
HydroShoot Documentation

Release 1.0.0

Rami Albasha

Jul 18, 2023

Contents:

1	Installation	1
2	Overview	3
2.1	What is HydroShoot about?	3
2.2	Why would you be interested in it?	4
3	Model description	5
3.1	Ecophysiological processes	5
3.1.1	Hydraulic structure	5
3.1.2	Gas exchange	7
3.1.3	Energy budget	10
3.2	Processes interactions	11
3.3	References	12
4	Usage	15
4.1	Set the required files up	16
4.2	Outputs	16
4.3	Run the model	17
5	Further reading	19
5.1	Credits	19
5.1.1	Development Lead	19
5.2	History	19
5.2.1	creation	19
5.3	Contributing	19
5.3.1	Types of Contributions	20
5.3.2	Get Started!	20
5.3.3	Pull Request Guidelines	21
5.3.4	Tips	21

CHAPTER 1

Installation

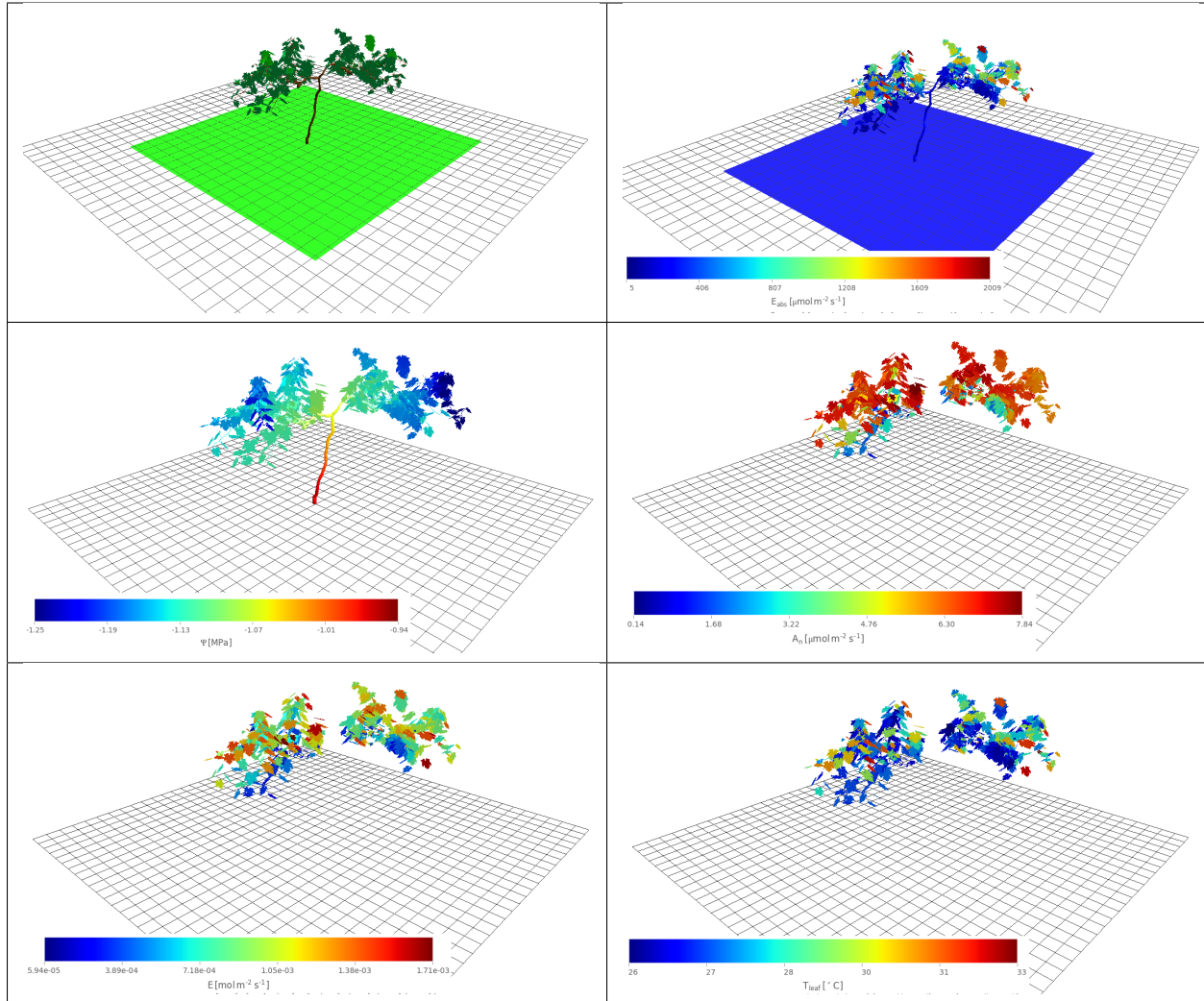
HydroShoot can be installed automatically using the openalea3 channel from anaconda:

```
conda create -n hydroshoot -c openalea3 -c conda-forge openalea.  
hydroshoot conda activate hydroshoot
```

You're done !

2.1 What is HydroShoot about?

HydrShoot is a functional-structural plant model that simulates the interactions between shoot's hydraulic structure, gas-exchange and energy-budget, at the organ level.



Plant shoot is static to HydroShoot. That is the model does not simulate plant growth nor its development. Its usage would therefore be only sound only if shoot's structure can be assumed constant across a given span of time (let's say a week or so for grapevine).

2.2 Why would you be interested in it?

HydroShoot may be useful for example if you would like to explore:

- how leaf net carbon assimilation rate (A_n), transpiration rate (E), temperature (T) and bulk water potential (Ψ_{leaf}) interact across the canopy
- how soil water status impacts A_n , E , T , and Ψ_{leaf}
- how the structure of the shoot impacts A_n , E , T , and Ψ_{leaf}
- how soil water potential (Ψ_{soil}) propagates across plant's shoot

All information in this section are taken from HydroShoot's paper (Albasha et al., 2019). Please refer to the this paper for a detailed description of model's formulae and evaluation.

