Reproduce MEND from Wang et al 2013

Kalyn Dorheim

June 05, 2020

Purpose

Run the basic out of the box MEND using the parameter values and initial conditions from Wang's 2013 paper. The objective of these runs is to make sure that the model has correctly been translated into R code in a way that is some what flexible and easy to manipulate.

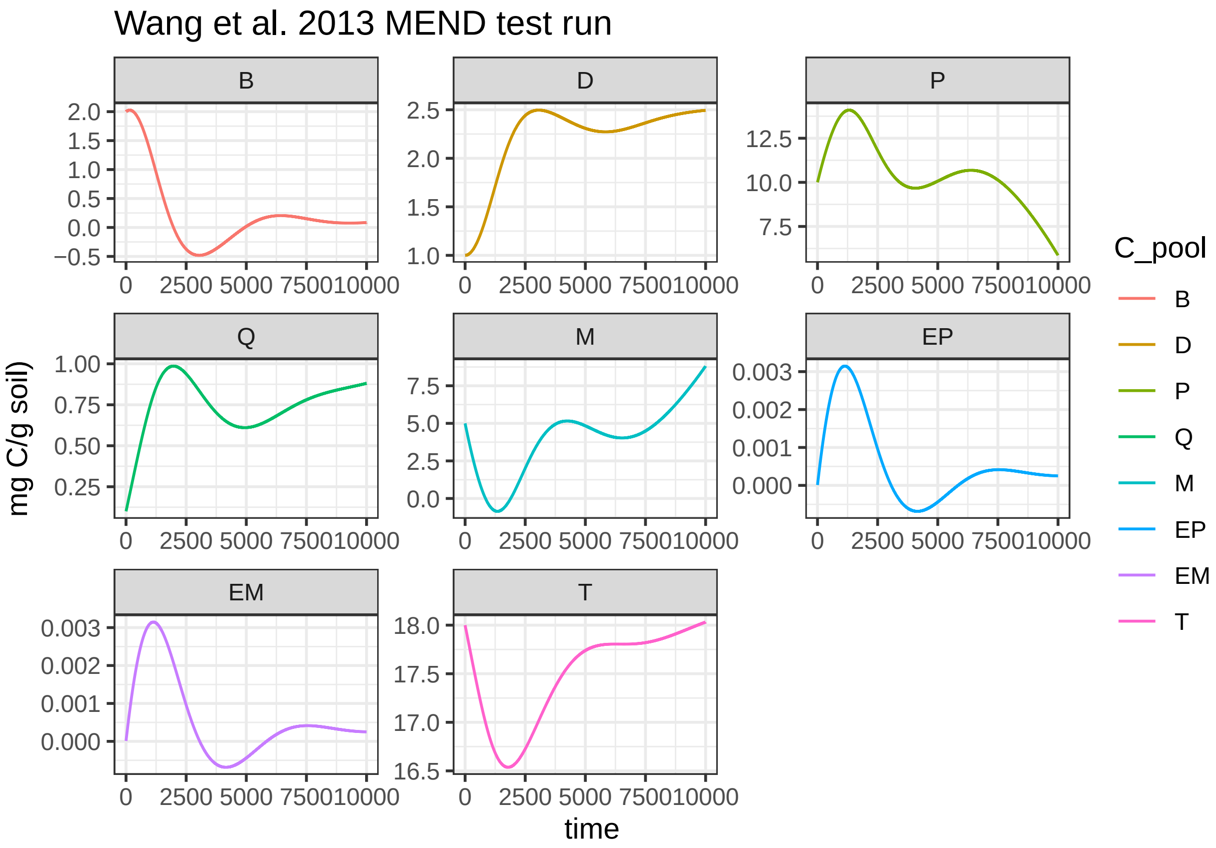
```
# Define the inital conditions of the carbon pool based on table 3 from Wang et al. 2013.
state <- c("B" = 2, "D" = 1, "P" = 10, "Q" = 0.1, "M" = 5, "EP" = 1e-5, "EM" = 1e-5, 'T' = 18)

times <- seq(0, 10000, by = 1)

out <- ode(y = state, times = times, parms = default_parameters,
           func = MEND_carbon_pools, flux_function = MEND_fluxes)

# Now format and plot results use the default value for Kdes the desorption rate.
results <- melt(as.data.table(out),  measure.vars = names(state),
                variable.name = "C_pool", value.name = 'value')

ggplot(data = results) +
  geom_line(aes(time, value, color = C_pool)) +
  theme_bw() +
  facet_wrap("C_pool", scales = 'free') +
  labs(y = 'mg C/g soil)', title = 'Wang et al. 2013 MEND test run') +
  NULL
```



