DYNAMIC MODELING OF TREE GROWTH AND ENERGY USE IN A NURSERY GREENHOUSE USING MATLAB AND SIMULINK

by

Jamison M. Hill

Submitted in partial fulfillment of the requirements
for the degree of

Masters of Engineering

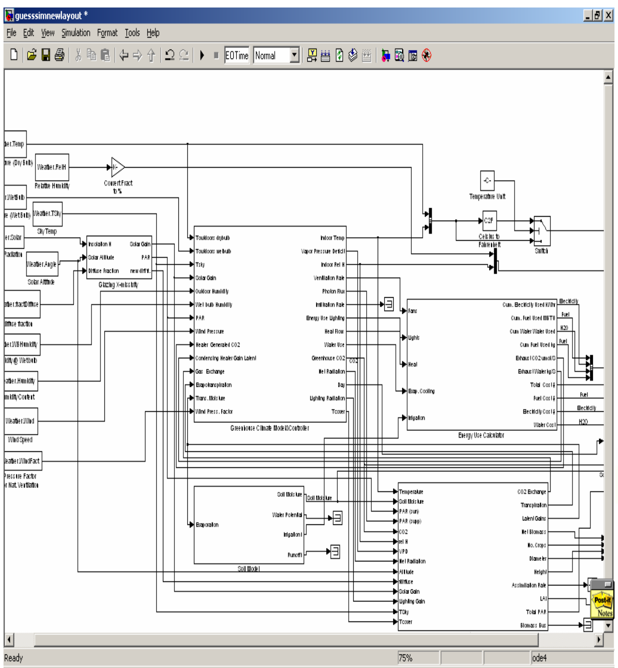
Agricultural and Biological Engineering

Cornell University

7/31/2006

Advisor: Dr. Louis D Albright

DYNAMIC MODELING OF TREE GROWTH AND ENERGY USE IN A NURSERY GREENHOUSE USING MATLAB AND SIMULINK



ACKNOWLEDGMENTS

The author wishes to thank:

Lou Albright, my advisor for all his wisdom, guidance and guidance;

my friends and family for their patience during this endeavor;

The various authors whose books proved to be invaluable resources while formulating this model

The "growers": Jeff Mehlschau and Don Reagan for sharing their time and extensive knowledge

COPYRIGHT INFORMATION

TABLE OF CONTENTS

Chapter 1

EXECUTIVE SUMMARY

The GUESS (*Greenhouse Use of Energy Seedling Simulator*) model is a lumped-parameter coupled dynamic simulation combining a carbon-based process model of seedling growth with a heat/mass transfer model of the greenhouse envelope. GUESS was created to provide nursery managers and engineers a tool to assess the impact of climate control decisions upon energy use and seedling growth rate. While there is a long history of using mathematical modeling to predict climate and crop production within greenhouses, most models deal with the indoor cultivation of herbaceous plants(vegetables and flowers), and thus are of little applicability to seedling nursery industry. Furthermore, personal communications made with Douglas-fir seedling growers (Don Reagan, Jeff Mehlschau, and Tom Landis) indicates a lack of use of greenhouse-crop models in either research or commercial seedling production. GUESS was written then to fill this modeling gap.

One aspect of GUESS that is unique among energy and crop production models is its dynamic approach towards modeling of climate and plant responses. A mechanistic history of climate and plant growth throughout the entire time course is obtained as opposed to discretely determined steady state values, making GUESS useful for model based control, and any other applications requiring short time scales. The same model, without modification can be used to determine energy consumption and climate controller performance, and short term physiological response.

GUESS requires as inputs an hourly weather data set (temp, rel. H., wind speed and direction, and instantaneous insolation), and parameter file characterizing the plant and greenhouse system. The model provides as outputs: energy costs and usage, indoor temperature and relative humidity, number of crops, and tree size (biomass, height, and diameter). The model was formulated using "industry-standard" mechanistic equations for photosynthesis and other processes, making the model easy to parameterize and applicable to a wide range of woody species.

The source code for GUESS was written using Simulink and MATLAB. Because of its highly graphical and intuitive nature, Simulink was well suited to this project. The high level graphical approach of Simulink allows the modeler to spend the bulk of his or her efforts formulating the problem, as opposed to designing user interfaces and numerical methods. Once a model is completed, the graphical approach simplifies the approach of customization and modification.

While a full scale validation study was not possible, test cases were run to check for reasonableness of the parameters and assumptions used while formulating GUESS. The goal of the test case was to determine the minimum light level necessary for year-round production of Douglas-fir in Willamette Valley region of Oregon. The effect of uncontrolled CO_2 enrichment by recycling flue gas from the heaters was tested as well.

The goal was to have at least 3 full growing cycles. Crop density was kept at 807 plants per m^2. Plants were started at 0.57 g dry weight, and harvested at 1.67 g dry weight.

The model was parameterized for a Douglas-fir crop in $581m^2$ glasshouse located in Corvallis, OR. A detailed list of parameters can be found in Appendix A: List of Parameters.

Preliminary results indicated that for year-round production and three crop cycles, 75 or $100\mu mole/m^2$-s of supplemental lighting was required, depending on whether uncontrolled enrichment was installed or not. This stood in direct contrast to commonly used grower practices discerned from the *Container tree nursery manual* (Landis 1990) and interviews with Donald Reagan, Greenhouse Coordinator: University of Idaho Center for Nursery and Seedling Research; and Jeffrey

Mehlschau, Agricultural Engineer, Weyerhaeuser Western Regeneration. All seemed to suggest that year-round production is possible in the Pacific Northwest using only photoperiodic lighting.

Subsequent modification of the total insolation to PAR (Photosynthetically Active Radiation) conversion factor from 2.1 μmoles/W (original value) to 2.35 μmoles/W (calculated from Langhans, 1990) assuming 50:50 split between PAR: and NIR resulted in 3 growing seasons without supplemental lighting and enrichment, which seemed more reasonable. One can conclude that lack of suitable parameterizing data can be as important in causing model failure of highly non-linear systems, as incorrect formulation.

In addition to the above test, a numerical heat transfer experiment which included the effects of longwave radiation and convection was run to test the assumption of constant cover conductance at indoor operating temperatures of 295K and 298K. For the most part, the assumption proved correct but significant deviations were noted for clear sky, high wind, and low outdoor-indoor temperature differential.

A greenhouse can be divided into the following surfaces: soil, canopy, air space and cover. In GUESS, only the air space is modeled dynamically, the other surfaces are ignored or incorporated into the air space. Using the previously determined resistances, time constants were calculated for the different surfaces (canopy, soil, cover, and air space). If a surface time constant is greater than that of the air space, it should not be ignored, and depending on the level of detail required can be modeled at a constant temperature or dynamically. If a surface time constant is less than that of the air space, then a dynamic balance is not required, although one may be used anyway for numerical simplicity, or to convey additional information to the user. It was found that treating the canopy and cover to be at static conditions equilibrium with respect to the air was found to be acceptable, but the numerical difficulties inherit in iterative methods made implementing as such unattractive for use in Simulink.

Further experimentation is necessary to determine what improvements if any can be made in modeling indoor climate and energy use estimates, by adding these additional complexities.

In its current state, GUESS shows some promise as a modeling tool, however continued improvement is necessary before adapting to commercial use.

Chapter 2

PURPOSE

The purpose of this project was to create a process based model of a tree seedling nursery as an aid to greenhouse operators concerned about energy management. This model termed GUESS, Greenhouse Use of Energy & Seedling Simulator, integrates a lumped parameter heat—mass transfer model of the greenhouse envelope with a process based model of the crop canopy, allowing the user to simultaneously assess the cost of production decisions alongside the impacts upon the health and growth of the crop.

Chapter 3

BACKGROUND AND LITERATURE REVIEW

Artificial regeneration is the practice of replanting logged sites with nursery raised seedlings as opposed to allowing self sowing from superior "seed" trees. Compared to natural regeneration, artificial regeneration allows for shorter downtimes between harvests, better competition with weeds, flexibility in seedling placement, and a greater level of control over diversity of species and genetics (*Container tree nursery manual*, Landis 1990). To produce high quality seedlings year-round, controlled environment systems are often employed (Landis 1990). With this level of control comes a higher production cost, so it is important for greenhouse managers to have models that can be used to assess production decisions. The GUESS model was created to provide nursery managers and engineers with a tool to assess the impact of climate control decisions upon energy use and seedling growth rate.

Horticultural professionals, researchers, ecologists, and engineers have a variety of models to choose from when modeling the growth and development of crop plants. Models can be distinguished based upon the type of crop being modeled, how the model is derived (empirically or mechanistically), the type of climate the model is used in (outdoors vs. indoors), and the type of submodels integrated in the main model.

Most of the common crop models in use are process-based. In process-based models, the rates of growth and development are derived from basic principles in heat and mass transfer and plant physiology. The typical processes for a plant are photosynthesis, respiration, growth and development, and depending on the intended use of the model: flowering and fruiting may be included as well.

In a typical process model, the plant and its environment (if it is in a greenhouse) are treated as control volumes with fluxes of heat and mass entering and leaving through the boundaries (Ch. 3, Bakker et al. 1995). The concentrations of heat and mass determine the system state, which in turn governs the fluxes and various physiological processes occurring within. The physical processes occurring at cellular and leaf level are, for the most part, the same for all plants. So, in theory, a single process model can model any type of plant in any type of environment (Bakker et al. 1995). In reality, due to the complex and non-linear interactions that occur between the plant and its environment, predicting crop performance for wide range of species and environments can be extremely difficult using simple empirical models (Bakker et al. 1995). Furthermore, since each crop has a different economic purpose for cultivation (fruit, flowers, wood, leaves, etc…), at the front end, process models need to be customized for the crop in question.

A wide array of process based models exist for greenhouse crops, a few of the more notable models include TOMGRO for tomatoes and HORTISIM for general purpose crops. For tree crops in outdoor cultivation, models like the Stockle-Riha fast growing tree model can be used.

However, none of the models mentioned before are particularly useful to the manager of a seedling nursery greenhouse. According to Landis (personal communication), although a market is in place for the adoption of crop models for model based control or design and research work, currently none of the available crop models are adequate for seedling nursery use.

While formulating GUESS, three models in particular served as a source of inspiration: Bot's model of indoor climate (Bot 1983), the Stockle-Riha fast growing tree (Riha 2004), and HORTISIM (Gijzen et al., 1998). Equations from Bot's 1983 model on greenhouse climate were used to check several assumptions in the greenhouse heat transfer model, the Stockle-Riha fast growing tree model served as framework for the plant model, and the software techniques described by Jones (1998) and Gijzen et al. (1998) were used when writing the code for GUESS.

In his thesis, Bot conducted an extensive analysis of the different modes of heat transfer in the greenhouse environment. Among several things he observed: a 3 layer model (canopy, glass, and sky) could be used to determine net radiation flux within the greenhouse system, the dominance of forced convection along the outer surface of the glazing, and free convection along the inner surface, and a convective environment within the greenhouse dominated by laminar convection and low, relatively constant (3 cm/s) wind speeds. These findings were used in parts of the energy balance where a constant conductance could not be used (leaves, condensation, and net radiation), and were used in a numerical analysis to check the validity of the constant cover conductance assumption.

The Stockle-Riha fast growing tree model is a model of biomass production for a young eucalyptus plantation. While specifically designed for outdoor use, being process-based, the Stockle-Riha model can be applied to many woody plants. This model served as a basic outline for the seedling model in GUESS. The equations and processes used in the Stockle-Riha were adapted to indoor use: the effects of atmospheric turbulence was ignored, soil moisture was assumed to be relatively constant, photoperiod, seasonal and age related effects were removed from the model as well. The general sequence used in Stockle-Riha and GUESS to calculate growth is as follows:

1. Canopy interception model: determine average irradiance at different leaf levels.
2. Photosynthesis model: use Farquhar equations or another mechanistic model to determine the rate of carbon assimilation at the different leaf levels.
3. Respiration: calculate CO_2 efflux from synthesis of new tissue and maintenance of old tissue.
4. Partition remaining carbon into tissue pools based upon allometry.

According to Jones (1998) one of the major flaws in crop models today is a lack of good software design practices. Unclear organization and a lack of standardized practices have hindered the maintenance and adoption of these models. As a way of working around some of these issues, both Jones and Gijzen et al (1998) have suggested the adoption of state machine/block diagram methodology for writing modeling software. An easy way of implementing these recommendations, and the way in which it was done for GUESS is to forgo traditional text-based computer languages, and write the code using a graphical language like Labview or Simulink, instead. With these kinds of languages, the state machine processes are built in, and the modeler need only focus on the mathematical formulation.

Chapter 4

MODEL DEVELOPMENT

This discussion begins with the basic concepts in heat and mass transfer and then works its way into how these concepts are implemented as equations that describe the transport and physiological processes occurring in the greenhouse. The discussion then finishes with a quick description of how the model was implemented on the computer; the source code is available in appendix B.

Introduction

GUESS is a lumped parameter model, meaning that spatial heterogeneity is ignored and the internal contents and the fluxes across the system boundary are assumed to be uniformly distributed.

The basic construct used to model the various processes taking place is the mass/energy balance. Here the system and its components are treated as a collection of black boxes, functioning as constant volume containers for quantities like heat (temperature), mass (humidity, fixed carbon for a plant, and CO_2), or momentum (wind, pressure, and ventilation). Inside these boxes, only the state variables describing the condition of the system plus any internal source sink term if they exist are of interest. Outside the box, only the fluxes in and out of the immediate boundary are of interest. Conservation equations are used to model the rate of change of system state.

- For a greenhouse these state variables would be temperature, humidity, PAR, and CO_2.
- For the plant the state variables would be water content, organ temperature, dry weight or biomass, and leaf internal CO_2 level.

A complete equation for the transport of some scalar quantity across a control volume is as follows:

$$CV\frac{\partial\Phi}{\partial t} = \int_{CS} \tfrac{\partial\Phi}{\partial x} * \left(\underbrace{\mathbf{u} \bullet \mathbf{n}}_{ADVECTIVE} + \underbrace{kdx}_{CONDUCTIVE} \right) + \int_{CV} Q_{\text{net}} \tag{4.1}$$

C is capacitance. V is system volume. Phi is some potential energy function driving the transport. U is the velocity vector, n is the unit outward normal vector. k is the boundary conductivity, and dx is the boundary thickness CS and CV refer to control volume and control surface, respectively. Equation (4.1) is a more complete representation of the process described in (4.2)

Using mass/energy balances with the lumped parameter assumption allows us to reduce the system of equations from second order partial differential equations in both space and time to a system of first order, albeit non-linear ordinary differential equations in time. :

$$C * V * \tfrac{\partial\phi}{\partial t} = A\left(F_{\text{in}} - F_{\text{out}}\right) + V(Q_{\text{source}} - Q_{\text{sink}}) \tag{4.2}$$

C is capacitance. Phi is some quantity describing the system state. V is system volume(control volume). A is area of flux boundary(control surface). F is the external flux terms. Q is the internal creation/destruction terms.