3.1 Ecophysiological processes

3.1.1 Hydraulic structure

The *hydraulic* module computes the distribution of xylem water potential across plant segments by analogy to Ohm's law (Fig. 3.1)

Water flux (F , $kg\ s^{-1}$) across the hydraulic segment of length (L , m) is driven by the difference of xylem pressures across this segment ($H_u - H_l$, MPa) and regulated by segment's hydraulic conductivity (K , $kg\ s^{-1}\ m\ MPa^{-1}$):

$$F = -K \cdot \frac{H_u - H_l}{L}$$

The hydraulic conductivity varies with water potential as a result of xylem cavitation under water deficit (Tyree and Sperry, 1989). This relationship is described in HydroShoot as:

$$K = K_{max} \cdot \frac{1}{\left(1 + \left(\frac{\Psi}{\Psi_{crit, stem}}\right)^{c_{x1}}\right)}$$

where K_{max} [$kg\ s^{-1}\ m\ MPa^{-1}$] is the maximum conductivity of the segment, Ψ [MPa] is the arithmetic mean of xylem potential of the segment, $\Psi_{crit, stem}$ [MPa] and c_{x1} [–] are shape parameters.

K_{max} is estimated empirically as proposed by Tyree and Zimmermann (2002) as:

$$K_{max} = c_{x2} \cdot D^{c_{x3}}$$

where D [m] is the average diameter of the segment, and c_{x2} and c_{x3} are shape parameters, mostly given within the ranges of [2.5, 2.8] and [2.0, 5.0], respectively.

The last two equations apply to all conducting segments (not leaves blades). Water potential of the upper extremity of the petiole is assumed equal to that of the lumped leaf water potential Ψ_{leaf} [MPa].

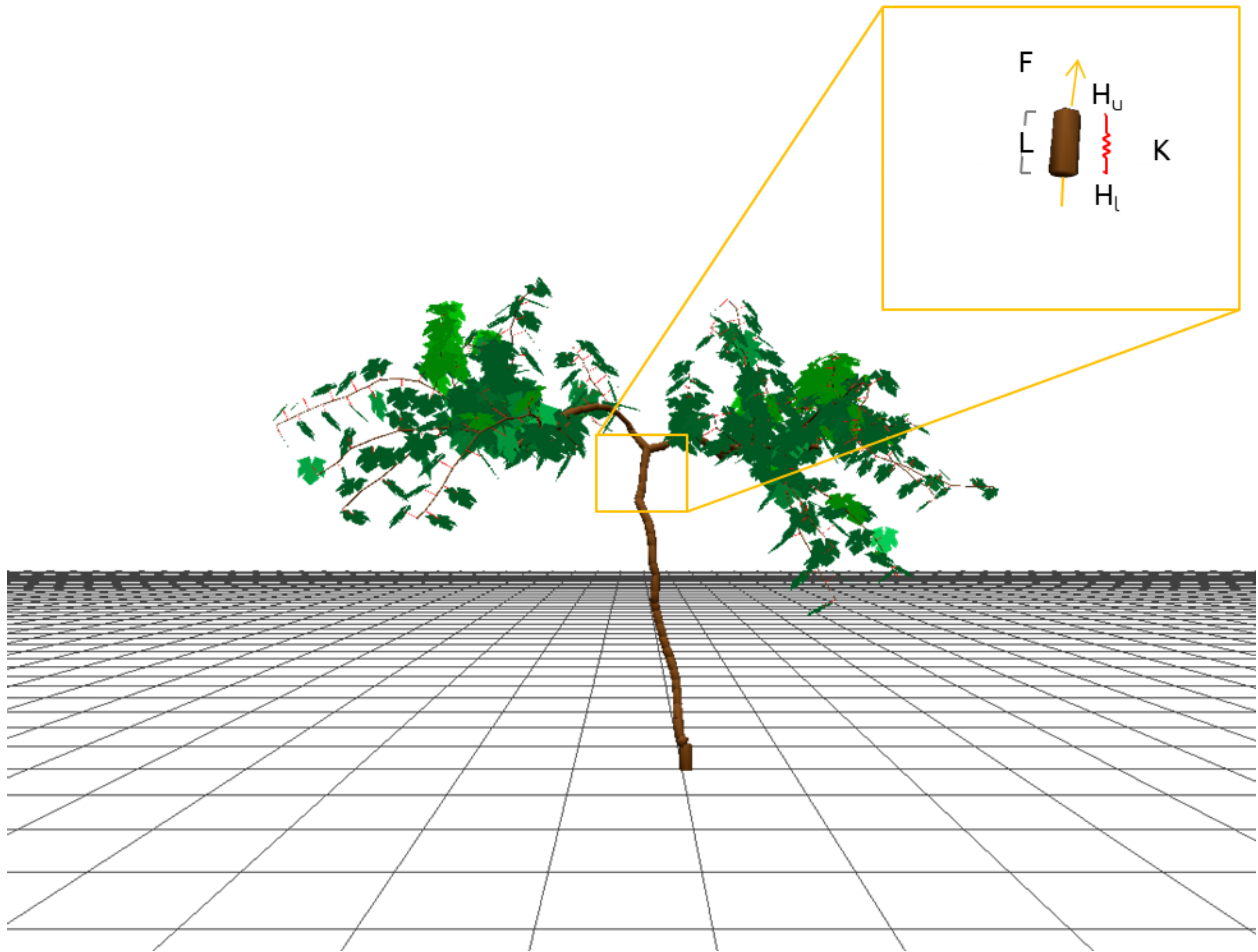
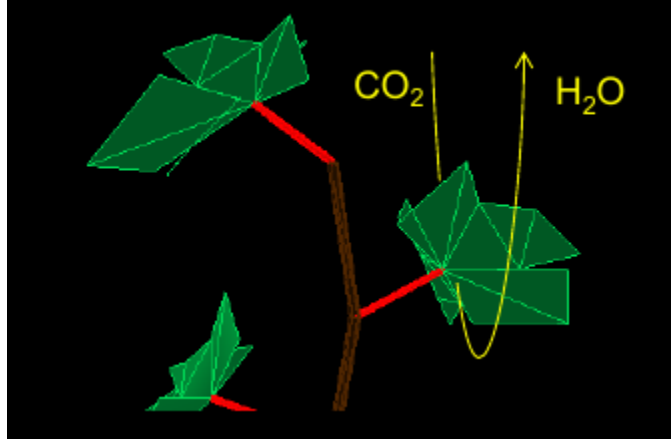


