Sperry's supply-demand-loss model

Liam Langan

May 25, 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 ($E_{p-canopy}$) is derived which calculates the potential rate/amount of water able to be supplied from the soil to the atmosphere, i.e. potential transpiration. Transpiration is influenced by xylem pressure (ψ_{xylem}), hydraulic conductivity of the plant (K_{plant}), the hydraulic conductivity of the soil K_{soil} and the rate at which hydraulic conductivity is reduced as xylem pressure increases or soil conductivity decreases.

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
psi_canopy <- seq(0.0, 5, length=1000)
predawn_soil_mat_pot <- seq(0,3, length=5) # this assumes initial plant matric potential is the same as th
cum_can_transport <- matrix(0,0,nrow=1000, ncol=5) # this is the supply function

## make transpiration demand ----------------------------------------------------------------------------
# Maximum stomatal conductance
Gmax <- 12563.1 # (Sperry 2016, 2130 kg h^-1 m^-2) NOTE should be (kg h^-1 MPa^-1 m^-2) (12563.1 in Excel
G <- rep(Gmax, length=5)
#VPD <- 0.5*0.001   # (Sperry 2016, leaf-to-air vapor pressure deficit 1 kPa) (0.001 converts to MPa)
VPD <- 6.5*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 proc
# amounts of transpirtational demand.
#evap_demand <- G*VPD

# need to define an Pcrit, i.e. a matric potential we choose where we decide conductance is effectively ze
# We use this to get Ecrit, i.e. maximum transpiration beyond which leads to runaway cavitation
P_crit <- 4 # 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:

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

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

```
for(j in 1:length(predawn_soil_mat_pot))
  {
    for(i in 1:1000)
      {
        ffx <- integrate(sperry_cond, predawn_soil_mat_pot[j], psi_canopy[i] )
```

```
        cum_can_transport[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] <-  sperry_cond(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(sperry_cond, predawn_soil_mat_pot[j], P_crit )
    E_crit[j] <- pmax(0, ffx$value)
    evap_demand[j] <- G[j]*VPD
    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 <- sperry_cond(0)

max_slo_sp <- rep(0, length=5)


for(j in 1:5)
```

```r
{
  max_slo_sp[j] <- sperry_cond(predawn_soil_mat_pot[j])
}

for(j in 1:5)
{
  for(i in 1:1000)
  {

  #  demandx[i] <- cum_can_transportx[i,1]
   psi_1[,j] <- seq(0, max(psi_canopy), length=1000) #psi_leaf[i]
   slope_demand[i,j] <- sperry_cond(psi_1[i,j])

   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(sperry_cond, predawn_soil_mat_pot[j], psi_1[i,j] )
   non_regulated_trans[i,j] <- pmax(0, ffx$value)
   non_regulated_Gs[i,j] <- non_regulated_trans[i,j]/VPD # E = G*VPD ---- G = E/VPD (VPD=1, 0.001 transfor
  #
   ffx <- integrate(sperry_cond, predawn_soil_mat_pot[j], predawn_soil_mat_pot[j] + reg_leaf_psi[i,j] )
   regulated_trans[i,j] <- pmax(0, ffx$value)
   regulated_Gs[i,j] <- regulated_trans[i,j]/VPD # E = G*VPD ---- G = E/VPD (VPD=1, 0.001 transforms to MP
   G[i,j] <- G[i,j]*loss_fun_sp_gs[i,j]

  }
}

demand_place_holder <- rep(0, length=5)
supply_place_holder <- rep(0, length=5)
min_diff_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[1]

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

    b <-which(regulated_trans[,i] <=  non_regulated_trans[demand_place_holder[i]])
    supply_place_holder[i] <-b[length(b)]

    print(demand_place_holder[i])
    print(supply_place_holder[i])
}

## [1] 801
## [1] 1000
## [1] 801
## [1] 1000
## [1] 801
## [1] 1000
```

```
## [1] 801
## [1] 1000
## [1] 801
## [1] 1000

# holder_4_max <- rep(0, length=5)
# holder_psi_max_threshold <- rep(0, length=5)
# max_leaf_regulation <- rep(0, length=5)

##---------------------------------------------------------------------------------------------------------
##---------------------------------------------------------------------------------------------------------
##---------------------------------------------------------------------------------------------------------
#
# for(i in 1:5)
#   {
#     holder_4_max[i] <- which(reg_leaf_psi[,i]==max(reg_leaf_psi[,i]))
#     holder_psi_max_threshold[i] <- psi_1[holder_4_max[i]]
#     max_leaf_regulation[i] <- max(reg_leaf_psi[,i])
#     print("--------------------------------")
#     print("i")
#     print(i)
#     print("max_leaf_regulation[i]")
#     print(max_leaf_regulation[i])
#
#     print("holder_psi_max_threshold[i]")
#     print(holder_psi_max_threshold[i])
#
#     # positions on transport curve where supply can meet demand
#     # this throws a warning but returns the first number which is what I want
#     demand_met_at_sperry[i] <- which(cum_can_transport[,i] >= evap_demand[i])
#     # the matric potential where demand is met
#     psi_demand_met_at_sperry[i] <- psi_canopy[demand_met_at_sperry[i]]
#     print("psi_demand_met_at_sperry[i]")
#     print(psi_demand_met_at_sperry[i])
#
#     if(psi_demand_met_at_sperry[i] < holder_psi_max_threshold[i])
#     {
#     # slope of tangent to supply/conductance curve where demand is met
#     demand_met_at_slope_sperry[i] <- sperry_cond(psi_demand_met_at_sperry[i])
#     # this is sperrys loss function (slope where demand is met / max slope)
#     loss_function_sperry[i] <- demand_met_at_slope_sperry[i] / cond_max_slope_sperry
#     # this is the adjusted leaf matric potential dP = dP' * (demand_met_at_slope_sperry / cond_max_slope
#     regulated_leaf_psi[i] <- ((psi_demand_met_at_sperry[i] - predawn_soil_mat_pot[i])*loss_function_sper
#     # this is the adjusted leaf matric potential dP = dP' * (demand_met_at_slope_sperry / cond_max_slope
#     # dP' = predawn_matric_potential - unregulated_matric_potential, this doesn't work unless dP is adde
#     }
#
#     if(psi_demand_met_at_sperry[i] < holder_psi_max_threshold[i])
#     {
#       regulated_leaf_psi[i] <-  max_leaf_regulation[i]
#     }
#
#     ffx <- integrate(sperry_cond, predawn_soil_mat_pot[j], predawn_soil_mat_pot[j] + regulated_leaf_psi[
#     regulated_transpiration[i] <- pmax(0, ffx£value) # regulated transpiration (E in Sperry)
#
#     # G[i] <- G[i]*loss_function_sperry[i] # regulated stomatal conductance
#     # E = G * VPD so G = E/VPD

#  }
```