For momentum, the potential function is velocity. For heat, the potential function is temperature. For mass, the potential function is concentration. With heat and mass, equation (4.1) becomes a linear scalar equation. For velocity, the non-linear Navier-Stokes equation must be used. In GUESS, only mass and energy balances are considered, and the added complexity of a momentum balance is ignored.

Heat transfer

Heat is transferred from an object by conduction across a boundary, by advection of a fluid moving through the object, by convection of a moving fluid at the surface or radiation via electromagnetic waves.

Advection

Advection is the transport of heat and matter by a bulk fluid moving through a control volume. This flux of fluid through a boundary results in a transfer of heat proportional to the product of the fluid velocity and temperature gradient.

$$Q_{1\to 2} = \mathbf{u} \bullet \mathbf{n} C_P A(T_1 - T_2) \tag{4.3}$$

Heat Flow by advection, u · n * A is the flux normal to the boundary. C_P is the volumetric heat capacity

Conduction

Heat can be transported across a boundary by molecular diffusion or conduction. This process obeys Fourier's Law, and using the lumped parameter assumption, can be modeled using Ohm's Law.

$$Q_{1\to 2} = UA(T_1 - T_2) \tag{4.4}$$

Heat Flow by conduction

Very often, in convection studies for example, heat conductances are expressed in velocity type units, so some means of conversion to U-value is required. The following equation can be used to convert between the two types of conductances.

$$U_{HEAT} \quad \left(\mathrm{W \cdot K^{-1} \cdot m^{2}}\right) \quad = \quad \rho C_P g_{HEAT} \quad \left(\mathrm{m \cdot s^{-1}}\right) \tag{4.5}$$

Conversion between U-value and velocity type heat conductances. ϱ & C_P are the density and specific heat of the conducting medium. g_{HEAT} is the thermal conductance in velocity type units.

Convection

Whenever a moving fluid encounters a solid boundary, because of the no-slip condition a boundary layer of slow moving fluid must develop at the interface. To enter the free stream, heat, mass, and momentum must diffuse across this boundary layer. The resistance of the boundary to the diffusion of heat is modeled using the Nusselt number:

$$Nu = \frac{g_{HEAT}}{\kappa L} \tag{4.6}$$

Definition of Nusselt #, g_{HEAT} is conductivity to heat, k is the thermal diffusivity and L is the total length of the object(diameter or width) parallel to the direction of the convection.

There are two forms of convection: forced convection, driven by inertial forces and dependent upon the Reynolds number; and natural convection, driven by buoyancy and dependent upon the Grashof number.

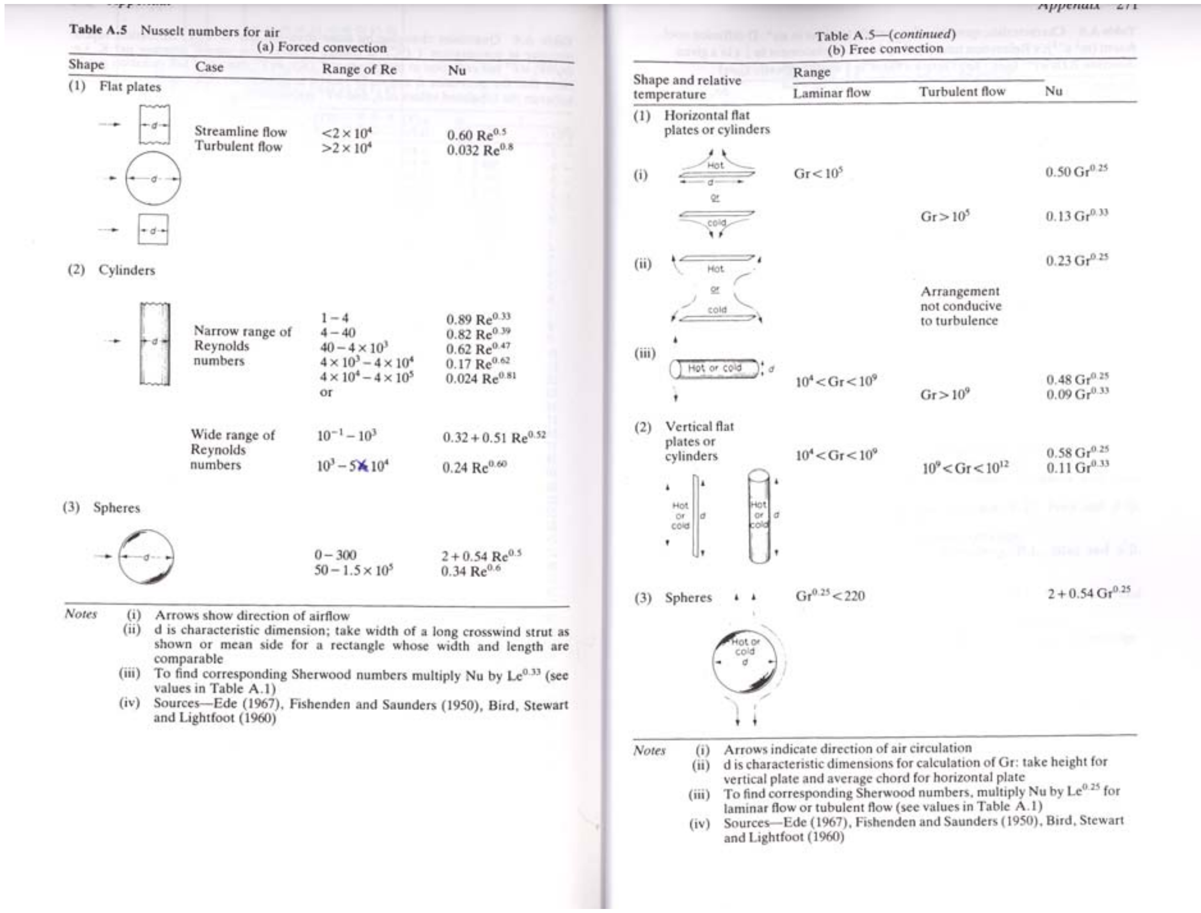
$$Re = \frac{u\ell}{\nu} \tag{4.7}$$

Reynolds number definition: where u is velocity, l is the length scale, and ν is kinematic viscosity

$$
\begin{aligned}
&Gr = \frac{\beta g \ell^3 \Delta T}{\nu^2} \\
&\text{where} \\
&\beta = \frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right) \\
&\text{for an ideal gas:} \\
&\beta = \frac{1}{T} \\
&Gr = \frac{\beta g \ell^3 \left|T_s - T_a\right|}{\nu^2 T_a}
\end{aligned}
\tag{4.8}
$$

Grashof number definition where β is thermal expansivity and ν is kinematic viscosity. g is gravitational acceleration, l is length scale, and T_a and T_s are the temperatures of the fluid and the surface.

Equations for forced and natural convection are given on the next page. The end result is that the resistances for convection are not constant material properties like conduction but depend non-linearly upon temperature gradient, as in natural convection ($T^{1/4}$ or $T^{1/3}$) or velocity ($u^{0.5}$ or $u^{0.8}$) as in forced convection. That being said, Ohm's Law still applies.



Appendix 271

Table A.5 Nusselt numbers for air

(a) Forced convection

Shape	Case	Range of Re	Nu
(1) Flat plates	Streamline flow	$<2\times10^4$	$0.60\ Re^{0.5}$
	Turbulent flow	$>2\times10^4$	$0.032\ Re^{0.8}$
(2) Cylinders	Narrow range of Reynolds numbers	$1-4$	$0.89\ Re^{0.33}$
		$4-40$	$0.82\ Re^{0.39}$
		$40-4\times10^3$	$0.62\ Re^{0.47}$
		$4\times10^3-4\times10^4$	$0.17\ Re^{0.62}$
		$4\times10^4-4\times10^5$	$0.024\ Re^{0.81}$
		or	
	Wide range of Reynolds numbers	$10^{-1}-10^3$	$0.32+0.51\ Re^{0.52}$
		$10^3-5\times10^4$	$0.24\ Re^{0.60}$
(3) Spheres		$0-300$	$2+0.54\ Re^{0.5}$
		$50-1.5\times10^5$	$0.34\ Re^{0.6}$

Notes
(i) Arrows show direction of airflow
(ii) d is characteristic dimension; take width of a long crosswind strut as shown or mean side for a rectangle whose width and length are comparable
(iii) To find corresponding Sherwood numbers multiply Nu by $Le^{0.33}$ (see values in Table A.1)
(iv) Sources—Ede (1967), Fishenden and Saunders (1950), Bird, Stewart and Lightfoot (1960)

Table A.5—(*continued*)

(b) Free convection

Shape and relative temperature	Range: Laminar flow	Range: Turbulent flow	Nu
(1) Horizontal flat plates or cylinders			
(i) Hot or cold	$Gr<10^5$		$0.50\ Gr^{0.25}$
		$Gr>10^5$	$0.13\ Gr^{0.33}$
(ii) Hot or cold		Arrangement not conducive to turbulence	$0.23\ Gr^{0.25}$
(iii) Hot or cold	$10^4<Gr<10^9$		$0.48\ Gr^{0.25}$
		$Gr>10^9$	$0.09\ Gr^{0.33}$
(2) Vertical flat plates or cylinders (Hot or cold)	$10^4<Gr<10^9$		$0.58\ Gr^{0.25}$
		$10^9<Gr<10^{12}$	$0.11\ Gr^{0.33}$
(3) Spheres (Hot or cold)	$Gr^{0.25}<220$		$2+0.54\ Gr^{0.25}$

Notes
(i) Arrows indicate direction of air circulation
(ii) d is characteristic dimensions for calculation of Gr: take height for vertical plate and average chord for horizontal plate
(iii) To find corresponding Sherwood numbers, multiply Nu by $Le^{0.25}$ for laminar flow or tubulent flow (see values in Table A.1)
(iv) Sources—Ede (1967), Fishenden and Saunders (1950), Bird, Stewart and Lightfoot (1960)

Figure 1 Table of Nusselt numbers for various convective modes and geometries, taken from Monteith and Unsworth, 1990 pgs 270-271

Radiation

Radiation exchange follows Stefan's Law, where the radiation admitted by a surface is proportional to the temperature in Kelvin to the fourth power. Because of this, radiation exchange between two surfaces is non-Ohmic, and must be expressed differently than the other two processes.

$$\begin{aligned} Q_{1\to 2} &= F_{1\to 2}\varepsilon_1\sigma\left(T_1^4\right) \\ Q_{2\to 1} &= F_{2\to 1}\varepsilon_2\sigma\left(T_2^4\right) \\ Q_{net} &= F_{avg}\varepsilon_{avg}\sigma\left(T_1^4 - T_2^4\right) \end{aligned} \tag{4.9}$$

Radiation flux between two diffuse gray surfaces. Sigma is the Stefan Boltzmann constant. Epsilon are emissivities, and F are view factors.

View Factor

The view factor of object B from object A is the fraction of total flux absorbed by object A that was emitted by object B. The view factor from one gray surface to another is equal to the ratio of projected surface area of the transmitter upon the receiver over the total surface area of the receiver. When only two objects are in consideration, the ratio of view factors is simply the ratio of projected surface areas.

$$\frac{F_{1\to 2}}{F_{2\to 1}} = \frac{A_2}{A_1} \tag{4.10}$$

View Factor

The determination of view factors often requires extensive numerical computation. However for some simple geometries, analytical solutions do exist. For a point source enclosed by an infinite hemisphere, like a greenhouse and the sky, F is simply 1 for the sky when viewed from the greenhouse, and 0 when for the greenhouse when viewed from the sky. For a flat plate suspended horizontally between the earth and the sky, like a roof, for example, the view factors for the ground and the sky are both 1 or 0.5, depending on whether the total(double-sided) or projected(single-sided) surface area is used.

For other geometries, analytical solutions can be found in handbooks (see Monteith & Unsworth 1990, Albright 1990, or ASHRAE 2001).

Emissivity

Emissivity is the ratio between the flux from an ordinary object and an idealized blackbody.

$$\varepsilon = \frac{Q_{actual}}{Q_{blackbody}} \qquad Q_{blackbody} = \sigma T^4 \tag{4.11}$$

Expression for emissivity

For two or more surfaces exchanging radiation, total or average emissivity can be expressed as follows:

$$1 + \frac{1}{\varepsilon} = \frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} \tag{4.12}$$

Effective Emissivity

Radiation as a conductance

For small temperature differences, like those encountered in most environmental systems, radiative transfer can be approximated as a linear conductive process. We divide eq. (4.9) by the temperature difference, take the binomial expansion, and remove the higher order terms.

$$h_{r:1\leftrightarrow 2} \approx 4F\varepsilon\sigma T_{base}^{\;3}$$
$$T_{base} = \frac{T_1 + T_2}{2} \tag{4.13}$$

Effective conductance due to radiation. Albright 1990

If the average temperature between the two surfaces in question is used for T_{BASE}, eq. (4.13) is accurate to within ±5% for common environmental conditions (Campbell, 1998). This conductive process is Ohmic, since Q is now an explicit linear function of temperature difference of the type: $\Delta T/R$; and for a given base temperature, R is constant. Radiation resistances can be added in series or parallel like conductive or convective resistances, and fast lookup tables can be used to determine h_r instead of solving for the individual fluxes directly.

Direct vs. Diffuse Radiation

Radiation can be divided into two pools: diffuse and direct. Direct radiation is unidirectional, and displays shadowing. On the other hand, diffuse radiation is randomly distributed, and does not display shadowing, and shows little directional preference.

Because of its directional nature, absorbance of direct radiation shows a strong angular dependence. Solar radiation, and artificial lighting are assumed to produce a mixture diffuse and direct radiation. Thermal radiation from all surfaces is assumed to be diffuse.

The flux of direct radiation absorbed by a black surface is calculated using Lambert's Cosine Law (Monteith and Unsworth 1990):

$$Q = I\alpha\left(\phi\right)\cos\phi \tag{4.14}$$

Lambert's Cosine Law. Direct radiation flux absorbed by an object, I is total direct flux, α is absorptivity, φ is angle of incidence.

For diffuse surfaces like biological materials, transmissivity, reflectivity and absorptivity are independent of the incidence angle. For specular surfaces like metals, clear liquids, and glass; these terms can show strong angular dependence at high angles of incidence. However, when a specular surface is illuminated by isotropic (all angles of incidence are equally represented) diffuse radiation , radiation properties are angularly independent.

$$\begin{aligned} \tau + \rho + \alpha &= 1 \\ \varepsilon &= \alpha \end{aligned} \tag{4.15}$$

Kirchhoff's Law: Relation between transmissivity, reflectivity, absorptivity and emissivity

For many common specular materials, transmissivity and reflectivity can be calculated using Fresnel's Law, as in Bot (1983). However a Fourier series or trigonometric power law is sufficiently accurate enough for most cases. The following formula was fitted to glass transmissivity data found on section 30 of the ASHRAE Handbook (ASHRAE 2001).

$$\tau\left(\alpha\right) = \tau_{\perp}\left(\cos\alpha\right)^{1/n} \tag{4.16}$$

Glazing transmissivity equation, where α is the deviation angle from normal. $\tau\perp$ is normal transmissivity, and n is a material exponent, approx. 2 for glass.