Fig. 3.1: Illustration of the variables required to calculate the hydraulic structure: water flux (F), hydraulic conductivity (K), water pressure at upper (downstream) and lower (upstream) extremities of the conducting element (respectively H_u and H_l), and the length of the segment (L).

3.1.2 Gas exchange



The *exchange* module computes the rates of net carbon assimilation of C3 plants (A_n), stomatal conductance to CO_2 (g_s, CO_2) and water vapor (g_s, H_2O), and transpiration (E) per unit leaf surface area as a function of micrometeorological conditions and leaf water status.

A_n and g_s, CO_2 are coupled based on the analytical solution proposed by **Yin et al. (2009)** which also considers mesophyll conductance to CO_2 diffusion g_m . The solution is based on the following equations (following **Evers et al. 2010** supporting information):

$$A_n = \frac{(C_c - \Gamma) \cdot x_1}{C_c + x_2} - R_d$$

$$C_c = C_i - \frac{A_n}{g_m}$$

$$g_s, CO_2 = g_{s0, CO_2} + m_0 \cdot \frac{A_n + R_d}{C_i - \Gamma} \cdot f_w$$

$$g_s, CO_2 = \frac{A_n}{C_a - C_i - A_n \cdot r_{tb}}$$

where A_n [$\mu mol\ m^{-2}\ s^{-1}$] is net carbon assimilation rate, R_d [$\mu mol\ m^{-2}\ s^{-1}$] is mitochondrial respiration in the light, Γ [μbar] is CO_2 compensation point in the absence of mitochondrial respiration, x_1 [$\mu mol\ m^{-2}\ s^{-1}$] and x_2 [μbar] are intermediate parameters, g_m [$\mu mol\ m^{-2}\ s^{-1}\ \mu bar^{-1}$] is mesophyll conductance for CO_2 diffusion, g_s, CO_2 [$mol\ m^{-2}\ s^{-1}\ \mu bar^{-1}$] is stomatal conductance to CO_2 , g_{s0, CO_2} [$mol\ m^{-2}\ s^{-1}\ \mu bar^{-1}$] is the residual stomatal conductance to CO_2 , m_0 [–] is a shape parameter regulating the slope between A_n and g_s, CO_2 , f_w [–] is a dimensionless function representing the response of stomatal conductance to soil or plant water status, r_{tb} [$m^2\ s\ \mu bar\ \mu mol^{-1}$] is the combined turbulence and boundary layer resistance for CO_2 , C_a [μbar] is air CO_2 partial pressure, C_i [μbar] is intercellular CO_2 partial pressure, and C_c [μbar] is chloroplast CO_2 partial pressure.

Net carbon assimilation rate per unit leaf area

A_n is given as:

$$A_n = V_c \cdot \left(1 - \frac{\Gamma}{C_c}\right) - R_d$$

where V_c [$\mu mol\ m^{-2}\ s^{-1}$] is carboxylation rate, Γ [$\mu mol_{CO_2}\ mol_{CO_2}^{-1}$] CO_2 compensation point in the absence of mitochondrial respiration, C_c [$\mu mol_{CO_2}\ mol^{-1}$] chloroplast CO_2 concentration, and R_d [$\mu mol\ m^{-2}\ s^{-1}$] is mitochondrial respiration rate in the light.

V_c is given as:

$$V_c = \min(W_c, W_j, W_p)$$

where W_c , W_j , and W_p [$\mu\text{mol m}^{-2} \text{s}^{-1}$] are respectively Rubisco-limited carboxylation rate, RuBP-limited carboxylation rate, and Triose phosphates-limited carboxylation rate, given as:

$$W_c = \frac{C_c \cdot V_{c, \max}}{C_c + K_c \cdot \left(1 + \frac{O}{K_o}\right)}$$

$$W_j = \frac{J}{4 + 8 \cdot \frac{C_c}{C_c}}$$

$$W_p = \frac{3 \cdot TPU}{\left(1 - \frac{C_c}{C_c}\right)}$$

where $V_{c, \max}$ [$\mu\text{mol m}^{-2} \text{s}^{-1}$] is the maximum carboxylation rate, J [$\mu\text{mol m}^{-2} \text{s}^{-1}$] electron transport rate, TPU [$\mu\text{mol m}^{-2} \text{s}^{-1}$] Triose phosphates transport rate, K_c [$\mu\text{mol mol}^{-1}$] Michaelis-Menten constant for the carboxylase, K_o [mmol mol^{-1}] Michaelis-Menten constant for the oxygenase, and O [mmol mol^{-1}] oxygen concentration.

Finally, J is given as:

$$J = \frac{\alpha \cdot PPFD}{\sqrt{1 + \frac{\alpha^2 \cdot PPFD^2}{J_{\max}^2}}}$$

where J_{\max} [$\mu\text{mol m}^{-2} \text{s}^{-1}$] is maximum electron transport rate, and $PPFD$ [$\mu\text{mol m}^{-2} \text{s}^{-1}$] is photosynthetic photon flux density.

