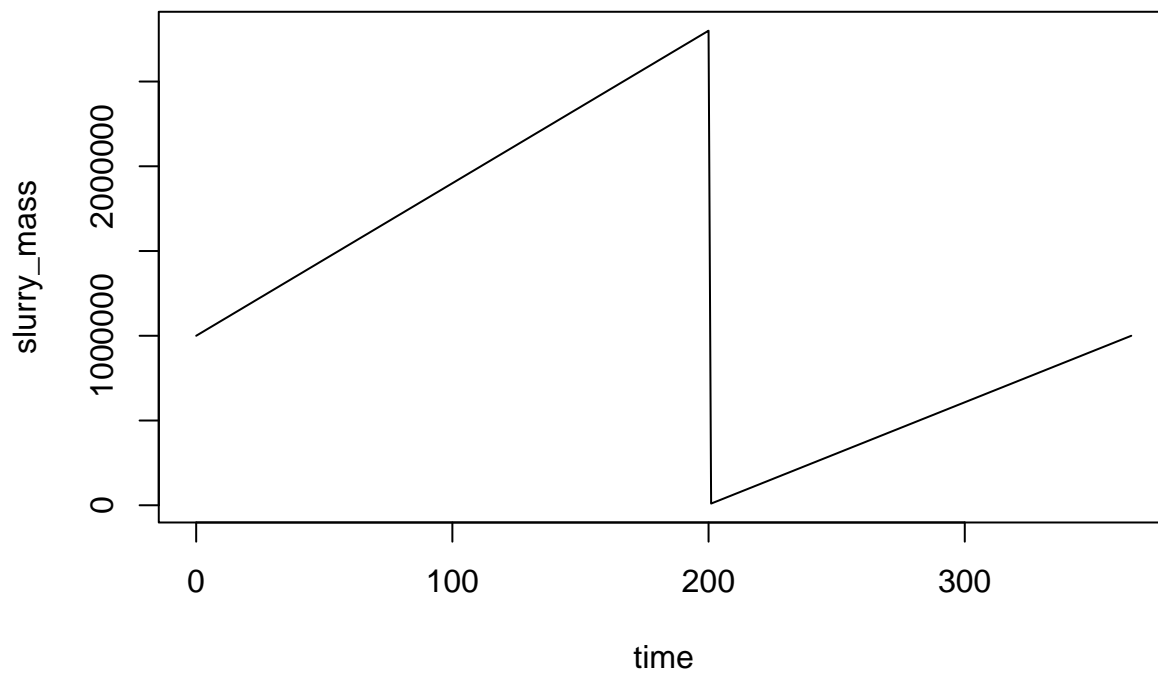
```
## Using starting conditions from `starting` argument
```

```
## user system elapsed
```

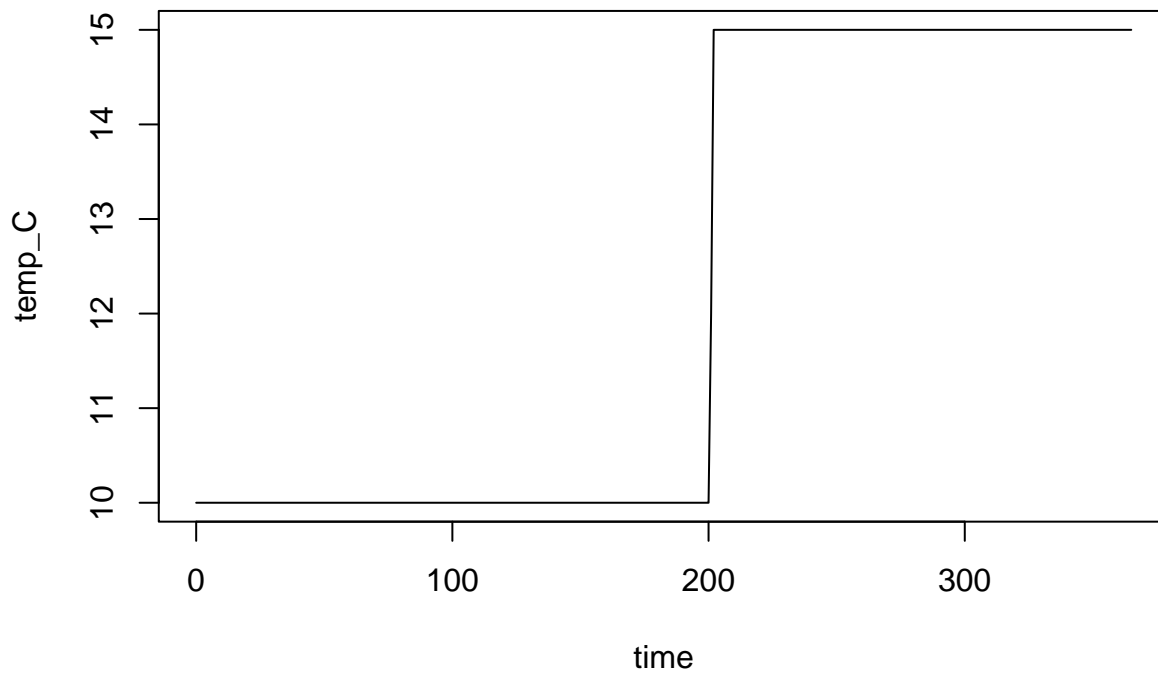
```
## 0.458 0.000 0.457
```

Here are some results.