Because of the differences in transmission, it is important to distinguish between diffuse and direct solar radiation. In GUESS, all solar diffuse radiation is assumed to be isotropic, and independent of solar altitude.

It is best when diffuse and direct radiation fluxes are measured separately, however like cloud cover, quantum content, and incoming longwave flux; separate diffuse and direct measurements are rarely available. Most often, only global solar radiation is provided (diffuse + direct). One must make use then of correlations to separate the global radiation into the two pools.

The amount of diffuse radiation available is a function of the amount of haze and clouds in the sky. Since that type of data is rarely available, one uses a proxy variable, the clearness index (k_t) which is the ratio between global solar radiation on a horizontal surface at ground level to extraterrestrial solar radiation.

$$k_t = \frac{Solar_{GLOBAL,GROUND}}{Solar_{ET}} \tag{4.17}$$

Clearness index

The term $\mathrm{Solar}_{\mathrm{ET}}$ in the clearness index formula refers to extraterrestrial radiation upon a flat plate, meaning the radiation flux the plate would experience were there no atmosphere. $\mathrm{Solar}_{\mathrm{ET}}$ in GUESS is calculated using the standard angle formulas outlined in Chapter 4 of Monteith & Unsworth, 1990, assuming a solar constant of 1360 $\mathrm{W{\cdot}m^{-2}}$. Using the clearness index, the fraction of diffuse radiation is determined using the formula from Erbs et al. 1982

$$f_{diffuse} = 0.9511 - 0.1604k_t + 4.388k_t^{\,2} - 16.638k_t^{\,3} + 12.3364k_t^{\,4} \tag{4.18}$$

Diffuse fraction of incident solar radiation

Beer's Law, Radiation in a continuum

In continuum materials like liquids or gases: emission, reflection, scattering (multiple reflections) and absorption are assumed to be first order processes, and thus follow Beer's Law.

$$\frac{I}{I_0} = e^{-Kl} \tag{4.19}$$

Beers Law for radiation transmission. K is the attenuation coefficient, l is the path length. Replacing I_0 with ideal blackbody emission can give the effective emissivity for a gray body of thickness, L at temperature T.

For diffuse radiation, one applies Beer's law to determine transmissivity for a given altitude and azimuth angle in the view field, and the result is integrated throughout the entire field of view to get the total diffuse transmitted radiation.

$$\tau_{diff} = \int_0^{2\pi} \int_0^{\pi/2} \tau_{beam}(\theta) \sin\theta \cos\theta \, d\gamma \tag{4.20}$$

Diffuse transmissivity, θ is altitude and γ is azimuth

In the strictest sense, Beer's Law only applies to uniform continuum materials, but with some clever statistical manipulation, it can be used to determine direct beam attenuation in heterogeneous materials like crop canopies (Campbell, 1998). If a canopy of ground area A is composed of N opaque, randomly arranged leaves all with an area of K**a* projected in the direction of the beam then the probability of light transmission is simply

$$\tau = (1 - Ka / A)^N \tag{4.21}$$

As the size of the leaves get smaller, we can approximate transmission with a Poisson process and we get the following Beer's Law analogy:

$$\tau = \mathrm{e}^{-LAI_{proj}} \tag{4.22}$$

For non-opaque leaves of absorptivity, α, one uses:

$$\tau = \mathrm{e}^{-\sqrt{\alpha} LAI_{proj}} \tag{4.23}$$

The square root term comes from the Kubelka-Munk theory for infinite reflections, see (Monteith and Unsworth, 1990)

The formulas above were initially derived for horizontally arranged leaves, but can easily be extended to three dimensional leaf arrangements by using $K(\psi)$, the attenuation coefficient, which is the fraction of leaf area projected to the beam where ψ is the zenith angle.

$$\tau = \mathrm{e}^{-\sqrt{\alpha} LAI * K(\psi)} \tag{4.24}$$

Beer's Law-type formula for canopy transmission

For direct radiation with a spherically distributed leaf angle arrangement (most canopies), the attenuation coefficient is given as:

$$K(\psi) = \frac{1}{2\cos\psi} \tag{4.25}$$

Spherical canopy attenuation coefficient

For non-spherical leaf arrangements, Campbell presents the following formula:

$$K(\psi) = \frac{\sqrt{x^2 + \tan^2\psi}}{x + 1.774(x + 1.182)^{-.733}} \tag{4.26}$$

Elliptical canopy attenuation coefficient. x is the elliptical parameter: 0 for vertical arrangement, 1 for spherical, and infinity for horizontal.

Canopy transmissivity to diffuse radiation is calculated using eq. (4.26), (4.20) and numerical quadrature. During the course of simulation, numerical lookup tables are used and eq. (4.24) to (4.26) are never actually solved directly.

Sky Temperature

A key loss term in the heat balance for any building is longwave radiation loss to the sky. But much like diffuse radiation, net longwave emissions are rarely available in weather data sets; instead, they are modeled from air temperature, cloud cover, and humidity.

While Beer's Law will work for calculating atmospheric longwave emissions, eq. (4.11), the Stefan-Boltzmann Law is far easy to implement. So use this law, one must create an artificial surface called the sky which as a blackbody emits the same level of radiation that is emitted by the atmosphere. Several numerical correlations have been derived in the past to obtain a clear sky emissivity given a set of surface conditions. One of the more popular correlations and the one implemented in GUESS is the one developed by Brutsaert (1975), which uses vapor pressure in kPa and air temperature in K.

$$\varepsilon_{sky} = 1.72\left(\frac{VP_{air}}{T_{air}}\right)^{1/7} \tag{4.27}$$

Brutsaert's correlation for clear sky emissivity

The clear sky emissivity thus produced is then corrected for cloud cover to obtain a net sky emissivity. But, since cloud cover data is not always available, a correlation involving clearness index (Sugita & Brutsaert 1993) is used:

$$\varepsilon_{sky} = 1.02\varepsilon_{sky,clear}k_t^{-0.0227} \tag{4.28}$$

Finally, an equivalent blackbody sky temperature can be calculated:

$$T_{sky} = \varepsilon_{sky}{}^{1/4}T_{air} \tag{4.29}$$

Radiation Partitioning

Using Wien's Law, the radiation environment in the greenhouse can be subdivided into two essentially non-overlapping pools: shortwave (color temperature: 6500K) and longwave (color temperature: 273K). Furthermore, the shortwave pool can be divided again into a PAR (Photosynthetically Active Radiation) band [400-700nm] which is available for photosynthesis, and a NIR (near infrared) band [700-5000nm] which is absorbed as sensible heat. For standard sunlight, it can be assumed that 50% of the total energy is partitioned into each band: PAR and NIR.

Because of the quantum nature of photosynthesis, PAR flux is typically measured in molar units, μEinsteins [μmoles/$m^2 \cdot s^{-1}$]. For the purpose of calculating an energy budget, a table is provided indicating the molar fluxes of PAR for a given rate of power consumption:

Light Source (excludes ballast)	μEinstein per Wm^{-2} of total power output
Sunlight	2.35
HPS	1.51
metal halide	1.40
incandescent	0.4
Cool White	0.97

Table 1 Various lighting sources and efficiencies in the PAR band. Data given is for 1000W HID lamps, for 500W HID lamps, multiply by 0.84. Sources: 1. molar efficiency in 400-700nm; "Plant Growth Chamber Handbook", Env. Growth Chambers, Inc. 1997. 2. radiant efficiency (W visible illumination/ W of total power), 500W luminairies (Bakker et al. 1995). 1000W luminairies (Aldrich and Bartok, 1994).

Mass transfer

Mass transfer operates in a similar manner to heat transfer, and can be described using the same processes. Radiation can be ignored, and Lewis #'s can be used to convert resistances from heat form to mass form. Mass balances must be performed for each substance of interest. For every chemical process that converts one substance to another, and for every physical process that converts one phase of the same substance to another, reaction source/sink terms must be specified. Since many of these processes result in the release or uptake of heat, and the rates of reaction are highly temperature dependent, these heat transfer terms must be included as well in coupled heat and mass transfer.

Convection & Conduction

Mass conducts through a porous medium or resting fluid by diffusion. Diffusion is an Ohmic process, where the rate of transfer is linearly proportional to the potential gradient. The conductance for diffusion in velocity form is the ratio between the diffusivity and the length scale.

$$g = \frac{\mathcal{D}}{\ell} \tag{4.30}$$

Conductance to diffusion

Since the potential for mass transfer is concentration, a means of conversion between velocity type and molar type conductances is needed.

To convert between the two, the function given on H. Jones, 1991 is used.

$$g_{molar} \quad \left(\mathrm{mol \cdot m^{-2} \cdot s^{-1}}\right) \quad = \quad g\frac{P}{RT} \quad \left(\mathrm{m \cdot s^{-1}}\right) \tag{4.31}$$

Conversion between molar and velocity type mass conductances

$$g_{molar} \quad \left(\mathrm{mmol \cdot m^{-2} \cdot s^{-1}}\right) \quad = \quad 40.3g \quad \left(\mathrm{m \cdot s^{-1}}\right) \quad @T = 25°C, \quad P = 101\,\mathrm{kPa} \tag{4.32}$$

Conversion between molar and velocity type mass conductances at standard conditions

Correlations for convection are often expressed in velocity type form with respect to heat. To convert between a velocity type heat transfer conductance and velocity type mass transfer conductance, the Lewis number is employed.

$$Le = \frac{\mathcal{D}}{k} \tag{4.33}$$

Below is a table of the various Lewis # relations employed in GUESS:

$$\frac{g_{mass}}{g_{heat}} = Le^{1} \qquad (\text{molecular diffusion})$$

$$\frac{g_{mass}}{g_{heat}} = Le^{3/4} \qquad (\text{natural convection})$$

$$\frac{g_{mass}}{g_{heat}} = Le^{2/3} \qquad (\text{forced convection})$$

$$\frac{g_{mass}}{g_{heat}} = Le^{2/3} \qquad (\text{turb. nat convection})$$

$$\frac{g_{mass}}{g_{heat}} = Le^{0} \qquad (\text{free stream turbulence})$$

Table of Lewis number relations

Psychrometrics

There are three ways of expressing the vapor content of air: humidity ratio (ratio of water vapor mass to mass of dry air) which is useful for mixing problems; vapor pressure deficit (difference between saturation and current vapor pressure) which is useful for potential driven flow problems like evaporation or condensation; and relative humidity which is the ratio between current vapor pressure and saturation vapor pressure which is used to characterize climate, and derive other measures of vapor content.

$$rh = \frac{VP}{VP_{sat}} \tag{4.34}$$

Definition of relative humidity: VP and VP_{sat} refer to vapor pressure at current and saturation conditions respectively

$$\begin{aligned} \rho_{VAPOR} &= \frac{VP}{R_{H_2O}T} \\ \rho_{AIR} &= \frac{P_{atm}}{R_{air}T} \\ H = e = \frac{\rho_{VAPOR}}{\rho_{AIR}} &= \frac{VP}{P_{atm}}\frac{R_{air}}{R_{H_2O}} = \frac{29}{18}\frac{VP}{P_{atm}} \end{aligned} \tag{4.35}$$

Definition and derivation of humidity ratio(H)

The vapor pressure at saturation for a liquid can be determined using the Clasius-Clapeyron relation. However, for water vapor in the range: -20-100C, more accurate correlations exist like the one given in the ASHRAE Handbook on p 6.2, which is the one used in GUESS. Due to the exponential non-linearities of the saturation vapor pressure equation, they are not solved directly during the simulation, instead a lookup table is prepared, and linear interpolation is used to find the intermediate values.

$$VP_{sat}\left(T\right)=\tfrac{1}{1000}\exp\begin{Bmatrix}-5800/T+1.391-48.64T+4.176\text{E-}5T^{2}-\\ 1.445\text{E-}8T^{3}+6.546\ln T\end{Bmatrix} \tag{4.36}$$

Saturation vapor pressure over liquid water from Hyland and Wexler (1983) presented in ASHRAE 2001.

The enthalpy of a mass of moist air is the sum of its latent and sensible components:

$$h=\lambda e+C_{P}T \tag{4.37}$$

Enthalpy of a moist air mass, C_P = constant pressure specific heat, λ = latent heat of evaporation, e = humidity ratio kg H_2O/kg moist air. Assumption: air mass much greater than vapor mass.

For an adiabatic process, the temperature drop is proportional to the increase in humidity ratio. The temperature will continue to drop as long as the vapor partial pressure is below saturation. The lowest possible temperature achievable is called the wet bulb temperature, and it is a unique property of the air temperature and humidity. The wet bulb temperature can be determined using the psychrometric equation.

$$rh_{air}*VP_{sat}\left(T_{air}\right)-VP_{sat}\left(T_{wb}\right)=\gamma P_{atm}(T-T_{wb}) \tag{4.38}$$

The Psychrometric Equation, used to determine wet bulb temperature a given relative humidity and temperature. γ is the psychrometric constant: C_p/λ. P_{atm} is atmospheric pressure in pascals.

The value of the psychrometric constant varies slightly with temperature; the value used in the GUESS model is 6.64E-4 K^{-1}.

Penman-Monteith Equation

Like saturation vapor pressure, lookup methods instead of direct evaluation are used to get wet bulb temperatures and equivalent absolute humidities. For a diabatic process, the rate of heat addition is proportional to the change in humidity ratio. For a wet surface being irradiated by a constant radiation source, we assume that the surface is at saturation, and that the sum of sensible and latent heat exchange is equal to net radiation.

$$\lambda E + C = R_{net}$$

The sensible heat and latent heat fluxes leaving the surface are

$$C = \rho C_P \frac{(T_0 - T_{air})}{r_H}$$

$$\lambda E = \rho C_P \frac{VP_{sat}(T_0) - VP_{air}}{\gamma r_H}$$

Where T_{air} is air temperature and T_0 is surface temperature, which is unknown.

Upon linearizing the latent heat term:

$$\frac{VP_{sat}(T_{air}) + \left.\frac{dVP_{sat}}{dT}\right|_{T=T} * (T_0 - T_{air}) - VP_{air}}{\gamma r_H}$$

And solving for the temperature difference:

$$T_{air} - T_o = \frac{r_H}{\rho C_p}\left(R_{net} - \lambda E\right)$$

And adding an additional resistance to vapor transfer to account for diffusion across the stomatal boundary, the Penman-Monteith equation for crop transpiration is obtained:

$$\frac{\Delta R_{NET}}{\Delta+\gamma^{*}}+\frac{\rho C_P\left\{VP_{sat}-VP\right\}}{\Delta+\gamma^{*}}$$
$$\gamma^{*}=\gamma*\left[1+\frac{r_{c,VAPOR}}{r_{a,HEAT}}\right] \tag{4.39}$$

Penman-Monteith Equation. Δ is the slope of the saturation vapor curve. γ is the psychrometric constant multiplied by atmospheric pressure. The resistances terms are r_a, the aero-dynamic resistance to heat transfer, and r_c the canopy resistance to vapor transfer. R_{net} is the net radiation falling upon the surface. To determine the R_{net} term we could perform a radiation balance:

$$R_{net}=\alpha I-F_{cover}\varepsilon_{cover}\left(T_o^{\,4}-T_{cover}^{\,4}\right)-F_{sky}\left(1-\varepsilon_{cover}\right)\left(T_o^{\,4}-T_{sky}^{\,4}\right) \tag{4.40}$$

Net Isothermal Radiation

The disadvantage of using a radiation balance to solve for R_{NET} is that it requires knowledge of surface temperature a priori, and that an iterative search procedure is required to determine surface temperature. The main motivation for using the Penman-Monteith equation is that it provides an explicit expression for vapor flux. So instead, we assume R_{NET} is net isothermal radiation which is the radiation absorbed by a surface if it were at air temperature. The difference in radiation emitted between air temperature and surface temperature is usually captured by inserting the radiative resistance $h_r(T_{air})$ into r_a. For canopy surfaces, h_r is sometimes ignored, since this can lead to better estimates of evapotranspiration, see Allen et al. 1994.