The impact of leaf temperature on the photosynthesis parameters is accounted for using Arrhenius functions. For $V_{c, \max}$, J_{\max} , TPU , and R_d temperature dependency writes:

$$P = P^{25} \cdot \frac{\exp\left(c - \frac{\Delta H_a}{R \cdot T_{leaf}}\right)}{1 + \exp\left(\frac{\Delta S \cdot T_{leaf} - \Delta H_d}{R \cdot T_{leaf}}\right)}$$

where P denotes any of $V_{c, \max}$, J_{\max} , TPU , and R_d parameters, P^{25} is the value of P at 25°C , c $[-]$ is a shape parameter, ΔH_a [$\text{kJ mol}_{CO_2}^{-1}$] is activation energy of the Arrhenius functions, ΔH_d [$\text{kJ mol}_{CO_2}^{-1}$] is deactivation energy of the Arrhenius functions, ΔS [$\text{kJ K}^{-1} \text{mol}_{CO_2}^{-1}$] is entropy term, R [$\text{kJ K}^{-1} \text{mol}^{-1}$] is the ideal gas constant, and T_{leaf} [K] is leaf temperature.

Finally, for Γ , K_c , and K_o temperature dependency writes:

$$P = \exp\left(c - \frac{\Delta H_a}{R \cdot T_{leaf}}\right)$$

Variable intra-canopy photosynthetic capacities

Leaf photosynthetic traits (V_{cmax} , J_{max} , TPU and R_d ; cf. Appendix I in **Albasha et al., 2019**) are set to vary as a function leaf nitrogen content per unit leaf surface area (N_a , $g_N \text{m}^{-2}$) following **Prieto et al. (2012)**:

$$P^{25} = S_{N_a} N_a - b_{N_a}$$

where P^{25} [$\mu\text{mol m}^{-2} \text{s}^{-1}$] is the value at 25°C for any of the rates of V_{cmax} , J_{max} , TPU or R_d , S_{N_a} [$\mu\text{mol g}_N^{-1} \text{s}^{-1}$] and b_{N_a} [$\mu\text{mol m}^{-2} \text{s}^{-1}$] are the slope and the intercept of the linear relationship with N_a specific to each rate.

N_a is calculated as the product of nitrogen content per unit leaf dry mass (N_m , $g_N g_{drymatter}^{-1}$) and leaf dry mass per area (LMA , $g_{drymatter} \text{m}^{-2}$). N_m linearly varies with plant age, expressed as the thermal time cumulated since budburst, and LMA is determined by leaf exposure to light during the last past days (**Prieto et al., 2012**), as expressed respectively in the two following equations:

$$N_m = a_N \cdot \sum_{i=budburst}^d (\max(0, T_{air, i} - T_b)) + b_N$$

$$LMA = a_M \cdot \ln(PPFD_{10}) + b_M$$

where $T_{air, i}$ [$^{\circ}C$] is the mean temperature of the day i , T_b [$^{\circ}C$] is the base temperature (minimum required for growth), set to $10^{\circ}C$ for grapevine and used for the calculation of thermal time since budburst, a_N [$g_N g_{drymatter}^{-1} ^{\circ}C d^{-1}$] and b_N [$g_N g_{drymatter}^{-1}$] are the slope and intercept of the linear relationship between N_m and accumulated thermal time since budburst, $PPFD_{10}$ [$mol_{photon} m^{-2} d^{-1}$] is the cumulative photosynthetic photon flux density irradiance intercepted by the leaf (output of the energy module) averaged over the past 10 days, a_M [$g_{drymatter} mol_{photon}^{-1} d^{-1}$] and b_M [$g_{drymatter} m^{-2}$] are the slope and intercept of the linear relationship between LMA and the logarithm of $PPFD_{10}$.

Photoinhibition

HydroShoot is provided with an empirical photoinhibition model which assumes that combined heat and water stresses inhibit photosynthesis by reducing the electron transport rate (J):

$$\Delta H_d = \Delta H_{d, max} - \max \left(0, (\Delta H_{d, max} - \Delta H_{d, T}) \cdot \min \left(1, \frac{\Psi_{leaf} - \Psi_{leaf, max}}{\Psi_{leaf, min} - \Psi_{leaf, max}} \right) \right)$$

$$\Delta H_{d, t} = \Delta H_{d, t1} - (\Delta H_{d, t1} - \Delta H_{d, t2}) \cdot \min \left(1, \max \left(0, \frac{T_{leaf} - T_{leaf1}}{T_{leaf2} - T_{leaf1}} \right) \right)$$

where ΔH_d [$kJ mol^{-1}$] is calculated after accounting for the joint effects of leaf water potential Ψ_{leaf} MPa and temperature T_{leaf} [K], $\Delta H_{d, max}$ [$kJ mol^{-1}$] is the value of ΔH_d without accounting for photoinhibition, $\Delta H_{d, T}$ [$kJ mol^{-1}$] is the value of ΔH_d after accounting for the effect of T_{leaf} , $\Psi_{leaf, max}$ and $\Psi_{leaf, min}$ [MPa] are leaf water potential values at which photoinhibition starts and reaches its maximum effect, respectively, finally, $\Delta H_{d, T1}$ and $\Delta H_{d, T2}$ [$kJ mol^{-1}$] are empirical thresholds corresponding to leaf temperatures T_{leaf1} and T_{leaf2} which are temperatures at which photoinhibition starts and reaches its maximum effect, respectively.