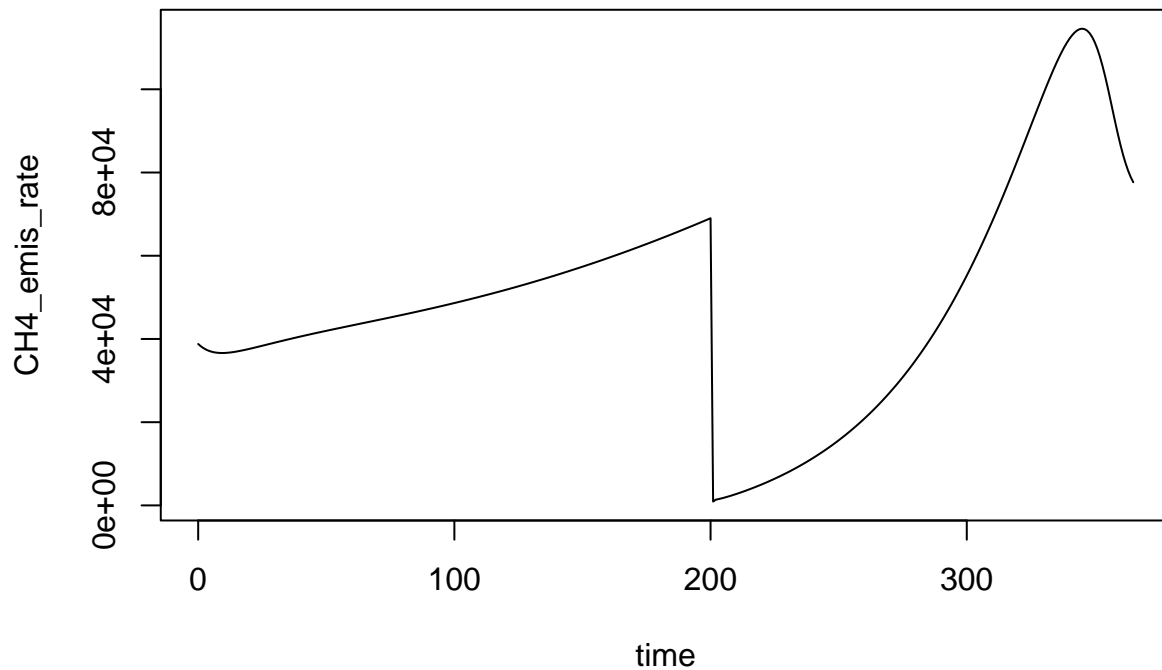
```
plot(slurry_mass ~ time, data = out1, type = 'l')
```



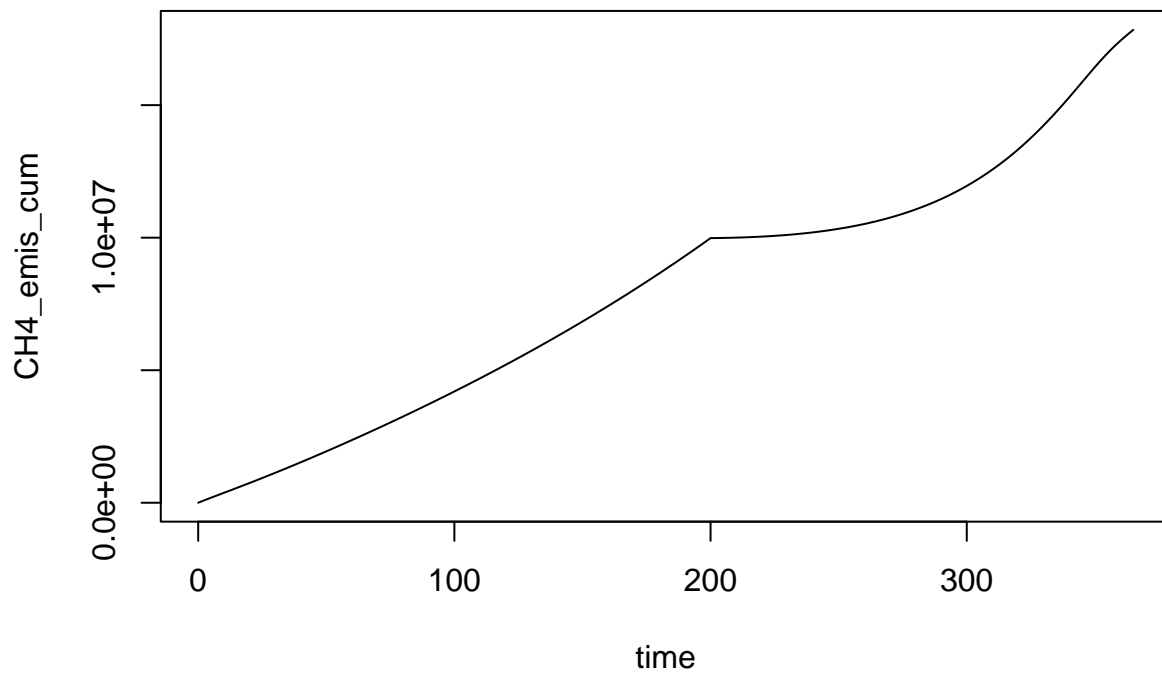
```
plot(temp_C ~ time, data = out1, type = 'l')
```



```
plot(CH4_emis_rate ~ time, data = out1, type = 'l')
```