Equilibrium Evapotranspiration

The Penman-Monteith equation can be used to estimate evaporation from any wet surface such as a lake or a canopy. Additional sources of heat (static and dynamic) can be added to the net radiation term. If the ratio of boundary layer resistance to stomatal resistance is high enough, as in a greenhouse, we can assume the system to be de-coupled (meaning that the air and water surface are in equilibrium, and flow is driven by radiation as opposed to vapor pressure deficit. The Penman-Monteith equation reduces to the following form:

$$\lambda E = \frac{\Delta R_{net}}{\Delta + \gamma^*} \tag{4.41}$$

This observation has been confirmed using the GUESS model, and independently by researchers (Bakker et al. 1995 and H. Jones, 1991). In full sun with T = 25C, about 67% of incoming net radiation is converted to latent heat. In GUESS, only equilibrium ET is modeled.

Transfer Processes in the Greenhouse Context

Heat Transfer Processes

Heat transfer in the greenhouse involves all main modes: Conduction, Convection (natural and forced), and Radiation. Sources of heat include: shortwave (solar) radiation, lighting, and condensation. Sinks include conduction, longwave radiation, infiltration, ventilation, and evaporation/transpiration.

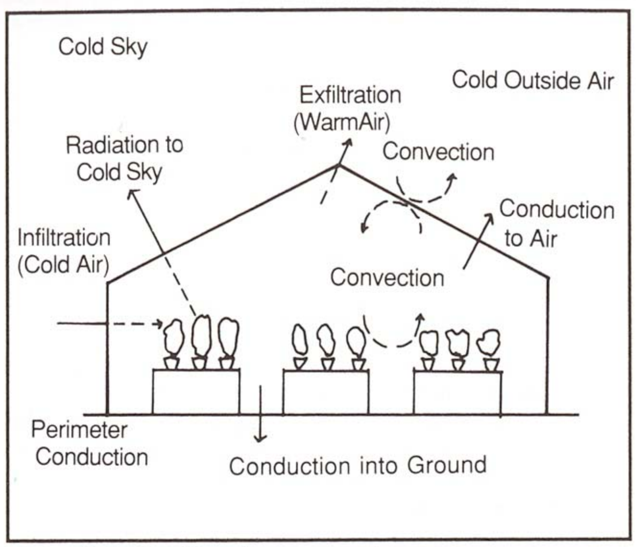


Figure 4–2. Heat loss from a greenhouse.

Figure 2 Heat Transfer: Conduction and Ventilation within the greenhouse, source: Aldrich and Bartok 1994

Shortwave Radiation

Solar radiation enters the greenhouse through the cover. A portion of the radiation is reflected by the cover or absorbed by the framing. The remainder reaches the greenhouse surfaces (floor, benches, and canopies) where it is converted to heat (sensible or latent), which reaches the air mass via convection. Solar radiation is referred to as shortwave gain in the greenhouse literature.

Although the Fresnel equations in conjunction with the equation for incident radiation on a tilted surface could be used to formulate a model for cover transmissivity (see Bot 1983 or Wang and Boulard, 2000), the complexity of such an approach makes parameterization difficult. Instead, a Fourier series type regression is used. The following formula is used by GUESS to calculate cover transmissivity:

$$\begin{aligned} \tau_b(\alpha) &= \tau_1 + \tau_2 \sin(\alpha)^{1+n} + \tau_3 \sin\left(\frac{2\beta}{\pi}\alpha\right) \\ \tau_{TOTAL} &= (1 - f_{diff})\tau_b + f_{diff}\tau_d \end{aligned} \tag{4.42}$$

τ_{TOTAL} is the total glazing transmissivity: the weighted average of τ_B (direct) and τ_D(diffuse). Alpha is the solar altitude angle. τ_1 refers to sunset/sunrise direct transmissivity, τ_2 refers to change in transmissivity with solar altitude, τ_3 corrects for roof tilt(optional). For GUESS, the default cover material is single pane glass, and the $\tau_{1\rightarrow 3}$ values are 0.5, 0.3 and 0.07. N is the material exponent (approx ½), see eq(4.16).

Material	Thickness (mm)	IR-trans-mittance	Light transmittance	
			diffuse	direct
Single glass	4	0	0.83	0.89
Double glass, Cavity 9 mm	2×3	0	0.72	0.82
PMMA acrylic Double-web Sheet (Röhm)	16	d.n.a. [1]	0.75	0.84
Polycarbonate Double-web Sheet (Qualex)	10	d.n.a.	0.65	0.74
PE, UV-stabilized	0.20	0.56	d.n.a.	0.89–0.92
EVA (ethylenevinyl-acetate)	0.18	0.22	d.n.a.	0.91
Teflon FEP	0.05	0.57	d.n.a.	0.96

[1] *d.n.a. = data not available.*

Figure 3 Average Glazing transmissivities from Bakker et al. 1995

Longwave Radiation

Aside from solar radiation, the other mode of radiative heat transfer is blackbody thermal emissions. All greenhouse surfaces while differing in shortwave albedo and transmissivity emit thermal radiation at or near blackbody levels, with the possible exceptions being a polyethylene cover and any unpainted metal conduits. Thermal radiation is emitted by the surfaces reaching the cover can either be transmitted to the sky or absorbed and re-radiated.

Emissivities differ among over materials; ranging from as low as 0.2 for polyethylene film to as high as 0.99-0.95 for high-iron glass (Aldrich and Bartok 1994), and are a major determining factor in the effective conductance of the greenhouse envelope. If we assume that the internal surfaces of the greenhouse are at the same temperature as the air, the longwave radiative fluxes can be expressed simply as:

$$\begin{aligned} Q_{lw:\ cover} &= F_{cover}\varepsilon_{cover}\sigma(T_{air}{}^{4} - T_{cover}{}^{4}) \\ Q_{lw:\ sky} &= F_{sky}(1-\varepsilon_{cover})\sigma(T_{air}{}^{4} - T_{sky}{}^{4}) \end{aligned} \tag{4.43}$$

Longwave Fluxes in Stephan-Boltzmann form

The view factor, F_{sky}, between the greenhouse (at T_{air}) and the sky is 1, which is to be expected for an object complete enclosed by a "black" hemisphere. The view factor between the ground (canopy and soil), and the cover, F_{cover} is also one. The view factor between the cover and the canopy however is usually less than one, typically 0.8 using one-sided cover area, (personal communication Albright), because a portion of the cover sees itself. The view factors for the various greenhouse surfaces can be estimated more precisely using formulas found in Takakura (1989). Rewritten using "full-view" blackbody radiation resistances, the net longwave flux leaving the ground (assumed to be isothermal with the air: is expressed below):

$$-Q_{LW} = (1-\varepsilon_{cover})h_{r:\,sky}(T_{air} - T_{sky}) + \varepsilon_{cover} h_{r:\,cover}(T_{air} - T_{glass}) \qquad (4.44)$$

Net longwave flux in resistor form

Conduction

Conduction is the primary means of heat exchange between the greenhouse and the outside. Conduction occurs between the floor and the soil and between the greenhouse air space and the outside atmosphere.

$$Q_{cond} = UA_{cover}(T_{in} - T_{out}) + UA_{floor}(T_{in} - T_{out}) \tag{4.45}$$

Total conductive heat flow in watts. T_{in} and T_{out} refer to indoor and outdoor air temperatures, respectively.

The conduction resistance(1/U) of the cover or its R-Value, can be found in many handbooks and textbooks on greenhouse construction, oftentimes radiative and convective components are included in the calculation for a total apparent conductance. For simple energy-use calculations, a single resistance can be used, although more accurate models may require consideration of indoor and outdoor climate. In GUESS, a single lumped resistance is used for the cover.

Table 4.3.1 – U-values of a square greenhouse of 0,5 ha at a wind velocity of 4 m s^{-1}.

Covering material	U-value (W m^{-2} K^{-1})
Single glass	8.8
Double glass in sidewalls	7.9
All double glass	5.2
Double acrylic	5.0
Double polycarbonate	4.8
Single PE film	8.0
Double PE film	6.0

GREENHOUSE CLIMATE CONTROL 171

Figure 4 Suggested U-Values for different covering materials, includes infiltration losses. Source: Bakker et al 1995.

The floor term deserves extra attention because it includes both the static and dynamic conduction of heat through the floor and surrounding soil. Several models can be used, but the simplest, and the one used in GUESS is the perimeter loss model.

$$Q_{perimeter} = P_{floor} \left(U\ell\right)_{perimeter} (T_{in} - T_{out}) \tag{4.46}$$

Perimeter Loss Model

The perimeter loss coefficient is a function of foundation depth, thermal capacitances of the soil and floor. Detailed calculation procedures can be found in the ASHRAE Handbook (ASHRAE 2001).

The conduction model, due to its ability to express transfer processes using Ohm's Law, is used to represent all heat and material transfers within GUESS: linear and non-linear. Linear resistances are assumed to be constant values, and non-linear resistances are simple functions of temperature.

The following diagram is an illustration of a greenhouse resistor network that models all major heat flows in resistor-capacitor form.

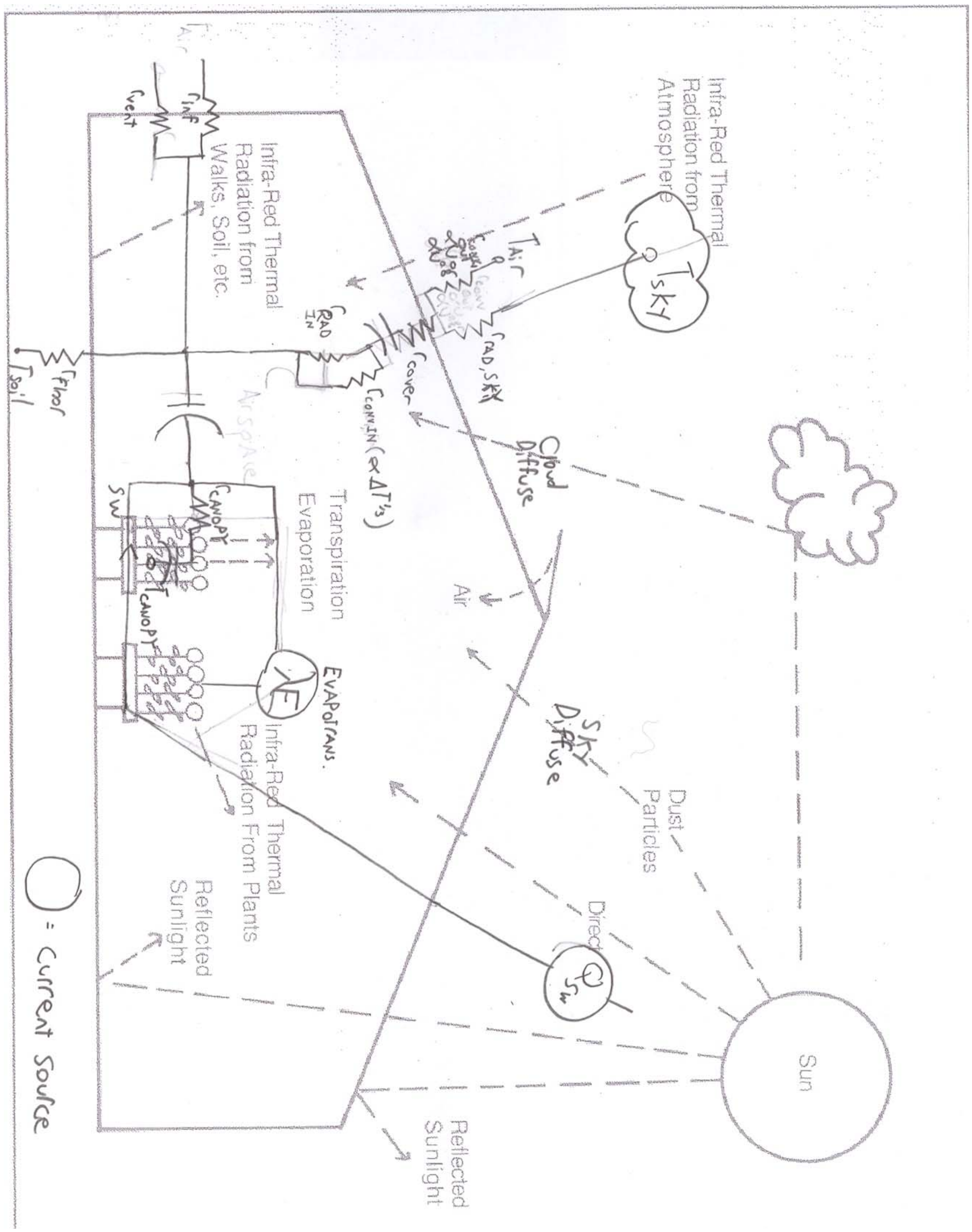


Figure 5 Resistor Diagram of Greenhouse. Annotations made by author. Original diagram from Aldrich and Bartok 1994.

Convection

Before heat can travel from a fluid to a solid boundary it must first be convected across the boundary layer. Major convection resistances occur at the canopy, and along the inner and outer walls of the greenhouse cover. Convection along the outer wall is modeled as turbulent forced convection, $R^{-1}\propto U^{0.8}$ or $R^{-1}\propto U$, where U is outdoor wind speed (Bot 1983, Kurata 1989). Convection along the inner wall is modeled as turbulent natural convection, $R^{-1}\propto(T_{glass} - T_{indoor})^{1/3}$. Convection along the canopy is modeled as mixed laminar forced and natural convection, length scale approx leaf or branch length, $R^{-1}\propto\Delta T^{1/4}$ or $u^{1/2}$.

The low indoor wind speed (order of 3 cm/s) results in the dominance of internal convection by laminar and natural modes, leading to high transfer resistances on the interior of the greenhouse versus its exterior. Even if the climate (temperature, humidity, and insolation) were the same, differences in resistance leads to vastly differing microclimate for indoor and outdoor grown plants.

Convective heat transfer coefficients vary widely in the literature, but generally tend to follow the trends mentioned above. A dated though still relevant review of greenhouse convective conductances can be found in Kurata 1989.

Advection: Ventilation and Infiltration

The advection of heat across the greenhouse envelope can be viewed as the sum of two processes: ventilation and infiltration; which differ in their controllability, energy requirement, and response to a change in pressure.