Transpiration rate per unit leaf area

The transpiration rate (E , $mol m^{-2} s^{-1}$) is calculated as:

$$E = \frac{1}{\frac{1}{g_{b, H_2O}} + \frac{1}{1.6 \cdot g_{s, CO_2}}} \left(\frac{VPD}{P_a} \right)$$

where P_a [kPa] is the atmospheric pressure and g_{b, H_2O} [$mol m^{-2} s^{-1}$] is the boundary layer conductance to water vapor derived from **Nobel (2005)** as:

$$g_{b, H_2O} = \frac{D_{H_2O} \cdot P_v}{R \cdot T_{leaf} \cdot \Delta x}$$

with

$$D_{H_2O} = D_{H_2O, 0} \cdot \frac{P_a}{P_v} \cdot \left(\frac{T_{leaf}}{273} \right)^{1.8}$$

where D_{H_2O} is the diffusion coefficient of H₂O in the air at $0^{\circ}C$ ($2.13 \cdot 10^{-5} m^2 s^{-1}$), P_a [MPa] is the ambient air pressure at $0^{\circ}C$, P_v [MPa] is water vapor partial pressure, and Δx [m] is the thickness of the boundary layer defined following **Nobel (2005)** as:

$$\Delta x = 0.004 \sqrt{\frac{l}{v}}$$

where l [m] is the mean length of the leaf in the downwind direction (set to 70% of blade length), and v [$m s^{-1}$] is wind speed in the vicinity of the leaf.

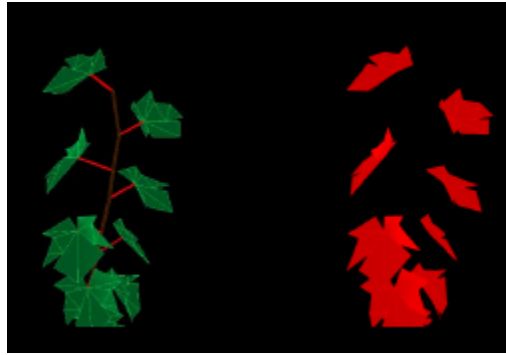
Finally, the impact of water stress on stomatal conductance (i.e. via the f_w function) is calculated using one of the following options:

$$f_w = \begin{cases} \frac{1}{1 + \left(\frac{VPD}{D_0}\right)} & (a) \\ \frac{1}{1 + \left(\frac{\Psi_{leaf}}{\Psi_{crit, leaf}}\right)^n} & (b) \\ \frac{1}{1 + \left(\frac{\Psi_{soil}}{\Psi_{crit, leaf}}\right)^n} & (c) \end{cases}$$

where VPD [kPa] is vapor pressure deficit (between the leaf and the air), D_0 [kPa] shape parameter, Ψ_{leaf} [MPa] leaf bulk xylem potential, Ψ_{soil} [MPa] soil bulk water potential (assumed equal to xylem potential at the base of the shoot), and $\Psi_{crit, leaf}$ [MPa] leaf water potential at which stomatal conductance reduces to half its maximum value.

In case the option *a* is used, stomatal conductance reduction is considered independent from the soil water status (i.e. following **Leuning, 1995**). In contrast, Both options *b* and *c* allow simulating stomatal conductance as a function either of leaf water potential (i.e. regarding shoot hydraulic structure) or soil water potential (i.e. disregarding the hydraulic structure of the shoot).

3.1.3 Energy budget



The *energy* module computes the temperature of individual leaves based on a detailed energy balance model (see Supporting Information S3 in **Albasha et al., 2019**).

Energy gain of each leaf comes from:

1. the absorbed shortwave;
2. thermal longwave radiation from the sky;
3. thermal longwave radiation from the sky;
4. thermal longwave radiation from the soil;
5. thermal longwave radiation from the neighbouring leaves.

Energy loss of each leaf is due to

1. thermal longwave radiation emitted by the leaf
2. latent heat due to transpiration (evaporative cooling)

Energy gain or loss may result from heat exchange between the each leaf and the surrounding air by thermal conduction-convection.

The resulting leaf-scale energy balance equation writes:

$$\begin{aligned}
 0 = & \alpha_{i, R_g} \cdot \Phi_{i, R_g} \\
 & + \epsilon_{i, leaf} \cdot \sigma \cdot (k_{i, sky} \cdot \epsilon_{sky} \cdot T_{sky}^4 + k_{i, soil} \cdot \epsilon_{soil} \cdot T_{soil}^4 + [1 - (k_{sky} + k_{soil})] \cdot T_i^4) \\
 & - 2 \cdot \epsilon_{i, leaf} \cdot \sigma \cdot T_i^4 \\
 & - \lambda \cdot E_i \\
 & - 2 \cdot K_{air} \cdot \frac{T_i - T_{air}}{\Delta x_i}
 \end{aligned}$$

where i refers to leaf identifier, j refers to neighbouring leaves identifier, Ω denotes the upper hemisphere surrounding the leaf i , α_{R_g} [–] is lumped leaf absorptance in the shortwave band, Φ_{R_g} [$W m_{leaf}^{-2}$] flux density of shortwave global irradiance, ϵ_{leaf} [–] emissivity-absorptivity coefficients of the leaf, ϵ_{sky} [–] emissivity-absorptivity coefficients of the leaf, ϵ_{soil} [–] emissivity-absorptivity coefficients of the soil, λ [$W s mol^{-1}$] is latent heat for vaporization, σ [$W m^{-2} K^{-4}$] the Stefan-Boltzmann constant, k_{sky} [–] form factor of the sky, k_{soil} [–] form factor of the soil, T [K] leaf temperature, T_{air} air temperature, T_{sky} [K] sky temperature, T_{soil} [K] soil temperature, K_{air} [$W m^{-1} K^{-1}$] the thermal conductivity of air, E [$mol m_{leaf}^{-2} s^{-1}$] transpiration flux, and Δx_i [m] thickness of the boundary layer.

Note: Only the forced convective heat transfer is currently considered in HydroShoot since forced convection dominates free convection once wind speed exceeds roughly $0.1 m s^{-1}$ (Nobel 2005). This wind speed threshold is generally exceeded during diurnal hours. However, under low wind conditions heat transfer may be underestimated.