From Sperry (2016) "Mathematically, P rises to a maximum before decreasing back to zero as E 0 increases to

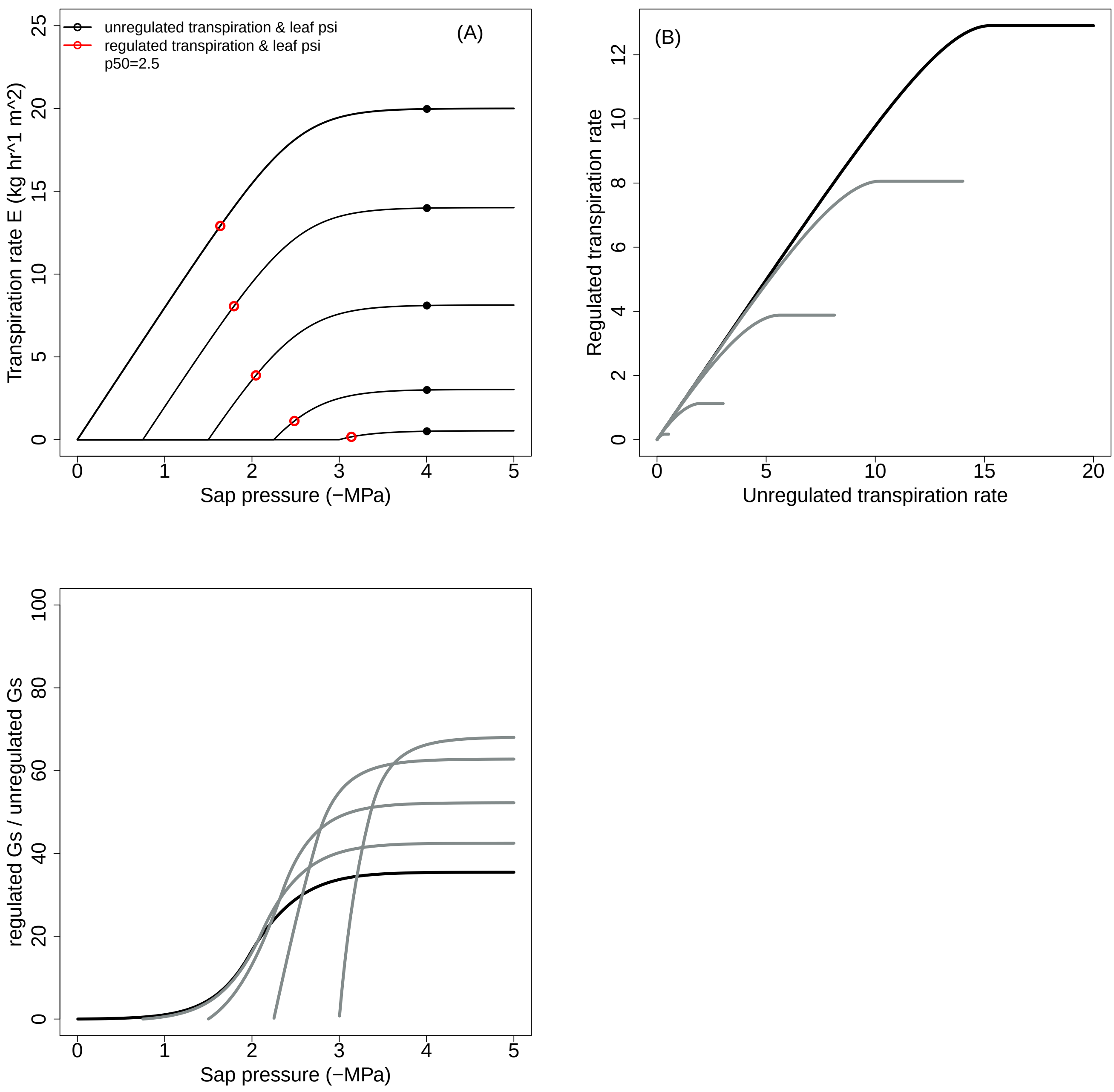

Figure 1: Hydraulic conductance and transpitation as a function of sap xylem pressure.

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 (Fig.2).