Run MEND with a series of different parameter combinations. The idea here is to reproduce figure 5 from the Wang et al. 2013 paper where MEND was solved for multiple times with different values for k_{des} (desorption rate).

```
results <- cbind(results, default_parameters[parameter == 'K.des',
                                              list('param_value' = value, parameter)])
times <- seq(0, 10000, by = 1)
param_values <- seq(from = 1e-4, to = 1e-2, length.out = 4)
p            <- default_parameters
lapply(param_values, function(param){

  # Subset the parameter values.
  p[parameter == 'K.des', ]$value <- param
  out <- ode(y = state, times = times, parms = p,
             func = MEND_carbon_pools, flux_function = MEND_fluxes, method = "radau", atol = 1e-10, rto

  # Now format and plot results use the default value for Kdes the desorption rate.
  melt(as.data.table(out),  measure.vars = names(state), variable.name = "C_pool", value.name = 'value'
    cbind(p[parameter == 'K.des', list("K.des" = value)])

}) %>%
  dplyr::bind_rows() ->
  results
```

```
##  EXIT OF RADAU5 AT X= 7055.23
##  STEP SIZE TOO SMALL, H= 6.03605e-12
```

```
## Warning in radau(y, times, func, parms, ...): step size becomes too small
## Warning in radau(y, times, func, parms, ...): Returning early. Results are
## accurate, as far as they go
```

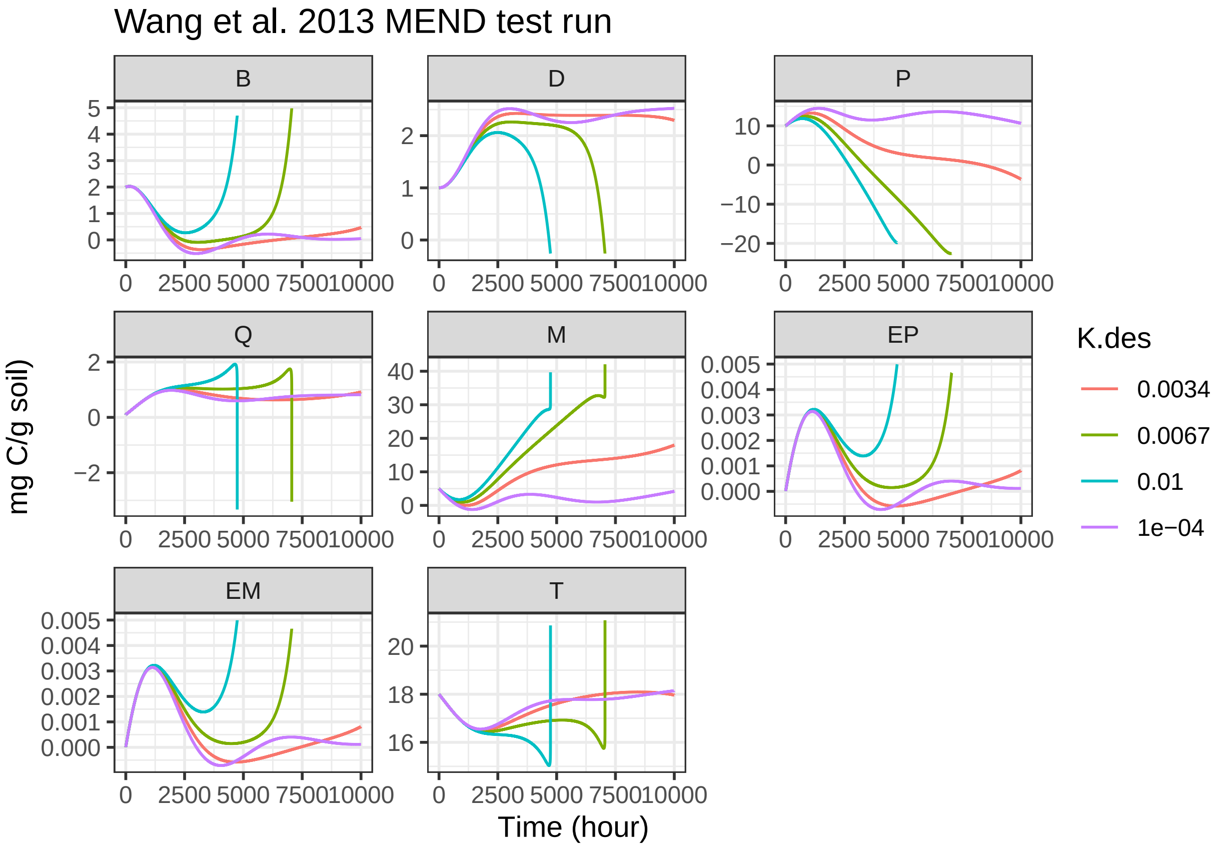
```
##  EXIT OF RADAU5 AT X= 4737.69
##  STEP SIZE TOO SMALL, H= 4.14415e-12
```

```
## Warning in radau(y, times, func, parms, ...): step size becomes too small
```

```
## Warning in radau(y, times, func, parms, ...): Returning early. Results are
## accurate, as far as they go
```

```
results[ , K.des := as.character(K.des)]

ggplot(data = results) +
  geom_line(aes(time, value, color = K.des)) +
  theme_bw() +
  facet_wrap("C_pool", scales = 'free') +
  labs(y = 'mg C/g soil)', title = 'Wang et al. 2013 MEND test run', x = 'Time (hour)') +
  NULL
```



But comparing these results with the ones from figure 5, in figure 5 Q ranged from 0 to 1.8 and DOC (D) ranged from 0.1 to 1. But in my runs D and Q fall outside these ranges. This has me wondering if I've made a mistake in the code or in when I defined the parameter values. I will have to double check on that but in the mean time do you have any ideas about tests to run to validate the model structure?