Sky and soil form factors: the *Pirouette Cacahuete* issue!

You may notice once you do your first run something like this:

```

GDD since budbreak = 1000 °Cd
Energy_budget: True
Hydraulic structure: True
Computing form factors...
... pirouette
... cacahuete
Soil class: Sand

```

This refers to the method used to calculate the lumped sky and soil form factors (respectively k_{sky} and k_{soil}).

In order to reduce calculation costs, k_{sky} and k_{soil} are obtained by flip flopping the canopy:

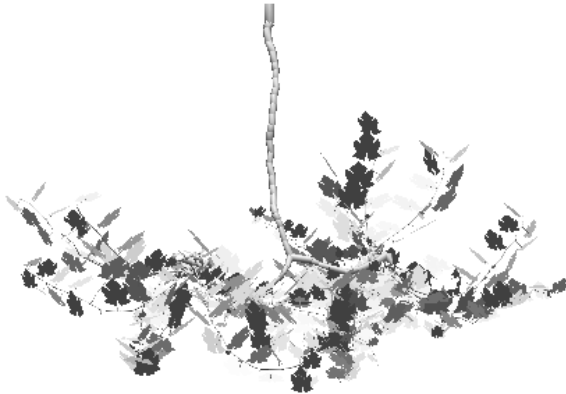
At first, the canopy is turned downwards. A unit irradiance is emitted from each sky sector and irradiance that is intercepted by a leaf i is assumed equivalent to the form factor between that leaf and the “soil”. In the second step, the canopy is turned upwards again and similarly, a unit irradiance is emitted from each sky sector. In this case, irradiance that is intercepted by a leaf i is assumed equivalent to the form factor between that leaf and the “sky”.

This method is clearly not 100% precise. It may need further improvements in the future.

3.2 Processes interactions

Interactions among the physiological processes occur.

Pirouette



Cacahuète



Fig. 3.2: Solving the interactions between leaf's water potential, transpiration rate and temperature !

From the one hand. Leaf water potential (Ψ) exerts a control on stomatal conductance to water vapor (g_{s, H_2O}) and, consequently, on transpiration flux (E). However, the latter also determines how much water will flow through the hydraulic segments to be withdrawn from the soil, which affects the distribution of water potential across those segments (Ohm's law). Hence, we have a reciprocity between the distribution of water potential across the shoot (hydraulic structure) and transpiration fluxes from individual leaves.

From the other hand, leaf temperature (T) determines the rate of leaf photosynthesis (Arrhenius functions) and stomatal conductance to both CO_2 (g_{s, CO_2}) and water vapor (g_{s, H_2O}). This means that T affects also the transpiration flux E which also means that T exerts a further control on the hydraulic structure.

Putting it all together:

For a given initial temperature (T_0) and soil water potential (Ψ_0), a leaf can fix CO_2 with a rate $A_{n, 0}$ while having a stomatal conductance rate of $g_{s, H_2O, 0}$. The corresponding water flux transpired by this leaf will be E_0 . Summed up across all leaves, plant transpiration will withdraw water from the soil, reducing thus soil water potential and resulting in a new Ψ value (Ψ_1). Ψ_1 , will hence impose a new stomatal conductance $g_{s, H_2O, 1}$ and consequently a new transpiration flux E_1 . Yet, this new E_1 means that leaf temperature is T_1 ... **REWIND!**

HydroShoot resolve this interactions iteratively until reaching steady rate values of A_n , and E (implying steady state distribution of water potential cross the shoot). Fig. 3.3 shows how interactions between the hydraulic, energy and gas-exchange processes is handled in HydroShoot.

3.3 References

- Albasha R, Fournier C, Pradal C, Chelle M, Prieto J, Louarn G, Simonneau T, Lebon E. 2019.** HydroShoot: a functional-structural plant model for simulating hydraulic structure, gas and energy exchange dynamics of complex plant canopies under water deficit - application to grapevine (*Vitisvinifera* L.) in silico Plants, diz007, <https://doi.org/10.1093/insilicoplants/diz007>
- Evers J, Vos J, Yin X, Romero P, van der Putten P, Struik P. 2010.** Simulation of wheat growth and development based on organ-level photosynthesis and assimilate allocation. Journal of Experimental Botany 61: 2203–2216

- Farquhar G, von Caemmerer S, Berry J. 1980.** A biochemical model of photosynthetic CO₂ assimilation in leaves of C₃ species. *Planta* 149: 78–90.
- Leuning R. 1995.** A critical appraisal of a combined stomatal-photosynthesis model for C₃ plants. *Plant, Cell and Environment* 18: 339–355.
- Nobel P. 2005.** Temperature and energy budgets. In Nobel S, eds. *Physicochemical and Environmental Plant Physiology*. Elsevier Academic Press.
- Prieto J, Louarn G, Perez Peña J, Ojeda H, Simonneau T, Lebon E. 2012.** A leaf gas exchange model that accounts for intra-canopy variability by considering leaf nitrogen content and local acclimation to radiation in grapevine (*Vitis vinifera* L.) *Plant, Cell and Environment* 35: 1313–1328.
- Tyree M, Sperry J. 1989.** Vulnerability of xylem to cavitation and embolism. *Annual review of plant physiology and plant molecular biology* 40: 19–38.
- Tyree M, Zimmermann M. 2002.** Xylem structure and the ascent of sap, Springer Series in Wood Science.
- Yin X, Struik P. 2009.** C₃ and C₄ photosynthesis models: An overview from the perspective of crop modelling. *NJAS - Wageningen Journal of Life Sciences* 57: 27–38.