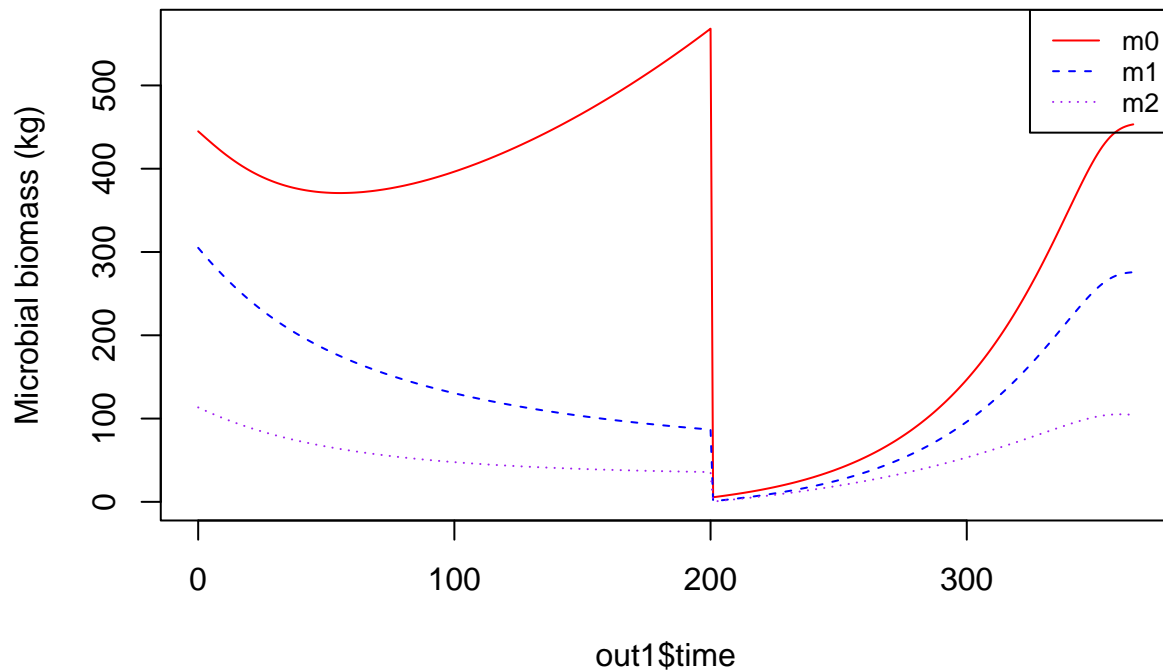
```
plot(CH4_emis_cum ~ time, data = out1, type = 'l')
```



Notice that two entered temperature values are ignored, because of 'early' setting and the other is at the last time.

And methanogens.

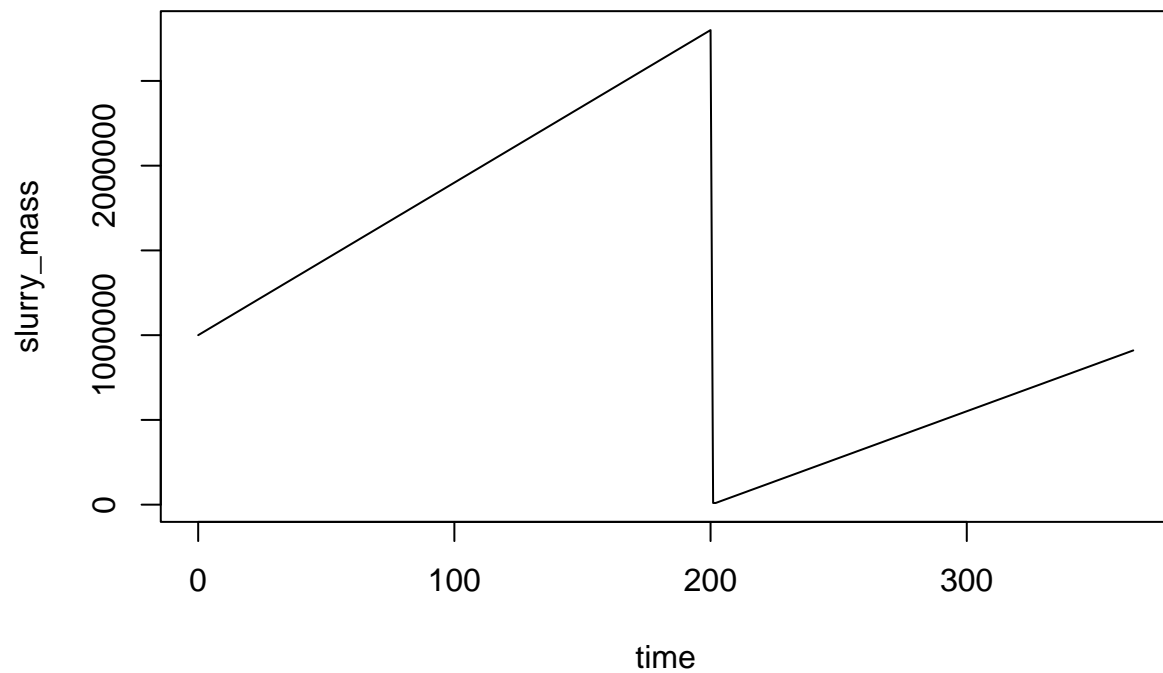
```
line_colors <- c('red', 'blue', 'purple','orange')
matplot(out1$time, out1[, nn <- c('m0','m1','m2')]/1000,
        type = 'l', lty = c(1:length(nn)), col = line_colors, ylab = 'Microbial biomass (kg)')
legend("topright", legend = nn, lty = c(1:length(nn)), col = line_colors, lwd = 1, cex = 0.8)
```