Recall, the heat removed by an advecting fluid:

$$Q = C_P \dot{V} \Delta T \tag{4.47}$$

Heat flow across a boundary envelope by advection, V_{dot} is volumetric flowrate: porosity*area*avg. velocity

For a given temperature drop, to determine total advective heat loss one must determine the flow rate, and to do that, a power-law relation with respect to pressure drop is often used. The net flow through any opening can be modeled as:

$$\dot{V} = A_{opening} C_d \left(\Delta p / \rho \right)^n \tag{4.48}$$

Flowrate (V_{dot}) through an opening, C_d = drag coefficient(0.6 for sharp crested orifice), Δp is pressure drop, ϱ = density, n = exponent(1= laminar flow, 0.5= orifice or fully turbulent flows, 0.67= crack flow)

Ventilation is mechanically controlled, intentional and often requires the use of energy. Because of the high Reynolds' number turbulence, ventilation flowrates are typically proportional to the square root of pressure drop. Ventilation can be mechanically induced by fans or naturally induced by wind pressure or density gradients.

Infiltration, on the other hand, is the uncontrolled and unintentional consequence of building materials and construction methods. It is almost always induced by natural forces, and requires no energy input. Very often infiltration is modeled as a constant value, but when more detailed methods are required, infiltration is modeled as being $\propto \Sigma$ (wind pressure + stack pressure [density differences]$)^{2/3}$. The 2/3 exponent can be derived from a combination of laminar pipe flow with sharp orifices at both sides.

The following table lists recommended infiltration rates for different construction types. As a form of standardization, infiltration and ventilation are expressed in non-dimensional air change units, where one air change is equal to the total volume enclosed by the greenhouse.

Table 4.4.1 – Natural air exchanges for greenhouses (ASAE, 1984).

Greenhouse construction system	Air infiltration rate (h^{-1})
New construction, glass or fibreglass	0.75–1.5
New construction, double layer plastic film	0.5–1.0
Old construction, glass, good maintenance	1–2
Old construction, glass, poor maintenance	2–4

GREENHOUSE CLIMATE CONTROL 181

Figure 6 Air Infiltration Rates, source: Bakker et al 1995

Since mechanical ventilation is driven by fans, energy must be consumed. The power required to move an air mass across a pressure gradient, is equal to Δp * flow rate:

$$P = \eta \Delta p \dot{V} \tag{4.49}$$

Fan power consumption, η is the fan efficiency

The solution to eq. (4.49) is presented on a fan curve available from the fan manufacturer. Sizing a fan involves calculating the pressure drop for the required flow rate using eq. (4.48) for sharp orifice flow and selecting the fan which most efficiently delivers the flow at the given pressure drop. Once this has been done, the marginal power consumption (Power/Flowrate) for a single fan or bank thereof can be determined. The marginal power consumption is used by GUESS to calculate the ventilation energy requirement per bank. GUESS assumes that ventilation is accomplished by: 1) a natural ventilation stage, 2) 3 settable banks of identical fans, with a single marginal power consumption.

In GUESS, fan control is mediated using a bang-bang type relay control. When the temperature exceeds the desired setpoint by an amount ≥ than the bandwidth, the fan will turn on. When the temperature falls below the setpoint by an amount ≥ the bandwidth, the fans will turn off. Dynamic behavior in response to step on/off signals is mimicked by a slew rate and a first order lowpass filter:

$$F(s) = \begin{cases} G(s)\ , & |sG(s)| < k_{slew} \\ \dfrac{k_{slew}}{s^2}\ , & |sG(s)| \geq k_{slew} \end{cases} \tag{4.50}$$

Slew rate in Laplace notation, G is input, and F is output.

$$\frac{F(s)}{G(s)} = \frac{1}{1 + (1/\tau)s} \tag{4.51}$$

First Order lowpass filter, where τ is the time constant.

Natural Ventilation

Natural ventilation is driven by a combination of wind and stack pressures. The expression for stack pressure will be included here for completeness (most useful for small-volume, tall and narrow greenhouses), in GUESS, only wind pressures are considered. The Bernoulli equation can be used to determine wind pressures.

$$P_{wind} = \tfrac{1}{2} C_p \rho u_{wind}^{\ 2} \tag{4.52}$$

Dynamic wind pressure where C_p is the pressure coefficient

C_p refers to the pressure coefficient and is equal to the ratio between the total dynamic pressure of the wind and the pressure read from a transducer mounted on a building wall. It corrects for angular effects, vena contracta, separation, turbulence, friction and other phenomena.

Typically pressure coefficients are determined at eave height which may or may not be equal to anemometer height, so a simple power law can be used to convert wind speeds.

$$\left(\frac{u_z}{u_{ref}}\right) = \left(\frac{h_z}{h_{ref}}\right)^a \tag{4.53}$$

Wind speed power law. Exponent, a varies with ground conditions and can range from 1/7(over flat water and ice) to 0.4(urban areas). For most rural areas, a is between 0.2 and 0.3.

For wall vents, the pressure coefficient is calculated using the following formula:

$$C_p = 0.6 * \ln \left\{ \begin{matrix} 1.248 - .703 \sin\left(\tfrac{\phi}{2}\right) - 1.175 \sin^2 \phi + 0.131 \sin^3 (2\phi) + \ldots \\ 0.769 \cos\left(\tfrac{\phi}{2}\right) + 0.07 \sin^2\left(\tfrac{\phi}{2}\right) + 0.717 \cos^2\left(\tfrac{\phi}{2}\right) \end{matrix} \right\} \tag{4.54}$$

Wind Pressure coefficient as a function of ϕ (incidence angle) for a vertical wall opening, $0° \leq \phi \geq 180°$, Burns and Deru 2003

For roof vents, a similar formula was not found in the literature, although eq. (4.54) might give approximate results. Once the pressure distribution around the building is known, the continuity equation is applied to get the ventilation rate.

$$\sum \rho_n u_n A_n = 0 \tag{4.55}$$

For two openings of equal area, equal height, and air density, eq. (4.55) reduces to

$$u_1 = u_2 \tag{4.56}$$

For natural ventilation, openings are assumed to behave like orifices, so:

$$u_n = C_D \sqrt{\Delta p / \rho} \tag{4.57}$$

ΔP is the difference in pressure between the external wind pressure and the internal pressure of the greenhouse, which is assumed to be a function of wind speed as well of the same form as eq. (4.52).

$$P_{wind,i} = \tfrac{1}{2} C_{p,i} \rho u_{wind}^{\ 2} \tag{4.58}$$

$C_{p,i}$ is unknown, but can be solved for quite easily using equations: (4.57) & (4.55). For two openings of equal height and area, which holds for most common vent installations, $C_{p,I}$ is simply the average of the pressure coefficient at the two vents. So in GUESS, the following formula is used to calculate flowrate due to natural ventilation:

$$\begin{aligned} \dot{V}_{VENT,NATURAL} &= A u_w \sqrt{\frac{(C_{Press,1} - C_{Press,2})^2}{\left| C_{Press,1} - C_{Press,2} \right|}} \\ Q_{VENT,NATURAL} &= \rho C_P (T_{in} - T_{out}) A u_w \sqrt{\frac{(C_{Press,1} - C_{Press,2})^2}{\left| C_{Press,1} - C_{Press,2} \right|}} \end{aligned} \tag{4.59}$$

Natural Ventilation flowrate through two equal height, equal area openings (Albright, 1990).

Evapotranspiration, Condensation, and Evaporative Cooling

These terms represent the interconversion of sensible and latent heat. The two major sources of evaporation in the greenhouse are evaporative cooling and canopy transpiration and soil evaporation.

Evapotranspiration

Canopy transpiration and soil evaporation can be modeled using the Penman-Monteith equation. In the Penman-Monteith there are two important resistances to vapor flow: aerodynamic, caused by inertial effects in the boundary layer surrounding the canopy, and canopy, caused by diffusion through the stomatal pores. To go from transpiration at the leaf level to total transpiration load per ground area, some form of scaling function is required, since radiation measurements are made on a per ground area basis, but transpiration occurs on a per leaf area basis. The Penman-Monteith equation was originally derived for outdoor weather conditions, where turbulence intensities are high, and aerodynamic resistances are small, but can be applied indoors provided that the appropriate scaling relations are made.

Outdoors, the canopy is treated as a uniform rough surface, and turbulence models are used to calculate aerodynamic resistances, which are typically much smaller than canopy resistances. For the canopy resistance, scaling from leaf to plant is trivial, it is assumed all leaves are wired in parallel, and $r_{canopy} = (1/LAI) * r_{leaf}$.

In a greenhouse, turbulence is greatly reduced and high aerodynamic resistances prevail. Relatively constant air velocities dominate, and both forced and natural convection and radiation heat transfer modes can occur. So one would expect scaling to occur in a different manner than it does outdoors. Little has been written so far on the topic of aerodynamic resistances at low wind speeds for conifer seedlings, see Landsberg(1970). From Landsberg's paper, the following can be discerned: For a given wind speed, as one scales from needle to branch to tree; two competing effects occur: increasing convection caused by parallel resistances, and decreasing convection caused by interfering boundary layers. Nonetheless, both branches and needles in isolation experience convective resistances proportional to $u^{0.5}$, which is similar to that of laminar flow over a flat plate, (Monteith & Unsworth, 1990)

For most greenhouse crops, the authors in Bakker et al. (1995) recommend using laminar flow over a flat plate for $r_{aerodynamic}$ and connecting them in parallel in the same manner as the stomata. This is how it is done in GUESS. When calculating the convection resistance, the branch length(l = 2cm) is taken as the characteristic dimension, to consider the effects of needle sheltering. For equilibrium transpiration (the only form considered by GUESS), the scaling factors (1/LAI) cancel out, and evaporation becomes independent of LAI.

Soil evaporation is also modeled using the Penman-Monteith equation with R_{net} = all radiation not intercepted or emitted by the canopy, with canopy resistance set to zero. Soil evaporation occurs predominantly at the surface, so corrections for relative humidity and a surface mass balance should be included. However, with small soil volumes of nearly saturated soil present in a nursery greenhouse, these corrections could be eliminated (Bot 1983). Taking this into account, and also realizing the minimal effect r_{canopy} has upon equilibrium evapotranspiration, we can unify soil evaporation and crop transpiration into a single flow with the ground area as the reference area.

In GUESS, the latent load due to evapotranspiration from both soil and plant is expressed simply as:

$$
\begin{aligned}
Q_{ET} &= -A_{floor}\eta_{utilization}\lambda E = -\frac{\Delta R_{net}}{\Delta+\gamma^*} \\
\dot{H}_{ET} &= -A_{floor}\eta_{utilization}E
\end{aligned}
\tag{4.60}
$$

Vapor and heat fluxes from evapotranspiration

As mentioned earlier, eq. (4.40), net isothermal radiation can be used in place of net radiation to eliminate the need to solve for surface temperature. In this case net isothermal radiation is simply the sum of the shortwave and longwave fluxes calculated with respect to air temperature.

$$
R_{net} = \alpha I + h_{r:cover}\varepsilon_{cover}(T_{cover} - T_o) + h_{r:sky}(1-\varepsilon_{cover})(T_{sky} - T_o) \tag{4.61}
$$

Net isothermal radiation on the canopy and soil

Evaporative Cooling

In a greenhouse, evaporative cooling devices are used reduce temperature when ventilation cannot achieve levels suitable for optimal plant growth. In greenhouses thus equipped, evaporative cooling constitutes the second portion of latent gain. Most evaporative cooling methods can be modeled as adiabatic cooling processes; the minimum temperature and maximum vapor pressure achievable are equal to that at wet bulb.

In GUESS, there are two possible evaporative cooling methods: pads and foggers. Although their limitations are similar (min. temp = wet bulb), they are modeled somewhat differently.

Cooling pads cool the outdoor to a temperature almost equal to wet bulb, the difference a function of the pad efficiency. The pad efficiency is function of the pore size and thickness of the pad. Typical pad efficiencies are about 85%. The rate of heat loss is dependent upon the fan speed.

$$\begin{aligned} H_{pad} &= H_{out} + \eta_{pad}(H_{wb} - H_{out}) \\ T_{pad} &= T_{out} - \eta_{pad}(T_{out} - T_{wb}) \\ Q_{PAD} &= \dot{V}_{FAN}\rho C_P \eta_{PAD}(T_{out} - T_{wb}) \end{aligned} \tag{4.62}$$

Cooling pad model equations

Foggers cool and humidify the internal greenhouse air. The fogging process is modeled as a mass transfer between the fog droplet and the air. The maximum humidity occurs at wet bulb, so the driving force is the difference between current vapor pressure and that at wet bulb. The conductance is the product of the boundary layer vapor conductivity and the total surface area of the mist, it is usually empirically determined. Ohm's law is used to determine the vapor and heat fluxes.

$$\begin{aligned} \dot{e} &= kA_{net}(VP_{sat}\left(T_{wb}\left[T_{air}, rh_{air}\right]\right) - VP_{air}) \\ q &= \lambda \dot{e} \end{aligned} \tag{4.63}$$

Model of fogging system

Condensation

Condensation works in a similar manner to the fogger, the rate of deposition being controlled by the difference between current vapor pressure and saturation vapor pressure at the inside wall of the cover. The cover temperature is a function of both indoor and outdoor temperatures. Very often, it is assumed that on average the cover temperature is 2/3 outdoor temperature, and 1/3 indoor temperature (see Bakker et al 1995). In the model validation section, a case study was conducted for a more thorough treatment of cover temperature.

$$\frac{r_{CONV,out} + r_{COND,cover}}{r_{CONV,in} + r_{COND,cover} + r_{CONV,out}} T_{in} + \frac{r_{CONV,in}}{r_{CONV,in} + r_{COND,cover} + r_{CONV,out}} T_{out} = T_{cover} \tag{4.64}$$

Equation for cover temperature

According to Bakker et al. (1995), the conductance to vapor transfer at the greenhouse cover is primarily caused by natural convection. Because of the large size of the cover, the convection is typically turbulent, becoming independent of length scale.

An equation for cover conductance is provided below:

$$g_{cond} = 1.64E\text{-}3 * \Delta T^{1/3} \quad \mathrm{m \cdot s^{-1}} \tag{4.65}$$

The above equation was derived for standard conditions (T_{indoor}= 295K and P_{atm}= 101 kPa). To account for the enhancement in natural convection caused by mass flux and density differences, the virtual temperature difference is used for ΔT.

$$\tilde{T} = \frac{T}{1 - .379\dfrac{VP}{P_{atm}}} \tag{4.66}$$

Virtual temperature

Using equations (4.65) and(4.66), and recognizing that equation (4.64) is a statement about the partitioning of heat flux between the inside and outside of the glass, expressions for the both mass flow and heat gain due to condensation can be obtained:

$$\begin{aligned} \dot{e}_{cond} &= \underbrace{k_{cond} A_{cover} [VP_{in} - VP_{sat}\left(T_{cover}\right)]}_{\text{condensation}} \\ \dot{q}_{cond} &= \underbrace{\lambda \frac{r_{CONV:out} + r_{COND:cover}}{r_{CONV:in} + r_{COND:cover} + r_{CONV:out}} * \dot{e}_{cond}}_{\text{heat gain from condensation}} \end{aligned} \tag{4.67}$$

Vapor flux and sensible gain due to condensation

To solve for cover temperature, another energy balance is required, however it is typically observed that cover temperature is a linear function of both indoor and outdoor temperatures. A linear function suggested by Stanghellini in Bakker et al (1995) is used

$$T_{cover} = \frac{2}{3}T_{outdoor} + \frac{1}{3}T_{indoor} \tag{4.68}$$

Equation for estimating cover temperature

Heaters

All heaters in GUESS are modeled as a constant flux per stage. No attention is paid to the internal transfer processes. Three stages are provided, and the amount of heat per stage is expressed as a fraction of the total heat output, the sum totaling one. To model dynamic response to step on/off signals, the same model as that of the fans is used.