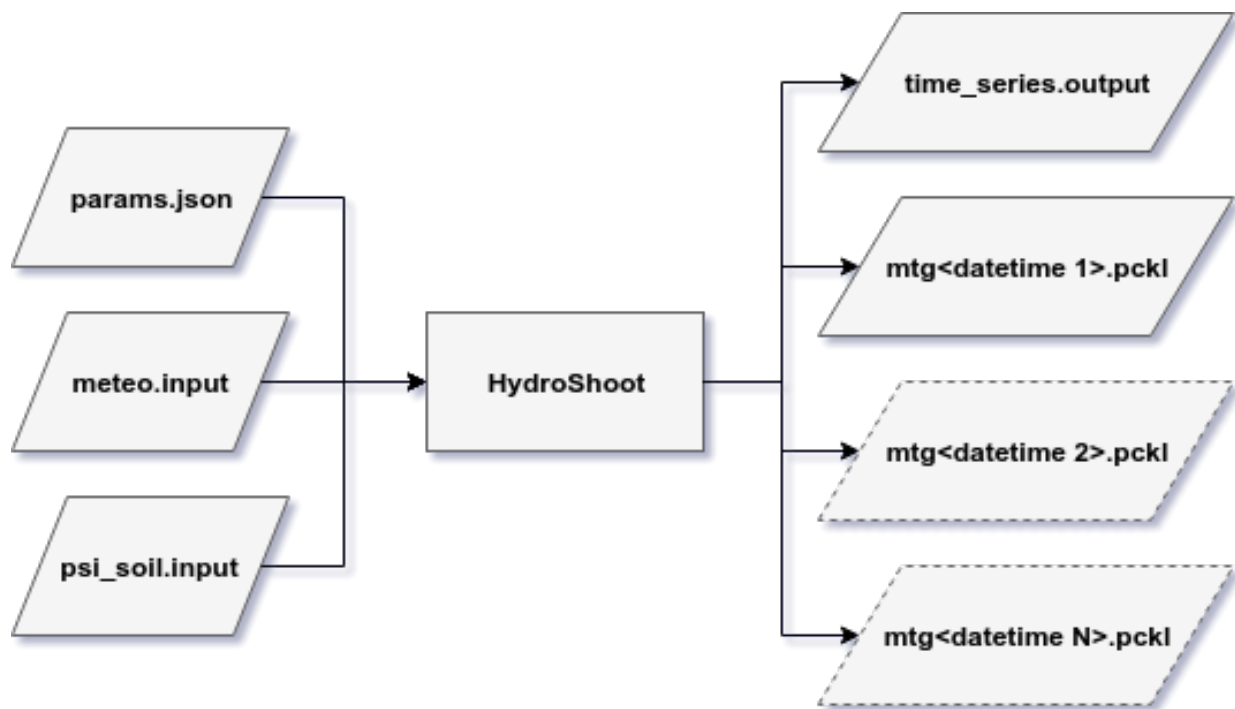


Fig. 4.1: Illustration of inputs and outputs of HydroShoot.

Three input files are required in order to run HydroShoot. The first contains all model parameters, the second meteorological data, and the third (optional) soil predawn water potential data. The model returns an undetermined number of files (≥ 3) of two types of outputs.

4.1 Set the required files up

1. Model parameters

Almost all model parameters are setup in the **params.json** file. *params.json* is a *JavaScript Object Notation (JSON)* file that contains the following sections:

- “simulation”
- “planting”
- “phenology”
- “mtg_api”
- “numerical_resolution”
- “irradiance”
- “energy”
- “hydraulic”
- “exchange”
- “soil”

Each of the above sections refers to its homonym process. An exhaustive and thoroughly detailed description of all the parameters inside each section, their units and their expected values, are given in the *params_schema.json* file (src/hydroshoot/). The reader is encouraged to refer to this file to know how exactly to fill up the *params.json* file.

2. Meteorological data (time series)

Meteorological data must be provided using a .csv file whose name must be given in the *params.json* file (“me-teo” parameters). This file must contain the following columns: * *time*: a *datetime* string having the format YYYY-MM-DD HH:MM:SS * *Tac*: air temperature [$^{\circ}C$] * *hs*: relative humidity (%) * *u*: wind speed [$m\ s^{-1}$] * *Rg*: solar radiation (shortwave irradiance) given even in [$W_{global}\ m_{ground}^{-2}$], [$W_{PAR}\ m_{ground}^{-2}$] or [$\mu mol_{PPFD}\ m_{ground}^{-2}\ s^{-1}$]. The user must provide the unit of *Rg* in the *params.json* file (“E_type” parameter).

3. Soil predawn water potential data (time series)

There are two ways to supply predawn soil water potential (Ψ_{soil}) to the model:

- using an external file that must be named “psi_soil.input” (the user will be able to give other names to the file in the future).
- using the keyword argument “psi_soil” when calling the simulator:

```
from hydroshoot import model

model.run(g, working_directory, psi_soil=-0.5)
```

In this case, the value given in the ‘psi_soil’ keyword overrides **all values** given in ‘psi_soil.input’ file. HydroShoot will force soil water potential value **at each time step** to that given by the keyword ‘psi_soil’.

4.2 Outputs

The first output file is a *comma-separated values (csv)* file named “time_series.output” (only one file is returned) that contains the following time-dependent outputs:

- **time**: datetime string (YYYY-MM-DD HH:MM:SS)
- **An**: net carbon assimilation flux of the whole plant [$\mu\text{mol s}^{-1}$]
- **E**: transpiration flux of the whole plant [$g h^{-1}$]
- **Rg**: absorbed shortwave irradiance by the whole plant [$W m_{ground}^{-2}$]
- **Tleaf**: median leaf temperature [$^{\circ}C$]

The second type of outputs is a pickle (pckl) file in which is stored mtg data **for each simulated time step**. Note that **pckl files do not contain geometry data**. Geometry is not currently stored as an output since the model expects it as an input.

4.3 Run the model

Let's take the simple example provided in '~/.example/potted_grapevine/'. We will use for this example **architecture** and **display** modules provided with HydroShoot package (these packages may be removed in the future).