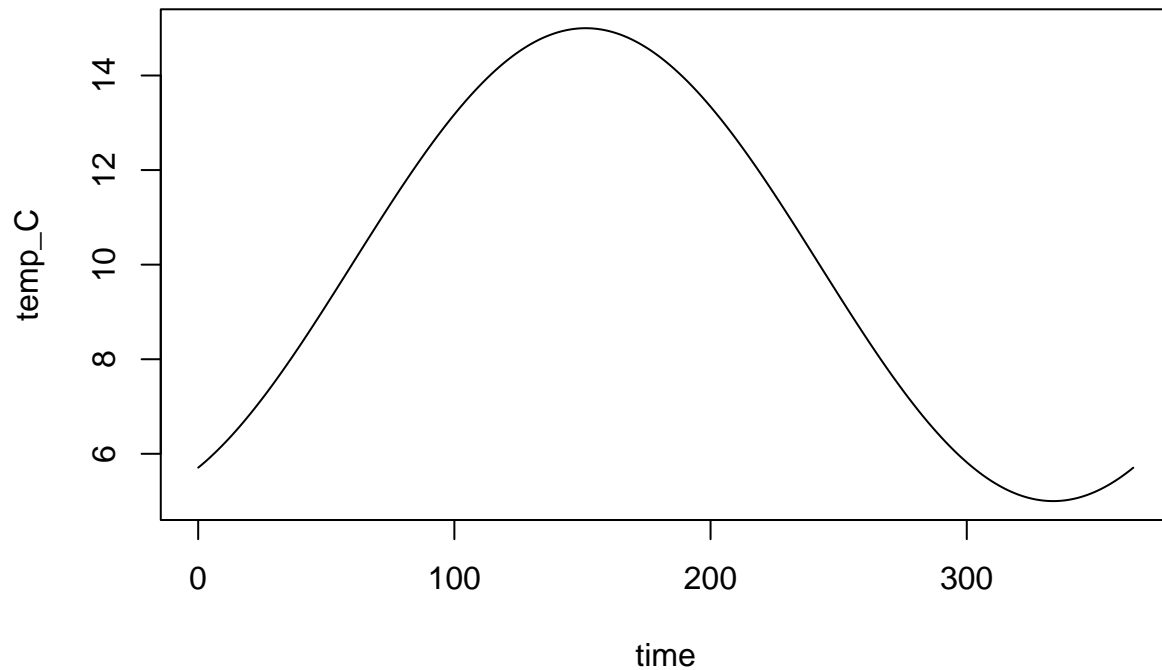
## 2. Daily

This scenario is more-or-less the same, but with daily resolution.

```
var_dat <- data.frame(time = c(0:200, 201, 202:365),
                      slurry_mass = c(1000 + 0:200 * 9, 10, 10 + 900 / 163 * 0:163) * 1000,
                      temp_C = 10 + 5 * sin((0:365 - 60) * 2 * pi / 365))
var_pars <- list(var = var_dat)
plot(slurry_mass ~ time, data = var_dat, type = 'l')
```



```
plot(temp_C ~ time, data = var_dat, type = 'l')
```



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

```
## i Loading ABM
```

```
system.time(
out1 <- abm(365,
  mng_pars = mng_pars,
  man_pars = man_pars,
  grp_pars = grp_pars,
  sub_pars = sub_pars,
  chem_pars = chem_pars,
  var_pars = var_pars,
  startup = 1)
)
```

```
##
```

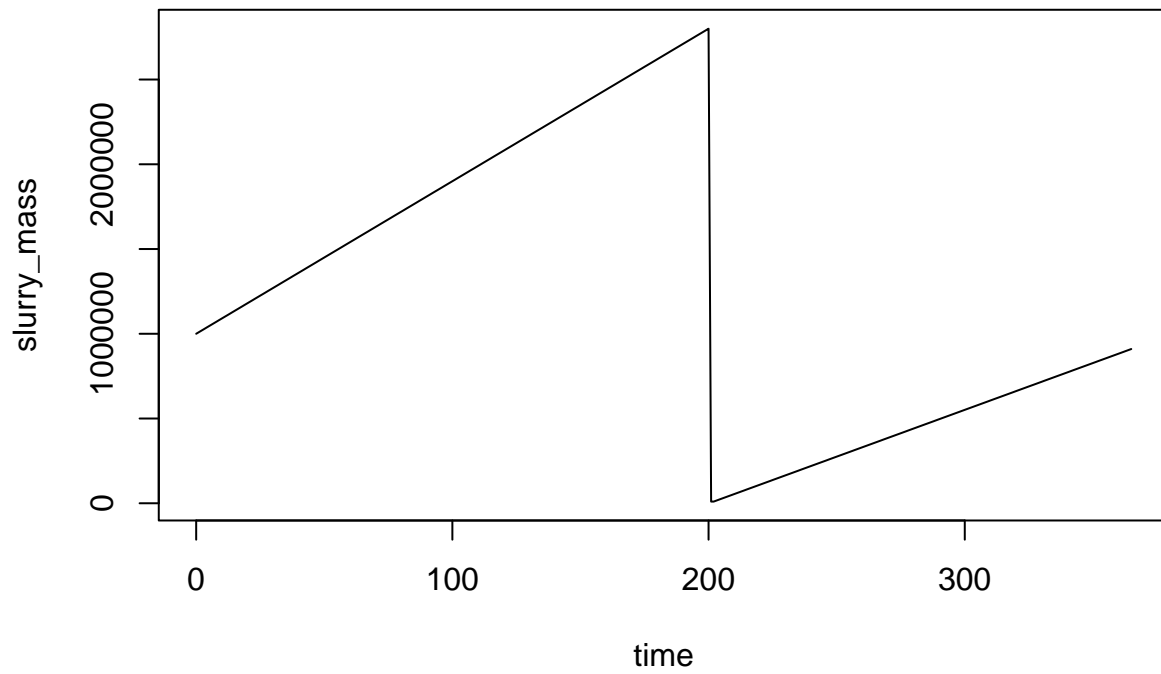
```
## Startup run 1x -> and final run
```

```
## Using starting conditions from `starting` argument
```

