Sperry's supply-demand-loss model

Liam Langan

May 28, 2018

1 Introduction

Sperry and Love (2015 (What plant hydraulics can tell us about responses to climate-change droughts)) developed a model where a supply function (transpiration rate) (E) (kg hr^{-1} m^{-2}) is derived which calculates the potential rate/amount of water able to be supplied from the soil to the atmosphere. Transpiration (E) is influenced by the canopy sap pressure (p$_c anopy)(MPa) via changes to the hydraulic conductance of the plant(k)(kghr^{-1}$ MPa^{-1} m^{-2}). Hydraulic conductance of the plant (k) is the conductance when there is no difference in matric potential between the soil and the leaf.

Below are parameters for the vulnerability-conductance curve,

```
p50 <- 2.5 # the matric potential where conductance is reduced by 50%
k_max <- 8 # maximum plant conductance - this is a trait in aDGVM2
res <- 1/k_max # resistance is simply the inverse of conductance
p_canopy <- seq(0.0, 8, length=1000)
# this assumes initial plant matric potential is the same as the soil matric potential
predawn_soil_mat_pot <- seq(0,3, length=5)
E_p_canopy <- matrix(0,0,nrow=1000, ncol=5) # this is the supply function

##------------------------------------------------------------------------------------------
## make transpiration demand ---------------------------------------------------------------
# Maximum stomatal conductance
# (Sperry 2016, 2130 kg h^-1 m^-2) NOTE should be (kg h^-1 MPa^-1 m^-2) (12563.1 in Excel doc)
Gmax <- 12563.1
G <- rep(Gmax, length=5)
D <- 0.5*0.001   # (Sperry 2016, leaf-to-air vapor pressure deficit 1 kPa)(0.001 converts to MPa)
# NOTE VPD converstion from kPa to MPa isn't documented in Sperry, I'm doing it as it makes
# sense and produces realistic amounts of transpirtational demand.

# Pcrit, i.e. a matric potential we choose where we decide conductance is effectively zero.
# Used this to get Ecrit, i.e. maximum transpiration beyond which leads to runaway cavitation
P_crit <- 6 # MPa - this is arbitrary and could be a plant trait.
# In Sperry (2016) a P_crit cutoff is chosen (either very low conductance or
# shallow slope of a tangent to the transpiration curve)

# get maximum transpiration possible based on Pcrit and soil matric potential
E_crit <- rep(0, lenght=5) # maximum transpiration beyond which leads to runaway cavitation
evap_demand <- rep(0, lenght=5) # evaporative demand
```

with the the conductance vulnerability curve we use in aDGVM2, which is analagous to Sperry's curve, defined as:

```
k_p_canopy <- function(p_canopy) { ((1 - (1 / (1 + exp(3.0*(p50 - p_canopy)))))) / res }
```

The transpiration rate is the integral of the vulnerability-conductance curve between the soil (pre-dawn) matric potential and (p$_c anopy) and is calculated as follows:$

```
for(j in 1:length(predawn_soil_mat_pot))
  {
```

```
    for(i in 1:1000)
      {
        ffx <- integrate(k_p_canopy, predawn_soil_mat_pot[j], p_canopy[i] )
        E_p_canopy[i,j] <- pmax(0, ffx$value)
      }
}
```

Here we get the slope of the line which is tangent to the tanspiration curve at any particulat water potential. This slope is the conductance at this water potential.

```
cond_max_slope_sperry <- rep(0, lenght=5) # get the maximum slope of conductance given pre-dawn water pote
for(i in 1:length(predawn_soil_mat_pot))
  {
    cond_max_slope_sperry[i] <-  k_p_canopy(predawn_soil_mat_pot[i])
    # for Sperry the maximum conductance is always the pre-dawn matric potential/soil matric potential
  }
```

We calculate the maximum transpiration beyond which leads to runaway cavitation E_{crit} based on a matric potential we choose where we decide conductance is effectively zero P_{crit}.

```
for(j in 1:5)
  {
    ffx <- integrate(k_p_canopy, predawn_soil_mat_pot[j], P_crit )
    E_crit[j] <- pmax(0, ffx$value)
    evap_demand[j] <- G[j]*D
    if(evap_demand[j] > E_crit[j]) evap_demand[j] <- E_crit[j] # demand can't be greater than maximum supp
  }
```