The energy requirement for operating a heater is the sum of the fan power to drive the air (forced draft systems), and the fuel burned to produce the actual heat. The default model used in the GUESS treat the heating system as a methane-fired condensing boiler. After condensing, a portion of the flue gas is returned to the greenhouse to serve as CO_2 source, the remainder leaves via the stack. The fraction sent to the greenhouse is termed the return ratio, *r*.

The heat produced per unit fuel is modeled as:

$$h_{\substack{combustion\\sensible}} = LHV + \lambda\phi * \left[\tfrac{36}{16}\phi^{-1} - e_{sat}\left(T_{exhaust}\right)\right] - (1-r)C_{P,air}T_{exhaust} \tag{4.69}$$

Sensible heat load from a condensing heater. LHV is the lower heating value. Φ is the air to fuel ratio, 36/16 is the mass ratio of water vapor produced to fuel burnt. $T_{exhaust}$ is the exhaust temperature and r is the return ratio.

Control

Control of greenhouse systems are mediated by simple bang-bang controllers. When the error in temperature: $T - T_{setpoint}$ (cooling) or $T_{setpoint} - T$(heating) exceeds the bandwidth, the device is set to the "ON" state, when it is below, the device is set to the "OFF" set. To achieve some measure of gradation in temperature control, up to three settable stages are available for heating and fogging, and up to four are available for ventilation (one natural, three mechanical). Temperature data, for simulating control, is sampled every n time steps with a zero-order hold method.

Synthesis

A complete dynamic model of all major greenhouse components (canopy, benches, cover, air space, floor, and pots and soil) simulating a year's worth of production would provide an excess of details, and be prohibitively slow to execute even using today's computers, so some form of simplification is required. For control purposes, only air temperature is relevant. For physiological purposes, the relevant temperatures are at leaf and root level.

In GUESS, two procedures are to simplify the calculation: static equilibrium models, and dynamic lumping. If indoor air space is the reference used to determine time step (Δt), then any component which has a time constant (RC) greater than Δt should be modeled dynamically. All others can be modeled statically, whereby $dT/dt = 0$. According to Bot (1983), in most greenhouses, only the soil has a time constants longer than the air space. In particular, a static model is often used for leaf temperature to determine photosynthesis rates (Riha 2004).

The other simplifying procedure is lumping, which means treating several objects to be at a single temperature (in this case air temperature). While somewhat dubious, lumping works well enough for basic energy use and climate modeling. Lumping can also be to internal objects if time constants and steady temperatures are similar. Biot numbers can be used to characterize the validity of the lumping assumption. In GUESS, the air floor, soil and pots are lumped together at air temperature.

$$\overbrace{\rho_{AIR} V_{GH} C_{P,GH} \frac{dT_{in}}{dt}}^{STORAGE} = \overbrace{\underbrace{\alpha_{SW} \tau_{glass} I}_{\text{Shortwave Solar}} + Q_{heaters} + \underbrace{\frac{r_{CONV,out} + r_{COND,cover}}{r_{CONV,in} + r_{COND,cover} + r_{CONV,out}} \lambda k_{cond} A_{cover} [VP_{in} - VP_{sat}\left(T_{cover}\right)]}_{\text{condensation}}}^{GAINS} - \ldots$$

$$\overbrace{\underbrace{h_{r:sky}(1 - \varepsilon_{cover})(T_{in} - T_{sky}) - 0.8 \varepsilon_{cover} h_{r:cover}(T_{in} - T_{cover})}_{\text{longwave}} - \underbrace{A_{floor} \eta_{utilization} \frac{\Delta R_{net}}{\Delta + \gamma^{*}}}_{\text{Evapotranspiration}}}^{LOSSES} - \ldots$$

$$\overbrace{\underbrace{\left[\frac{1}{r_{CONV,in} + r_{COND,cover} + r_{CONV,out}} A_{cover} + P_{floor} \left(U\ell\right)_{perimeter}\right] (T_{in} - T_{out})}_{\text{Conduction}} - \underbrace{\lambda k A_{net} (VP_{sat}\left(T_{wb}\left[T_{air}, rh_{air}\right]\right) - VP_{air})}_{\text{Foggers}}}^{LOSSES\ CONT.}$$

$$\overbrace{- \underbrace{\rho C_{P,air} \dot{V}_{INF} (T_{in} - T_{out}) - \rho C_{P,air} \dot{V}_{VENT} (T_{in} - T_{pad})}_{\text{Advection}}}^{LOSSES\ CONT.}$$

Equation (4.70) Complete GUESS heat transfer model, e_{sat} in the combustion submodel refers to saturated humidity ratio (kg vapor/kg air)

The above model while complete (includes all major sources/sinks) is not in closed-form, for T_{cover} is unknown. A separate model is needed to determine cover temperature, which could be either dynamic or static. Because of radiation and natural convection, a static model would be implicit and non-linear requiring a Newton-Rapheson solution method. A dynamic method would be explicit, and faster per individual solution, no iterations required, but may require an increase in number of time steps to achieve stability. The simplest method, and the one implemented in GUESS is to roll the conduction, convection, and radiative conductances of the cover into one universal cover conductance. For covers with high IR transmissivity, such as single film polyethylene; GUESS incorporates a parallel conductance that represents radiation to the sky. This eliminates solving for cover temperature altogether (although an empirical correlation like eq. (4.68) is needed to calculate condensation resistance).

The one serious disadvantage of this approach is that it ignores the effects of outdoor wind speed and sky conditions upon envelope heat transfer rates. For long-term measurements and energy conservation studies, “universal conductances” are more than adequate, but for short-term dynamic models with changing wind speeds and sky temperature, significant errors could result. To test the validity of this assumption a case study was conducted for a glasshouse, see **Model Validation**.

Mass transfer in the Greenhouse

To determine greenhouse climate, two species need to be considered: CO_2 and water vapor. Radiation does not occur and the cover is impervious enough that conduction can be ignored. Thus, only ventilation and infiltration are considered for transferring mass in and out of the greenhouse. At the canopy level, conduction and convection through the stomata and boundary layer need to be considered, the details of which will be described later.

The mass balance for humidity in the greenhouse can be written as:

$$\rho_{air} V_{Greenhouse} \frac{de_{in}}{dt} = -\dot{V}_{INF} \rho_{air} (H_{in} - H_{out}) - \dot{V}_{VENT} \rho_{air} (H_{in} - H_{pad}) + \underbrace{\tfrac{1}{\lambda} A_{floor} \eta_{utilization} \frac{\Delta R_{net}}{\Delta + \gamma^*}}_{\text{Evapotranspiration}}$$

$$- \underbrace{k_{cond} A_{cover} [VP_{in} - VP_{sat}(T_{cover})]}_{\text{Condensation}} + \underbrace{kA_{net} (VP_{sat}(T_{wb}[T_{air}, rh_{air}]) - VP_{air})}_{\text{Foggers}} + \underbrace{r\phi e_{sat}(T_{exhaust}) \frac{Q_{heat}}{h_{combustion}}}_{\text{Combustion}}$$

Equation (4.71) Humidity Mass Balance

Likewise, a similar mass balance could be written for CO_2:

$$\rho_{air} V_{Greenhouse} \tfrac{1000}{29} \frac{dC_{CO_2 in}}{dt} = -\rho_{air} \tfrac{1000}{29} (\dot{V}_{INF} + \dot{V}_{VENT})(C_{CO_{2\,in}} - C_{CO_{2\,out}}) + \dot{F}_{photosynthesis} - \ldots$$

$$\dot{F}_{respiration} + \underbrace{r\zeta \tfrac{1000}{MW_{fuel}} \frac{Q_{heat}}{h_{combustion}}}_{\text{Combustion}}$$

Equation (4.72) CO_2 Mass Balance in molar units (ppm or μmol CO_2 per mol air). ζ is the number of moles of carbon per mole of fuel

To determine the mass balance of CO_2 in the greenhouse, one needs to consider the effect of physiological fluxes ($F_{photosynthesis}$ and $F_{respiration}$). To determine them, a process based model of plant growth is needed. Also note that the CO_2 mass balance is expressed in molar units of concentration, since photosynthesis models are calibrated to molar units of light and CO_2.

Plant Processes

The insight gained from the model of greenhouse climate may lead to improvements in energy efficiency. However these improvements are meaningless unless they are interpreted in the context of the crop's health and productivity. Thus a complete greenhouse model must not only model the climate control systems but the crop systems as well.

The net growth rate of a plant is its rate of biomass acquisition, and in the simplest terms, a mass balance of carbon. Carbon is acquired from the atmosphere by photosynthesis and converted to carbohydrate. A portion of that carbohydrate gets respired and returned to the atmosphere; the remaining carbohydrate gets used to synthesize new biomass.

$$\frac{dM}{dt} = C\left(P - R\right) \tag{4.73}$$

Biomass(M) Balance. P is photosynthesis, R is respiration, and C is the conversion from CO_2 to biomass

$$6CO_2 + 6H_2O + \text{light} \rightarrow C_6H_{12}O_6 + 6O_2 \tag{4.74}$$

Chemical equation for Photosynthesis

$$C_6H_{12}O_6 + 6O_2 \rightarrow CO_2 + 6H_2O \tag{4.75}$$

Chemical equation for Respiration

Photosynthesis

Photosynthesis is a far more complicated process than Eq. (4.74) suggests. The fixation of CO_2 and subsequent conversion to carbohydrate is not a single reaction, but a series of steps, the Calvin cycle (see diagram below).

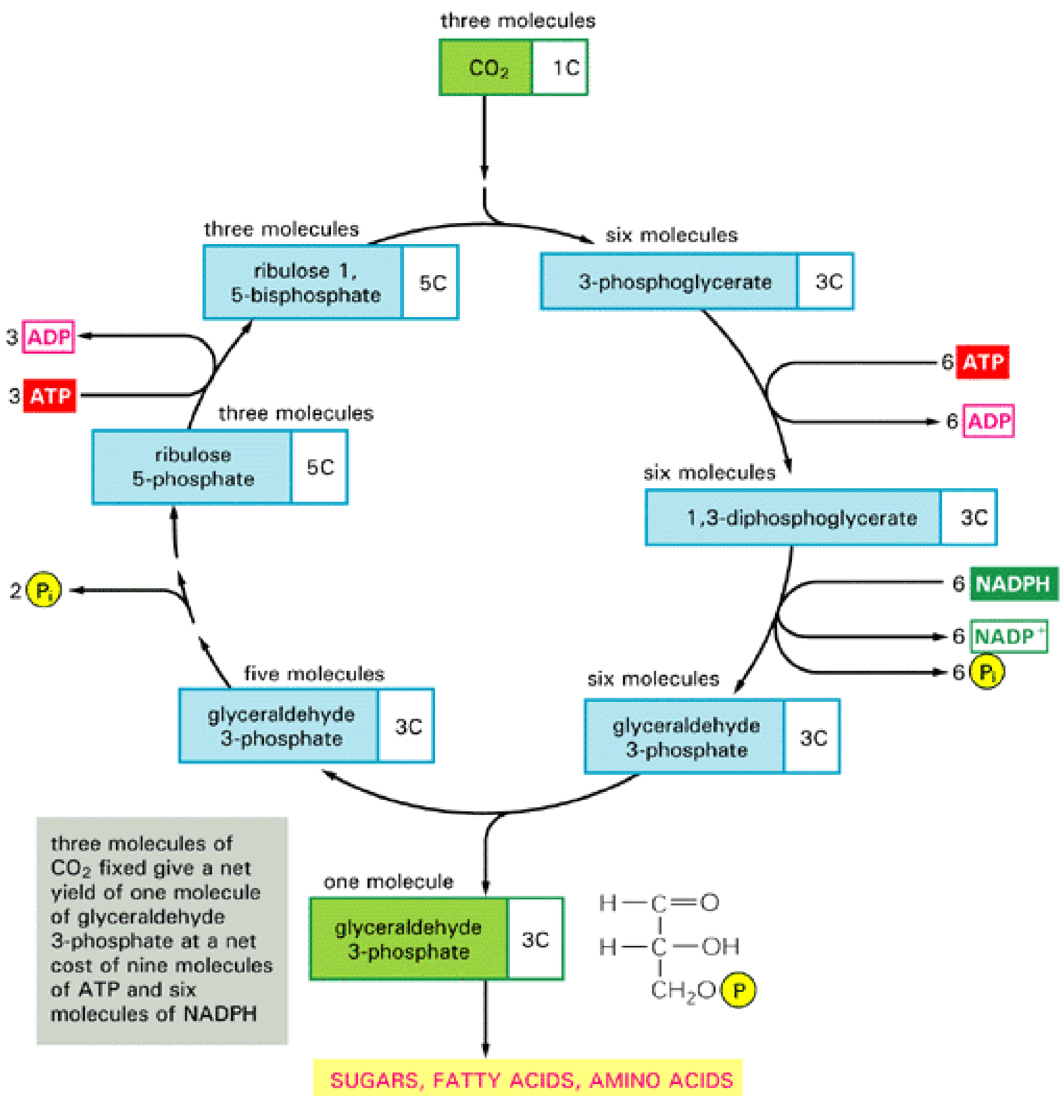


Figure 7 Calvin Cycle Diagram. The reaction at the very top(fixing CO_2 and RuBP) is catalyzed by the enzyme, Rubisco. This reaction controls the rate of carbon assimilation, and is the one modeled by

the Farquhar et. al. equations. Source: *Cellupedia*, "Calvin Cycle" http://library.thinkquest.org/C004535/calvin_cycle.html

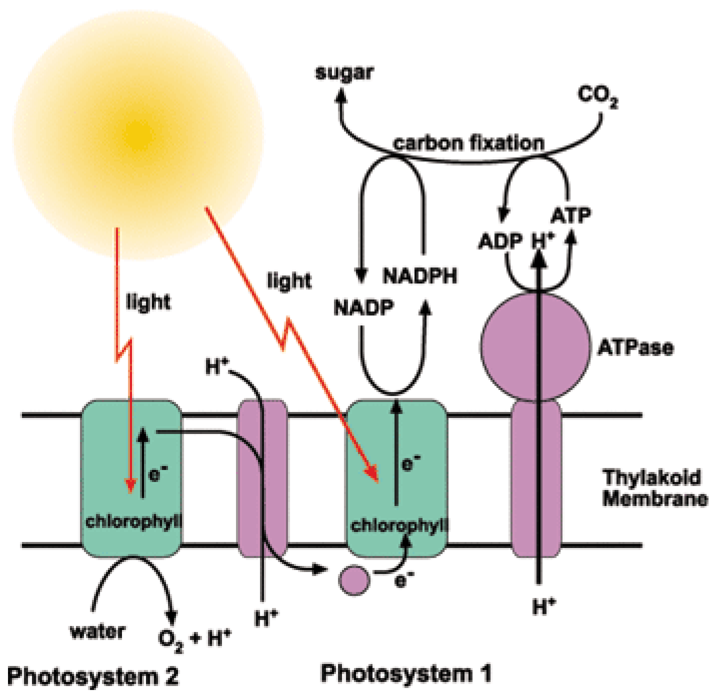


Figure 8 Schematic of the light or electron transport reactions. The light reactions produce ATP & NADPH which regenerate the RuBP in the Calvin Cycle. Source: *Cellupedia*, "Photosystems" http://library.thinkquest.org/C004535/photosystems.html

For the sake of modeling, however, only the rate limiting reaction need be considered. In this case, the reaction is primarily governed by the step at the top of the diagram: the reaction between ribulose bis-phosphate (RuBP) and CO_2, which is facilitated by the enzyme Rubisco. The reaction rate is limited in four ways: supply of ATP and NADPH to regenerate RuBP, supply of CO_2, supply of free Rubisco, and the competing reaction of Rubisco with oxygen: photorespiration.