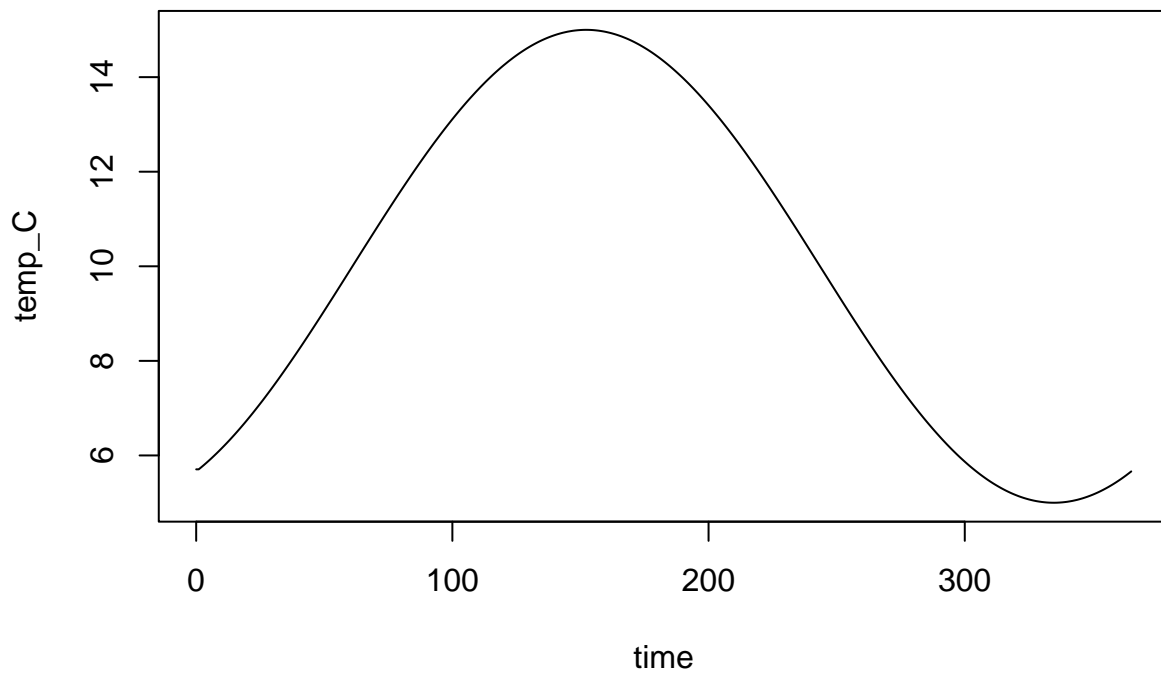
```
##   user  system elapsed
##  3.155   0.000   3.158
```

Here are some results.

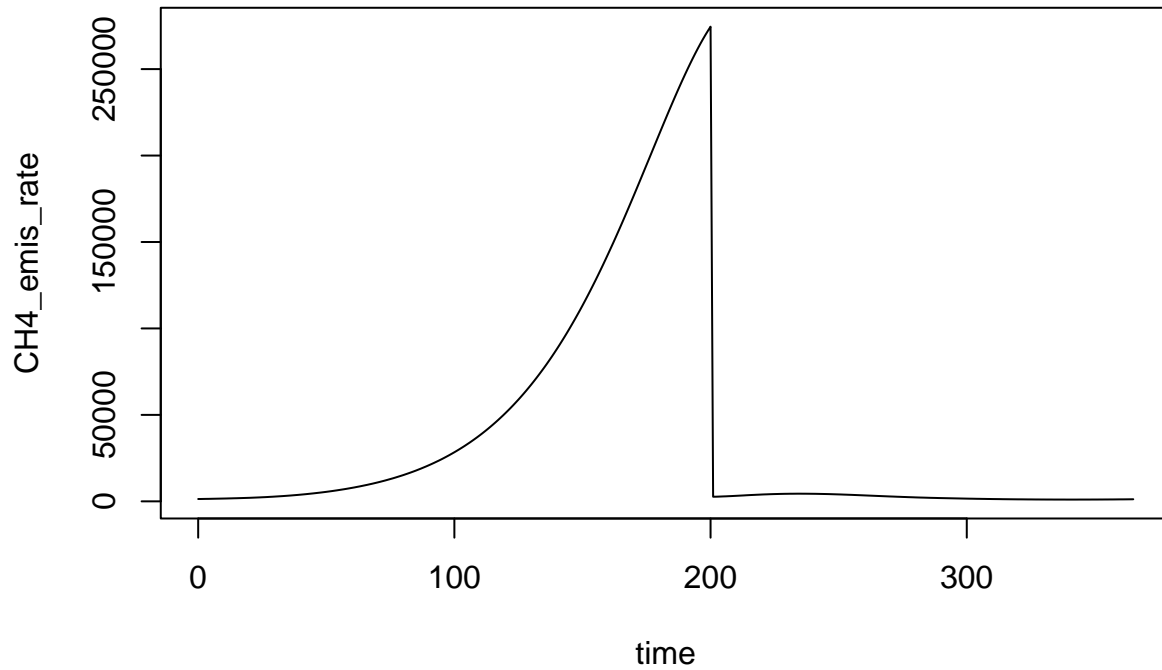
```
plot(slurry_mass ~ time, data = out1, type = 'l')
```



```
plot(temp_C ~ time, data = out1, type = 'l')
```