We then calculate the matric potential where evaporative demand is met

```
## quick and dirty method to find the psi where demand is met.
demand_met_at_sperry <- rep(0, length=5)
psi_demand_met_at_sperry <- rep(0, length=5)
demand_met_at_slope_sperry <- rep(0, length=5)
loss_function_sperry <- rep(0, length=5)
regulated_transpiration <- rep(0, length=5)
regulated_leaf_psi <- rep(0, length=5)

##------------------------------------------------------------------------------------------------------
##------------------------------------------------------------------------------------------------------
## max regulation is the point where delta P hits its maximum, it should be held at max once passed.
G <- matrix(Gmax, Gmax, ,nrow=1000, ncol=5)

psi_1 <- matrix(0,0,nrow=1000, ncol=5)#
slope_demand <- matrix(0,0,nrow=1000, ncol=5)
loss_fun_sp_gs <- matrix(0,0,nrow=1000, ncol=5)
reg_leaf_psi <- matrix(0,0,nrow=1000, ncol=5)

regulated_trans <- matrix(0,0,nrow=1000, ncol=5)
regulated_Gs <- matrix(0,0,nrow=1000, ncol=5)

regulated_trans_trap <- matrix(0,0,nrow=1000, ncol=5)
regulated_Gs_trap <- matrix(0,0,nrow=1000, ncol=5)

non_regulated_trans <- matrix(0,0,nrow=1000, ncol=5)
non_regulated_Gs <- matrix(0,0,nrow=1000, ncol=5)

max_slo_sp <- rep(0, length=5)
```

```r
for(j in 1:5)
{

  max_slo_sp[j] <- k_p_canopy(predawn_soil_mat_pot[j])

  for(i in 1:1000)
  {
   psi_1[,j] <- seq(0, max(p_canopy), length=1000) #psi_leaf[i]
   slope_demand[i,j] <- k_p_canopy(psi_1[i,j])

   ffx <- integrate(k_p_canopy, predawn_soil_mat_pot[j], psi_1[i,j] )
   non_regulated_trans[i,j] <- pmax(0, ffx$value)
   non_regulated_Gs[i,j] <- non_regulated_trans[i,j]/D # E = G*VPD ---- G = E/VPD (VPD=1, 0.001 transforms

   loss_fun_sp_gs[i,j] <- slope_demand[i,j] / max_slo_sp[j]
   #reg_leaf_psi[i,j] <- predawn_soil_mat_pot[j] + ((psi_1[i,j] - predawn_soil_mat_pot[j])*loss_fun_sp_gs[
   if(i==1) reg_leaf_psi[i,j] <- pmax(0, ((psi_1[i,j] - predawn_soil_mat_pot[j])*loss_fun_sp_gs[i,j]))

   if(i>1)
   {
     reg_leaf_psi[i,j] <- pmax(0, pmax(reg_leaf_psi[i-1,j], ((psi_1[i,j] - predawn_soil_mat_pot[j])*loss_f
   }

  #
   ffx <- integrate(k_p_canopy, predawn_soil_mat_pot[j], predawn_soil_mat_pot[j] + reg_leaf_psi[i,j] )
   regulated_trans[i,j] <- pmax(0, ffx$value)
   regulated_Gs[i,j] <- regulated_trans[i,j]/D # E = G*VPD ---- G = E/VPD (VPD=1, 0.001 transforms to MPa)
   G[i,j] <- G[i,j]*loss_fun_sp_gs[i,j]

  }
}

demand_place_holder <- rep(0, length=5)
supply_limit_place_holder <- rep(0, length=5)
min_diff_place_holder <- rep(0, length=5)
supply_place_holder <- rep(0, length=5)


for(i in 1:5)
{
    a <-which(non_regulated_trans[,i] >= evap_demand[i])
    demand_place_holder[i] <-a[2]

    min_diff <- regulated_trans[,i] - non_regulated_trans[demand_place_holder[i]]
    min_diff_max <- which(min_diff == max(min_diff))
    supply_limit_place_holder[i] <- min_diff_max[1]

    abs_min_diff <- abs(min_diff)
    abs_min_diff <- which(abs_min_diff == min(abs_min_diff))
    supply_place_holder[i] <- abs_min_diff[1]
}
```

```
                    ## Error in p_canopy[, 1]:  incorrect number of dimensions
    ## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
    ## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
    ## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
    ## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
    ## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
    ## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
```

```
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type, ...):  plot.new has not been called yet
## Error in strwidth(legend, units = "user", cex = cex, font = text.font):  plot.new has not been
called yet
## Error in text.default(7.5, 24.5, "(A)", cex = 2):  plot.new has not been called yet
## Error in plot(psi_canopy, (1 - sperry_cond(psi_canopy)/sperry_cond(0)) * :  object 'psi_canopy'
not found
```

From Sperry (2016) "Mathematically, P rises to a maximum before decreasing back to zero as E 0 increases to Ecrit. This decline in P is unrealis- tic (Saliendra et al., 1995), so it is assumed that P saturates at its maximum as E 0 increases. Eqn 5 expresses the outcome that xylem pressure is regulated in proportion to the damage caused by taking no action. (4) The regulated E corresponding to P is determined from the supply function. (5) The G is solved from E/D to determine how much it is reduced below Gmax. The model does not partition G into stomatal vs boundary layer com- ponents, but G is controlled by stomatal regulation. Cuticular water loss is assumed to be zero." I was unsure what Sperry ment with ΔP saturates however examining Fig.1B one can see the regulated response is too extreme. Plotting the regulated leaf matric potential against the demand defined matiric reveals that the regulated leaf matric potential reaches a maximum and then decreases. Fixing the regulated leaf matric potential to its maximum value once it has passed this value solves this issue. The next issue to solve is how sperry is calculating his percentage loss of stomatal conductance.