Farquhar, von Caemmerer, and Berry (1980) were among the first to synthesize the available knowledge of photosynthesis biochemistry into a useful mechanistic model of C_3 photosynthesis that worked well with experimental data. Despite its complexity, the original Farquhar et al. model and its various versions have become the most widely used models for photosynthesis.

The large body of data available for parameterization, widespread use by other modelers (Baldocchi, 2004) and flexibility are among the reasons for the implementation of the Farquhar model in GUESS.

In the Farquhar model, half the rate of oxygenation (photorespiration) is subtracted from the rate of carboxylation to get the rate of photosynthesis. It is assumed that photosynthesis follows the law of limiting factors, where the slowest process dominates the rate of reaction. Since there are two substrates (CO_2 and RuBP) there are two processes that can limit photosynthesis: electron transport for the generation of RuBP, and carboxylation catalyzed to fix CO_2 together with RuBP.

$$\begin{aligned} P &= V_c - 0.5V_o \\ V_c &= \min\left\{W_c, W_j\right\} \\ P_{net} &= P - R_d \end{aligned} \tag{4.76}$$

The Farquhar Model. V_c is the rate of carboxylation, and V_o is the rate of oxygenation. V_{cc} and V_{cj} refer to the rates of carboxylation limited by Rubisco and electron transport, respectively. P, P_{net}, and R_d refer to photosynthesis, net photosynthesis, and leaf respiration and are expressed as fluxes on a leaf area basis.

Whenever oxygen is present, the competing reaction of photorespiration can occur. A CO_2 compensation point, Γ, where $V_c = 0.5*V_o$ is used to account for the photorespiration term in the Farquhar et al model.

$$P = \left(1 - \frac{\Gamma}{C_i}\right) * \min\left\{W_c,\ W_j\right\} \tag{4.77}$$

Farquhar model with Γ, CO_2 compensation point.

The limiting process in the Farquhar model, the "**min**" term could be modeled in three ways: a rectangular hyperbola, the discontinuous Blackman response, and a non-rectangular hyperbola. A rectangular hyperbola transitions gradually and smoothly from one regime to another, this is the model used for Michaelis-Menten kinetics, however it saturates too slowly to model photosynthesis (Marshall and Briscoe 1980):

$$P = \frac{W_c + W_j}{W_c W_j} \tag{4.78}$$

Rectangular Hyperbola model of photosynthesis

The Blackman response transitions abruptly and discontinuously from one rate limiting step to the other, and better approximates actual photosynthetic behavior.

$$P = \begin{Bmatrix} W_c & , & W_c \le W_j \\ W_j & , & W_c > W_j \end{Bmatrix} \tag{4.79}$$

Blackman response

In reality although photosynthesis transitions sharply from Rubisco limiting to electron transport limiting, it is never discontinuous. So a third model, first suggested by Marshall and Biscoe (1980) is used that is hybrid between the Michalis-Menten and Blackman curves:

$$\theta P^2 - P(W_j + W_c) + W_j W_c = 0$$
$$0 \le \theta \le 1 \qquad (4.80)$$
$$P \ge 1$$

Equation for non-rectangular hyperbola. θ is the smoothing parameter, and it indicates sharpness of the transition: use 0 for Michaelis-Menten and 1 for Blackman.

$$P = \frac{-(W_j + W_c) + \sqrt{(W_j + W_c)^2 - 4\theta W_j W_c}}{2\theta} \qquad (4.81)$$

Analytical solution to equation (4.80). When θ falls between 0 and 1, the quadratic formula(above) can be used. For θ = 0 or 1, use (4.78) or (4.79).

Typically for C_3 plants, θ falls between 0.8 and 0.95.

Carboxylation Limited Photosynthesis

When RuBP levels are high, the rate of photosynthesis is governed by the amount of CO_2 present in the chloroplast. Here, one can use classical Michaelis-Menten kinetics to model the carboxylation reaction, considering competitive inhibition by O_2.

$$W_c = \frac{V_{c_{\max}} C_i}{C_i + K_c(1 + O / K_o)} \qquad (4.82)$$

Reaction for photosynthesis when CO_2 concentration is limiting. V_{cmax} is property of the amount of free unbound Rubisco present in the system. C_i is the **internal** CO_2 concentration. O is the oxygen concentration. K_o and K_c are kinetic parameters of Rubisco and tend to remain constant across species.

Carboxylation limited photosynthesis usually occurs whenever C_i, temperature or light levels are low, Baldocchi 2004.

Electron Transport Limited Photosynthesis

The absorption of photons by chlorophyll causes the release of electrons in the chloroplast. The energy generated by the electron flow powers the synthesis of ATP and NADPH, which are the raw materials for regenerating RuBP. So a model of light-driven electron transport can be used to model RuBP-limited photosynthesis.

Typically, electron transport displays linear kinetics at low light ($J = \alpha I$) and saturation ($J = J_{max}$) at high light, with the transition typically happening quicker than a rectangular hyperbola. The constant, α is termed the quantum efficiency, and it represents how many moles of electrons are moved down the transport chain mole per incident mole of photons required to move a mole is termed the quantum efficiency, α. It is measure of the maximum rate of photosynthesis under non-saturating conditions. The theoretical value for α is ¼, however typical values are about 90-95% of that (Baldocchi 2004).

Equation (4.80) could be used to model the transition. In GUESS, another type of saturation curve, suggested by Harley & Tenhunen (1991) is used.

$$J = \frac{\alpha I}{\sqrt{1+\left(\frac{aI}{J_{max}}\right)^2}} \qquad (4.83)$$

Electron transport model, where J is the electron flux transport (μmol electrons/s). I is the incident radiation in μmol PAR. α is the quantum efficiency of photosynthesis: mol electrons/mol photons), theoretical value: 0.25, typical value: 0.23. J_{max} is the maximum rate of electron transport.

For every molecule of CO_2 fixed, theoretically four electrons are required, but because of the competing reaction of oxygenation, the ratio of e^- to CO_2 is typically higher. The following function is used to correct for oxygenation and convert from electron transport to CO_2 fixation:

$$W_j = J\left(\frac{C_i - \Gamma}{4C_i + 8\Gamma}\right) \quad \text{when} \quad V_c = W_j \tag{4.84}$$

Conversion from electron transport rate to photosynthesis rate.

Dark Respiration

Dark respiration refers to maintenance respiration performed by the leaf during photosynthesis. As a general rule: thicker, more protein-rich leaves tend to show higher values of R_d. Because of its direct effect upon internal CO_2 conc. it is often lumped in with photosynthesis to yield P_{net}. For a given leaf, its variation can be modeled as a function of temperature only (Walcroft et al 1997).

$$R_D = R_{D_0} \exp\left(\frac{H_a}{T_{ref}} - \frac{H_a}{RT}\right) \tag{4.85}$$

Expression for Dark Respiration

Temperature Response

J_{max}, and V_{cmax} show significant variation with temperature. Like most biological reactions the response is bell-shaped, with a relatively flat optimal region, and an exponential incline at low temperatures, and an exponential decline at high temperatures. The following function is used by Leuning (1995) to model temperature response for J_{max}, V_{cmax}:

$$\frac{V(T)}{V(T_{ref})} = \frac{\exp\left(\frac{H_a}{RT_{ref}} - \frac{H_a}{RT}\right)}{1+\exp\left(\frac{ST - H_d}{RT}\right)} \tag{4.86}$$

Temperature response model #1. H_a and H_d are energies of activation and deactivation. S is the entropy change for the reaction. Temperatures must be expressed in degrees Kelvin.

V_{cmax} and J_{max} are not the only parameters that show temperature variation: K_c and K_o and Γ show significant variation as well. Since K_c and K_o are properties of the Rubisco enzyme they show a much simpler Arrhenius-type response to temperature that remains constant across species (Baldocchi 2004):

$$\frac{k(T)}{k(T_{ref})} = \exp\left(\frac{H_a}{RT_{ref}} - \frac{H_a}{RT}\right) \tag{4.87}$$

Temperature response model #2: The Arrhenius curve

Γ also varies with temperature. Unfortunately, its response is far more complicated depending on the rates of V_o, V_c, and the differential solubility of CO_2 and O_2. Simple analytical expressions like eqs. (4.86) & (4.87) cannot be used, instead one makes use of regressions. In GUESS, the following polynomial regression, suggested by Leuning, 1995 is used:

$$\Gamma(T) = \gamma_0[1+\gamma_1(T-T_0)+\gamma_2(T-T_0)] \tag{4.88}$$

CO_2 compensation conc. γ are regression constants and T_0 is the reference temperature.

Synthesis

Although the Farquhar et al. model is a seven parameter model, two of the parameters K_o and K_c are properties of Rubisco and remain constant across species. Γ and α also show little variation among species (Farquhar and Brooks 1984), so only three parameters need be measured per species: V_{cmax}, R_d and J_{max}. Since they are non-linear, implementing the Farquhar equations in GUESS, like implementing psychrometrics involves the creation of lookup tables and linear interpolation.

Integrating From Leaf to Canopy

The next step is to apply the Farquhar equations to the entire canopy to calculate the total rate of photosynthetic production. The simplest procedure that comes to mind is that of a big leaf: summing all the leaves into one large area, and taking the average radiation absorbed and using that to determine net rate of assimilation. As easy as the big leaf approach may be, it gives incorrect results, differing from actual measurements by as much as 25% (dePury & Farquhar 1998). Thus a different technique is required, one that takes into consideration the non-linear nature of the net photosynthesis curve.

This simplest model that can account for this non-linearity is the two leaf (sun/shade) model. For the two leaf model, we divide the canopy into two leaf classes: those that receive direct radiation (sun leaves), and those that only receive diffuse and scattered radiation. Although many plants do have different sun and shade leaf morphologies, here the difference between the two types is only radiometric, as such a single set of values for J_{max}, V_{cmax}, and R_d are used. We denote the leaf area indices of the sun and shade classes as LAI_{sun} and LAI_{shade}, respectively:

$$LAI_{total} = LAI_{sun} + LAI_{shade} \tag{4.89}$$

Division of the canopy into sun and shade leaves

After dividing the leaf area into the two classes, we calculate the average incident radiation, use the Farquhar et al. (1980) model to determine the net photosynthetic flux for each class, and then multiply this flux by the leaf area LAI * A_{ground} to get the net carbon gain per plant. This procedure is formally outlined in Campbell 1998 and in dePury and Farquhar 1997 and briefly described below.

Incident Radiation

The average radiation incident on a sun leaf is:

$$Q_{sun} = Q_{direct,0} K_{be}(\theta) + \overline{Q_{diffuse}} + \overline{Q_{scattered}} \tag{4.90}$$

Sun leaf irradiance.

Likewise, the average radiation incident on a shade leaf is:

$$\overline{Q_{shade}} = \overline{Q_{diffuse}} + \overline{Q_{scattered}} \tag{4.91}$$

Shade leaf irradiance

The reason for why K_{be}(extinction coefficient for black leaves) * $Q_{direct,0}$ (direct radiation at the top of the canopy) instead of some average value, is due to the finite and heterogeneous nature of absorption. At the element (leaf) scale, a unit area of irradiance is either completely absorbed by a sun leaf, or it is completely transmitted as a sunfleck, recall eq. (4.21). Beer's Law assumes continuous absorption which may be the case for overall transmission to the understory, but the assumption breaks down at the leaf scale, as one would expect.

Any direct shortwave radiation not absorbed by a shade leaf is either reflected to the sky, transmitted through the leaf, or scattered by the multiple reflections within the canopy. Scattering then can be viewed as the difference between what would be absorbed if the canopy where composed of black leaves, and what is absorbed in reality. At the top of canopy, it is assumed that all scattered irradiance leaves towards the sky, and scattering incident on the canopy is zero, whereas at the bottom all radiation not absorbed by the canopy is available as scattered irradiance. We can use the expressions for a canopy of black leaves and non-black leaves to compute the average scattered irradiance.

Black Leaf Model:

$$\tau_{canopy,B} = \exp\left[K_{be}\left(\theta\right)\right] \tag{4.92}$$

Non-black Leaf Model:

$$\tau_{canopy,NB} = \exp\left[\sqrt{\alpha}K_{be}\left(\theta\right)\right] \tag{4.93}$$

Subtracting the transmissivities and integrating over the entire canopy:

$$\overline{Q_{scattered}} = \tfrac{1}{2}Q_{direct,0}(\tau_{canopy,NB} - \tau_{canopy,B}) \tag{4.94}$$

Average scattered irradiance

Like direct radiation, diffuse radiation is attenuated exponentially, but due to its more isotropic nature, the outer canopy is illuminated relatively uniformly. Hence heterogeneous effects like sunflecks are not as noticeable as they are with direct radiation and we can integrate Beer's Law over the entire canopy and divide by LAI_{total} to get the average diffuse radiation incident on a leaf within the canopy.

$$\overline{Q_{diffuse}} = \frac{Q_{od}[1-\exp\left(-\sqrt{a}K_{diffuse}LAI_{total}\right)}{\sqrt{a}K_{diffuse}LAI_{total}} \tag{4.95}$$

Average canopy diffuse irradiation

Sun and Shade Leaf Area Indices

The fraction of the canopy that is not illuminated by direct radiation is simply $1-\tau_b$, and τ_b can be calculated using eq (4.93). Dividing that expression by the absorption coefficient for a sunlit leaf(4.90), which is simply the extinction coefficient, we arrive at an expression for the sunlight LAI.

$$LAI_{sun} = \frac{1-\exp\left[-K_b\left(\psi\right)\right]}{K_b\left(\psi\right)} \tag{4.96}$$

Expression for Sunlight Leaf Area Index

Sunlit leaf area index can be calculated using the above equation and equation (4.89) is to determine the shade leaf area index. Equations (4.90) – (4.95) can then be applied to calculate average irradiances upon sun and shade leaves. Using this information, the Farquhar equations can be applied and net canopy photosynthesis estimated.