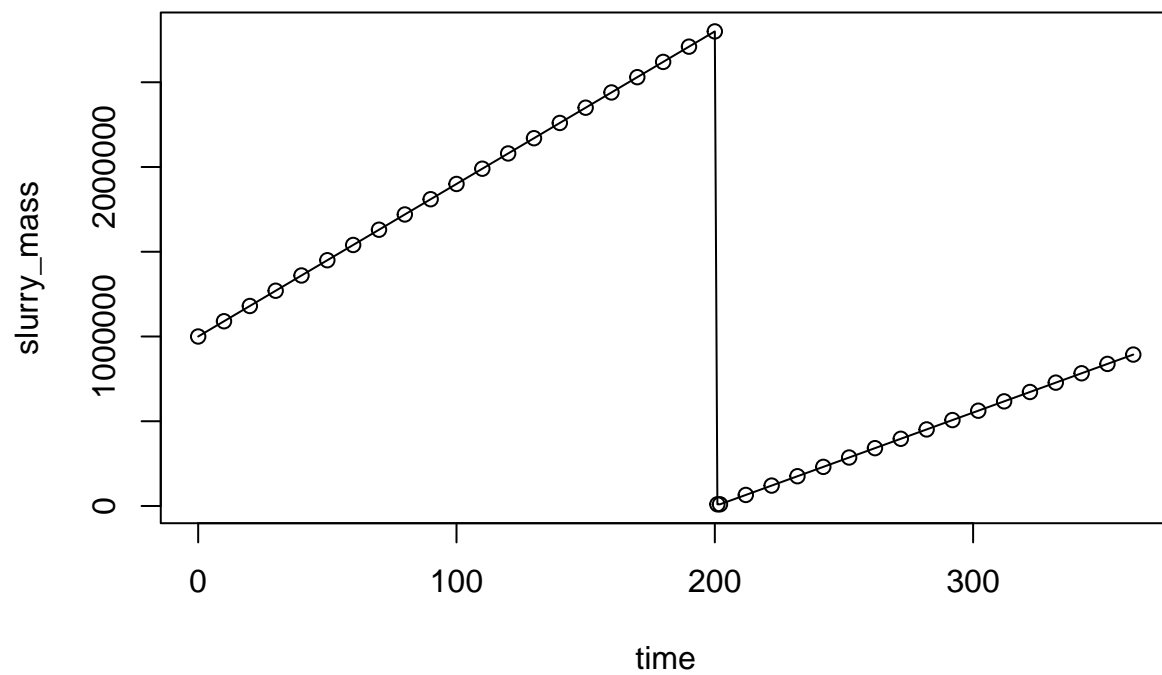
```
plot(CH4_emis_rate ~ time, data = out1, type = 'l')
```