This example runs HydroShoot on a small potted grapevine canopy having two primary shoots.

Let's build the mock-up using the digitalization data provided in "grapevine_pot.csv" file:

```
from os import getcwd

from openalea.mtg import traversal
from openalea.plantgl.all import Scene
from hydroshoot import architecture, display, model

# Path for plant digitalization data.
g = architecture.vine_mtg('grapevine_pot.csv')

for v in traversal.iter_mtg2(grapevine_mtg, grapevine_mtg.root):
    architecture.vine_phyto_modular(grapevine_mtg, v)
    architecture.vine_mtg_properties(grapevine_mtg, v)
    architecture.vine_mtg_geometry(grapevine_mtg, v)
    architecture.vine_transform(grapevine_mtg, v)
```

Display now the scene using **display** module, you should be having something like Fig. 4.2:

```
scene = display.visu(g, def_elmnt_color_dict=True, scene=Scene(),
                    view_result=True)
```

You're now ready to run the model using the meteorological data provided in "meteo.input", and forcing soil water potential to a fixed value of $-0.5 MPa$. Ensure first that the model will run over one hour only by setting start and end dates of the simulation in the "params.json" file ("sdate": "2012-08-01 11:00:00", "edate": "2012-08-01 11:00:00").

```
model.run(g, str(getcwd()) + '/', scene, psi_soil=-0.2, gdd_since_budbreak=100.)
```

You should now have created the '/output/' folder where output files ("time_series.ouput", "mtg20120801110000.pckl") are stored. "time_series.ouput" file should contain close values to those below:

```
time;Rg;An;E;Tleaf
2012-08-01 11:00:00;69.83;7.99;179.17;26.05
```

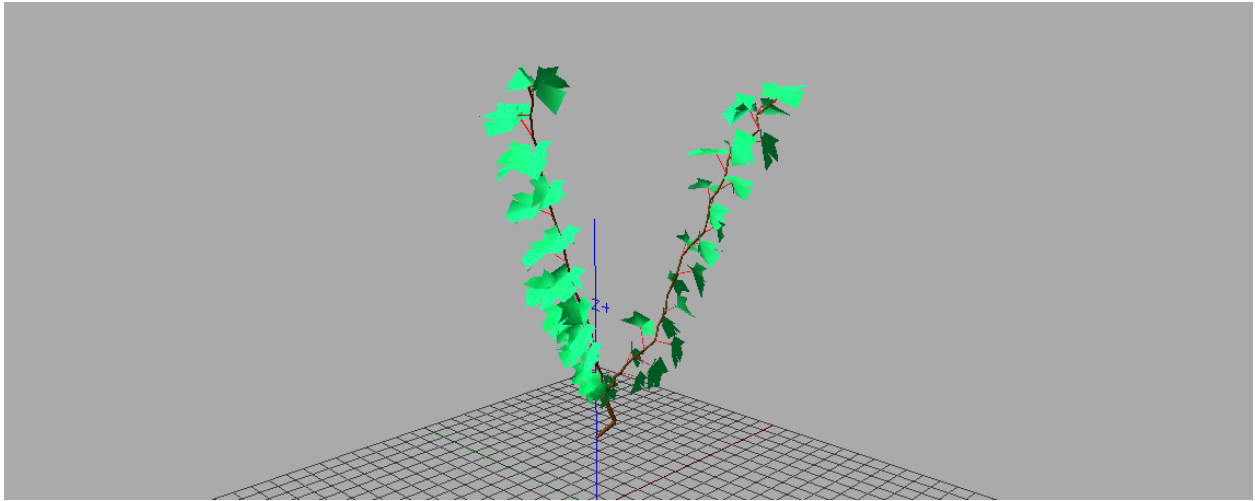


Fig. 4.2: Screen shot of the mock-up produced by the “potted_grapevine” example.



5.1 Credits

5.1.1 Development Lead

- Rami Albasha, <rami.albasha@itk.fr>
- Christian Fournier, <christian.fournier@inrae.fr>
- Christophe Pradal, <christophe.pradal@inria.fr>
- Eric Lebon

5.2 History

5.2.1 creation

HydroShoot was created by Rami ALBASHA during his postdoctoral research (July 2015 to December 2016) under the supervision of Eric LEBON†, in close collaboration with Christian FOURNIER and Christophe PRADAL.

The model is partly based on prior developments led by Gaëtan LOUARN and Jorge PRIETO who developed the FSPM TOPVINE (Louarn et al., 2008, Prieto et al., 2012). The main difference between TOPVINE and HydroShoot is that the latter simulates explicitly the hydraulic network of the shoot, which is coupled to both energy and gas exchange processes.

5.3 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

5.3.1 Types of Contributions

Report Bugs

Report bugs at [issues](#).

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

Write Documentation

pkglts could always use more documentation, whether as part of the official pkglts docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at [issues](#).

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.3.2 Get Started!

Ready to contribute? Here’s how to set up *hydroshoot* for local development.

1. Fork the *hydroshoot* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/hydroshoot.git
```

3. Install your local copy into a virtualenv. Assuming you have [virtualenv](#) installed, this is how you set up your fork for local development:


```
$ virtualenv dvlpt
$ dvlpt/script/activate
(dvlpt)$ python setup.py develop
```

4. Create a branch for local development (wip stands for work in progress):

```
(dvlpt)$ git checkout -b wip_name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
(dvlpt)$ cd hydroshoot
(dvlpt) hydroshoot$ flake8
(dvlpt) hydroshoot$ nosetests
(dvlpt) hydroshoot$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin wip_name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.3.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.4, and 3.5. Check https://travis-ci.org/moi/hydroshoot/pull_requests and make sure that the tests pass for all supported Python versions.

5.3.4 Tips

To run a subset of tests:

```
$ nosetests test/test_XXX
```