### 3. Ten day blocks

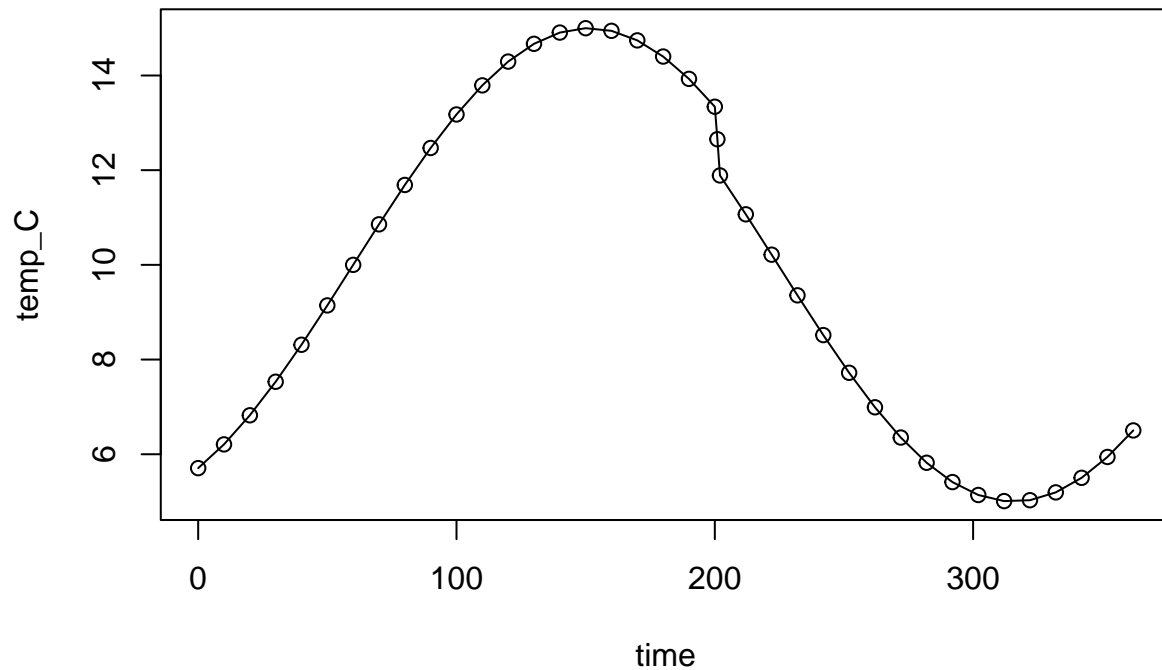
With an intermediate resolution

```
var_dat <- data.frame(time = c(0:20 * 10, 201, seq(202, 365, by = 10)),
                      slurry_mass = c(1000 + 0:20 * 90, 10, 10 + 900 / 163 * 0:16 * 10) * 1000,
                      temp_C = 10 + 5 * sin((0:38 * 10 - 60) * 2 * pi / 365))
var_pars <- list(var = var_dat)
plot(slurry_mass ~ time, data = var_dat, type = 'o')
```





```
plot(temp_C ~ time, data = var_dat, type = 'o')
```

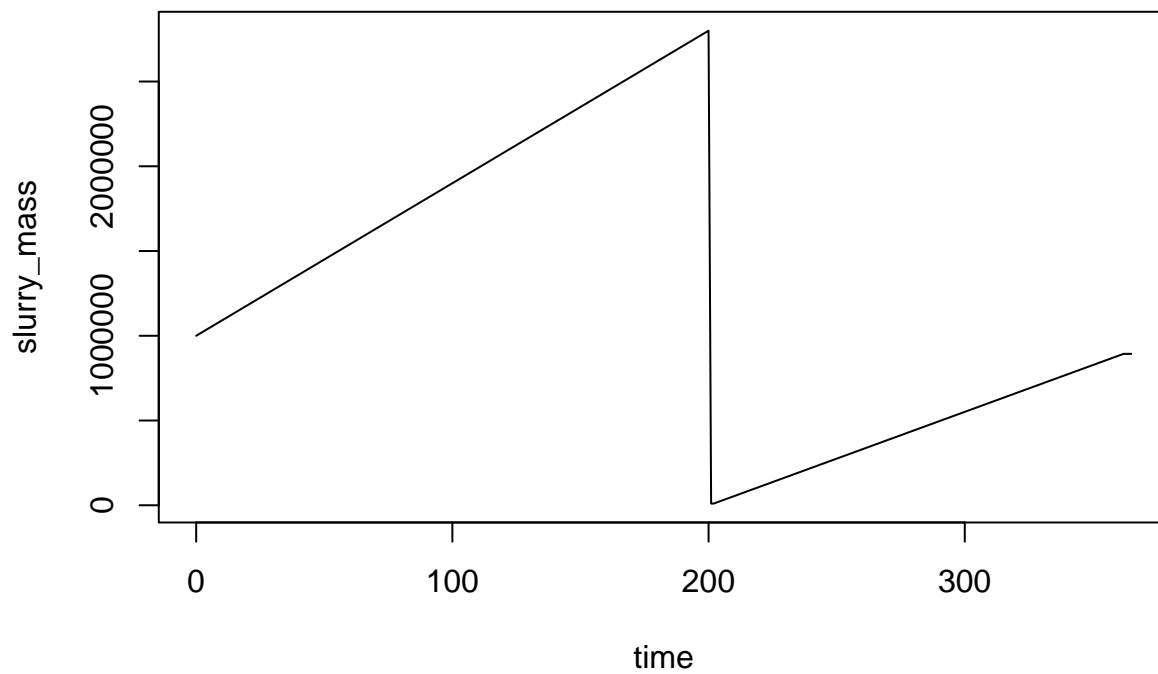


```
system.time(
out1 <- abm(365,
  mng_pars = mng_pars,
  man_pars = man_pars,
  grp_pars = grp_pars,
  sub_pars = sub_pars,
  chem_pars = chem_pars,
  var_pars = var_pars,
  startup = 1)
)
```

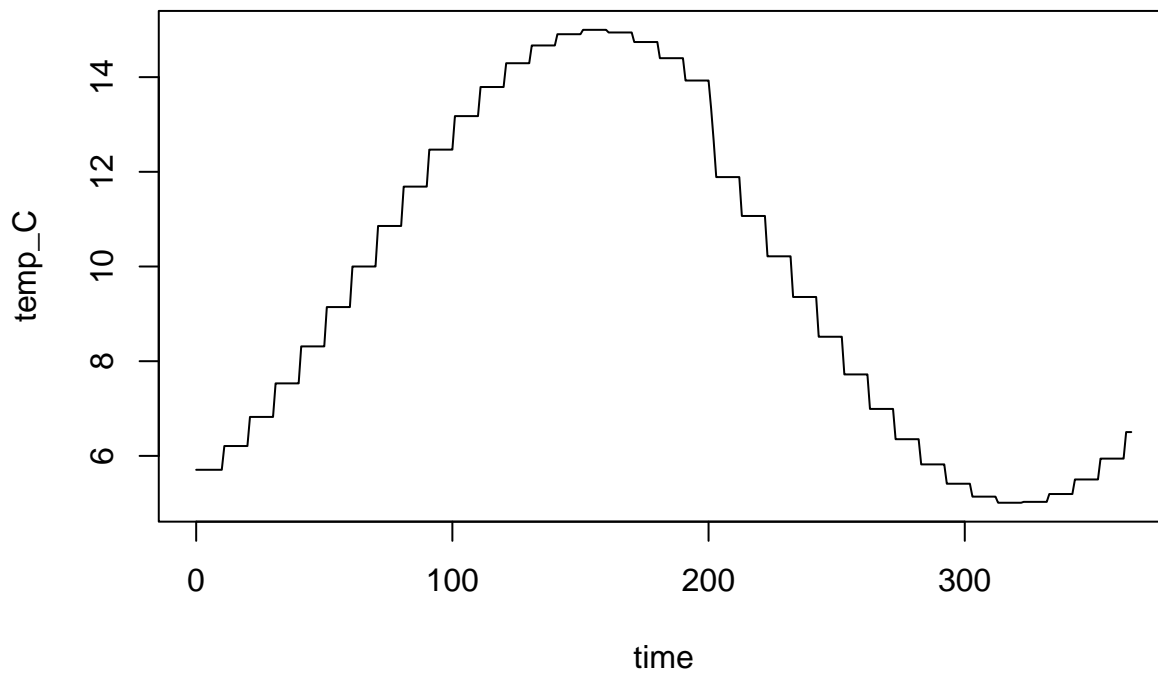
```
##
## Startup run 1x -> and final run
## Using starting conditions from `starting` argument
```

```
##   user  system elapsed
## 0.735  0.000  0.735
```

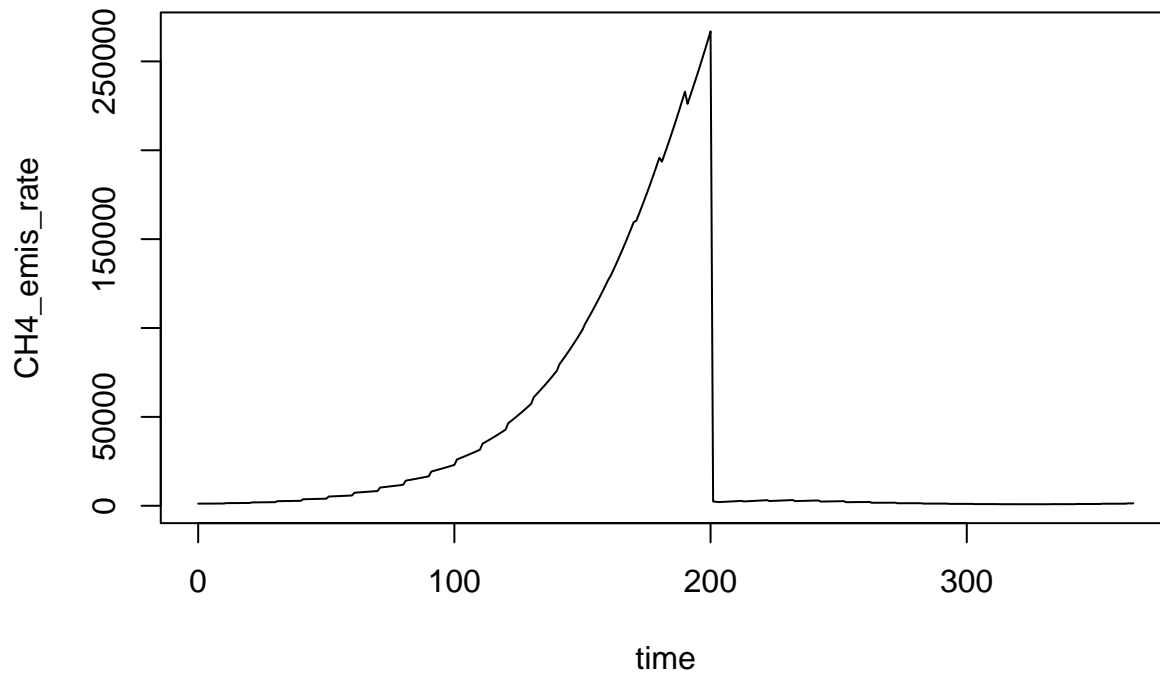
```
plot(slurry_mass ~ time, data = out1, type = 'l')
```



```
plot(temp_C ~ time, data = out1, type = 'l')
```



```
plot(CH4_emis_rate ~ time, data = out1, type = 'l')
```



And with more startup time

```
system.time(
out1 <- abm(365,
  mng_pars = mng_pars,
  man_pars = man_pars,
  grp_pars = grp_pars,
  sub_pars = sub_pars,
  chem_pars = chem_pars,
  var_pars = var_pars,
  startup = 3)
)
```

```
##
## Startup run 1x -> 2x ->
## Using starting conditions from `starting` argument
## 3x ->
## Using starting conditions from `starting` argument
## and final run
## Using starting conditions from `starting` argument
##   user  system elapsed
##  1.451   0.001   1.454
```