Respiration

Respiration is the conversion of carbohydrates into energy and CO_2, and is the inverse of photosynthesis. Respiration plays an important role in recharging the depleted greenhouse atmosphere with CO_2. Plant respiration can be modeled as two separate processes: growth respiration and maintenance respiration.

Growth Respiration

Growth respiration is the release of CO_2 during the synthesis of new tissue. It is independent of temperature and is usually a constant fraction of net assimilation: P_{net} - $\Sigma R_{maintenance}$. Typically the constant is taken to be 0.25 (Riha 2004).

Maintenance Respiration

Maintenance respiration is release of CO_2 due to metabolic activity, it shows an exponential response to temperature, and is measured on a per mass basis. Due to the complex nature of the temperature response, a piecewise function is used as opposed to the Arrhenius-type equations (4.86) and(4.87). The function typically used is the Q_{10} curve, where Q_{10} refers to the percentage increase in respiration for an increase in temperature of 10°C. While Q_{10} can vary with temperature & and other conditions, using a single value of approx 2.0, is reasonably accurate for the range of biological temperatures, 10°C - 40°C (Tjoelker et al 2001).

$$R(T) = R_{ref} * Q_{10}^{\frac{T - T_{ref}}{10}} \tag{4.97}$$

Q_{10} curve for the estimation of respiration rates

Respiration rates tend to differ among tissue types and with tissue age, so each tissue must be modeled separately. In GUESS, tissue is divided into three pools: leaf, stem and root.

$$\Sigma_{maintenance} = R_{leaf} + R_{stem} + R_{root} \tag{4.98}$$

Since leaf respiration is already taken by eq (4.85) in the Farquhar model, the Q_{10} function is only needed for the root and stem.

Allometry and Partitioning

Using the concepts outlined in the previous section, a carbon mass balance can be created, and this mass balance can be used to estimate net growth rate:

$$\frac{dM_{total}}{dt} = \varphi_{carbon}\left(P_{net}A_{canopy} - M_{root}R_{root} - M_{stem}R_{stem}\right)\left(1-\phi_{growth}\right) \tag{4.99}$$

Biomass balance: φ_{carbon} is the carbon content of dry biomass (mmol C/g d wt). ϕ is amount of amount of growth respiration (mol CO_2/ mol biomass).

The biomass balance is not in closed form. A simple mass balance says nothing about how the tissues are being partitioned to the different pools. Moreover, total fixed carbon is of little interest to the grower. In the case of tree seedlings, easy to measure, non-destructive indices such as height and stem diameter would be more appropriate measures of growth.

With crops like tomatoes and corn, allocation can be performed on a leaf-by-leaf, fruit-by-fruit, ear-by-ear basis; but with trees, this method becomes too complex to use (Riha 2004). Instead allometric relationships between biomass and dimension are used.

Partitioning

Canopy

Although leaf mass and thickness vary throughout the canopy, the total leaf mass is linearly proportional to the canopy area (Rippulone et al 2003). For simplicity, canopy area is normalized to ground area, and expressed using the leaf area index:

$$M_{canopy} = SLA * LAI * A_{ground} \tag{4.100}$$

Leaf Mass equation: SLA is specific leaf area (g leaf dw/m^2 leaf area). LAI is leaf area index (m^2 leaf area/ m^2 ground area).

However without knowing how much biomass gets devoted to the stem, we cannot formulate a differential equation for the canopy and close the biomass balance.

Roots

There is evidence showing that plants allocate biomass to the root on a functional equilibrium basis (Riha 2004). Basically, the flux of water and nutrients coming from the root should equal the rate of dissipation at the canopy, so low levels of soil water and nitrogen should lead to comparatively higher rate of root growth.

$$\frac{R_{root}E}{\psi_{soil} - \psi_{plant}} = A_{root}$$
$$dM_{root} \propto \frac{1}{d\left(\psi_{plant} - \psi_{soil}\right)} \tag{4.101}$$

Functional equilibrium for root growth with respect to soil-plant water balance. E is evapotranspiration rate, and ψ are water potentials.

For seedlings in a well-watered, well-fertilized greenhouse, the allocation to the root is approximately constant (personal communication, Albright) and the functional equilibrium model can be ignored. In GUESS, a constant root allocation of 16% is assumed.

$$\frac{dM_{root}}{dM_{total}} \in \{0.15 \ldots 0.25\} \tag{4.102}$$

Typical greenhouse root allocation levels

Stem

In their 1964 study, Shinozaki et al. concluded that area is conserved throughout the stem of a tree. His theory, termed the pipe model is the basis for most tree morphology models. The name pipe derives from the observation that each branch can be viewed as a continuous pipe from the root zone to the leaf zone, and the trunk is an aggregate of all the pipes. If there is a constant ratio of unit wood area per leaf area, then according to the pipe model, basal area is linearly proportional to canopy area:

$$A_{base} = C_{pipe} * A_{canopy} \tag{4.103}$$

Pipe model equation

Allometry

A common observation made by foresters is that there is a log-linear relationship between total stem mass and basal area; and between stem mass and basal diameter[D]^2 * height[H]. While there is some variation among species and age class (seedling versus mature, for example), the average slope for mass vs. basal area is 2/3, and for mass vs. D^2H, the average slope is 1, which is what one would expect from dimensional analysis (Parde 1980).

$$
\begin{aligned}
&M_{stem} = kA_{base}^{\frac{3}{2}} \\
&\text{where} \\
&A_{base} = \tfrac{\pi}{4} D^2
\end{aligned}
\qquad (4.104)
$$

Allometric relationship between stem basal area and stem mass

Using equation (4.104), we can solve for the differential biomass allocated to the stem: Substituting (4.103) into (4.104):

$$
M_{stem} = K\left[C_{pipe}\left(\frac{M_{canopy}}{SLA}\right)\right]^{3/2} \qquad (4.105)
$$

Taking the derivative of both sides:

$$
\frac{dM_{stem}}{dM_{canopy}} = 1.5K\left(\frac{C_{pipe}}{SLA}\right)^{3/2} M_{canopy}^{\;1/2} \qquad (4.106)
$$

Ratio of stem to canopy allocation

If we substitute $M_{aboveground}$ (M_{canopy}+M_{stem}) for M_{stem}, which is typically done for seedlings, we can use Eq. (4.106) directly to model the fraction of biomass allocated to the canopy. If not, additional math is required:

$$dM_{aboveground} = dM_{canopy}\left[1 + 1.5K\left(\frac{C_{pipe}}{SLA}\right)^{3/2} M_{canopy}\right] \qquad (4.107)$$

Allocation to canopy

The value for K as of yet is unspecified, and an expression is needed to predict the height. To solve for both these quantities, the second allometric relationship (between M and D^2H) is invoked:

$$M_{stem} = JD^2H \qquad (4.108)$$

Allometric relationship between diameter and height. J is wood density multiplied by various correction factors accounting for tapering (dD/dh) and branching (total wood mass/trunk mass).

Equating equations (4.108) & (4.105), we can solve for K:

$$K = J * \left(\frac{D}{H}\right)\left(\frac{\pi^{1.5}}{8}\right) \qquad (4.109)$$

Expression for K

If we assume K to be constant, which is reasonable for small ranges of H (see Carlson and Miller 1990), then D is simply a linear function of H.

Using eqs (4.100) to (4.109), the question of biomass partitioning is answered; the carbon balance is in closed form; and biomass can be expressed as the more useful quantities of height and diameter.

Plant Water Status Balance

$$C_{PLANT} \frac{d\Psi_{plant}}{dt} = \frac{\left(\psi_{soil} - \psi_{plant}\right)}{R_{root} A_{root}} - E \tag{4.110}$$

Plant water status is an important physical quantity, and is measured in terms of pressure potential. Water status has been known to affect the rates of growth, respiration, photosynthesis (J_{max} and R_{max}), see H. Jones, 1991, Ch. 10, and Anekonda and Adams 2000. Furthermore, stomata have been shown to respond to water status in a feedback manner, H. Jones Ch. 6.

In the GUESS model, the assumption is made that the soil is well watered, so that physiological effects of water status should be minimal except at the stomatal level. The additional assumption is also made that the feedback response to water status is smaller than the feedforward signals from the roots in response to soil moisture. Thus in GUESS, the plant water status balance is ignored. Since water status is of value, in determining irrigation control performance, users may elect to add a water status balance in future versions.

Stomatal Conductance and CO_2 Balance

The rate of photosynthesis in the Farquhar model is dependent upon the internal CO_2 concentration. To determine CO_2 concentration, a mass balance is performed at the leaf.

$$C_{leaf}\frac{d[CO_2]_i}{dt}=\left(\frac{1}{g_{stomatal}}+\frac{1}{g_{aerodynamic}}\right)^{-1}*([CO_2]_e-[CO_2]_i)-P_{net} \qquad (4.111)$$

Substomatal CO_2 balance. $[CO_2]_i$, $[CO_2]_e$ are internal and external CO_2 conc. C_{leaf} is the leaf capacitance in units of mol•m^2. $g_{stomatal}$ is the stomatal conductance in units of mol-s^{-1}-m^{-2}.

The primary means of control a plant has over transpiration and internal CO_2 concentration is through the adjusting the aperture of the stomata. To explain stomatal behavior, plant physiologists have applied the methods of control theory. See chapter 6 of H. Jones, 1991. Stomatal control models come in many forms, but the most widely use form is the Jarvis (1976) form:

$$g_{stomatal}=g_{closed}+g_{open}f_1(\alpha_1)f_2(\alpha_2)... \qquad (4.112)$$

Jarvis model of stomatal conductance. G_{open} is the stomatal conductance when stomata are fully open. G_{closed} is the stomatal conductance(mainly through the cuticle) when stomata are fully closed. $F_1(\alpha_1)$, $F_2(\alpha_2)$, are control signals with values between 1 and 0 that represent stomatal response to a change in light, humidity, soil water, temperature, etc…

To maintain consistent levels of internal water potential and CO_2 concentration, stomata have been found to employ control strategies resembling those used by feedback or feedforward controllers.

In a feedback control loop, the controller responds to disturbance from the setpoint or error by increasing or decreasing the flow until a balance is achieved. The error value (proportional), its derivative, or integral over time have been used to determine controller responses. The term feedback derives from the fact that the current system state can only be determined from its previous state. An oven that uses only the oven temperature to control gas flow to the burner is a feedback control system. The ventilation/heater logic used in GUESS, where deviations about successively higher setpoints activate progressively larger stages of fans/heaters is also an example of feedback control. An example of stomatal feedback control is the response to declining leaf water content:

Declining water potential in the leaf causes water to diffuse out of the stomata, causing closure, transpiration ceases, water flows in from the roots, turgor increases and stomata reopen.

In feedforward control, outside information and a model of the system are used predict the future state of the system. Once this state is known, an optimal control path to restore equilibrium can be found. In the greenhouse, a feedforward controller may use a radiometer to determine the solar gain, and indoor and outdoor temperature sensors to determine ventilation rate required to remove the excess solar load. If no suitable ventilation rate can be found, then evaporative cooling can be activated.

In the case of stomata, an example of feedforward control of stomata would be using outside humidity and CO_2 levels to determine a stomatal conductance that leads to optimal water use efficiency (ratio of photosynthesis to transpiration). Although there is much debate over stomatal control: what biological signals are stomata responding to, is it leaf water potential controlling stomatal aperture or is it stomatal aperture controlling leaf water potential, simple control models can still be applied and reasonable results can be obtained.

A simple, robust, and often used model of stomatal conductance is the Ball-Woodrow-Berry model (Ball, Berry and Woodrow 1986). It is a simple feedforward model where the goal is to optimize water use efficiency, mass of carbon fixed per mass of water transpired.

The Ball-Woodrow-Berry Model

$$g_{stomatal} = \min\left\{ g_{closed} + m\frac{rh_{leaf}P_{net}}{[CO_2]_{leaf}}, g_{open} \right\} \qquad (4.113)$$

Original model

In GUESS, a modified version of the Ball-Woodrow-Berry (1986) model is used which includes an additional term for soil water status. This extra term is mainly empirical in nature and represents the combined feedforward and feedback response to leaf water potential and drought induced abscisic acid.

$$g_{stomatal} = \min\left\{ g_{closed} + m\left(\frac{rh_{leaf}P_{net}}{[CO_2]_{leaf}}\right) * \left(\frac{\theta_{soil} - \theta_{WP}}{\theta_{FC} - \theta_{WP}}\right), g_{open} \right\} \qquad (4.114)$$

Modified Ball-Berry model used in GUESS.

Plant Energy Balances

Since the various organs of the plant are not at the same temperature, and since temperature response is approx. exponential, significant errors can result from using a single standard temperature. To account for this we let $T_{root} = T_{soil}$ and $T_{stem} = T_{leaf} = T_{canopy}$. In GUESS, the dubious assumption is made that $T_{soil} = T_{air}$, this is not entirely true due to the larger time constant for the soil compared to that of the air, and due to dampening in temperature fluctuations that occur in the soil. For canopy temperature, the steady state assumption is applied, and the following formula from Campbell, 1998 is used:

$$T_{leaf} = \frac{(1-\varepsilon_{cover})h_{r:sky}T_{sky} + \varepsilon_{cover}h_{r:cover}T_{cover} + g_{aerodynamic}T_{in} + \tau_{cover}I_{SW} + \lambda E}{(1-\varepsilon_{cover})h_{r:sky} + \varepsilon_{cover}h_{r:cover} + g_{aerodynamic}} \quad (4.115)$$

Equation for Leaf Temperature

Because of 1) the similar shortwave absorptivities and reflectivities for coniferous foliage and wet dark soil (Campbell, 1998, H. Jones 1991) and 2) similar surface humidity levels, he leaf canopy and soil surface can be treated as the same surface medium, which greatly simplifies the use of eq. (4.115).

Cost Computations

One reason for energy modeling is to estimate daily operating costs and see where improvements can be made. To this end a simple cost calculator was included in GUESS. Costs were determined for heating (gas), lighting and fans (electricity), and irrigation(water). Unit costs are in $/MBTU, cents/kwhr and cents/gallon and are assumed to be constant throughout the range of consumption. Energy costs estimates are assumed to be approximate, as only the major loads are included, which according to Albright (personal communication) are: space heating, ventilation, and lighting). Pump loads and other equipment loads are omitted from the cost model. Since GUESS is a model of energy consumption, rather than a full economic model, labor costs have also been omitted.

Chapter 5

MODEL ORGANIZATION

Introduction

While a complete list of equations may show the relationships between quantities, it provides no indication of how these equations are to be solved numerically on the computer, let alone how they are to be expressed and organized as part of the overall model software. Esoteric mathematical equations must be translated into computer code, which upon compilation and execution translates raw input data into meaningful output. Nothing is said about the different pre & post processing steps which must taken to go from raw input data to meaningful output graphs. This is the often neglected software engineering element of modeling.

According to Jones (1998) one of the major deficiencies of crop models in the past was in this crucial software engineering stage. While developed by brilliant researchers in the fields of ecology and soil science, the code was often written in a manner that left much to be desired. "Spaghetti" style organization and cryptic variables names made maintenance and modification difficult particularly by those unfamiliar with the model's development.

The GUESS model was developed using a form of model organization suggested by Gijzen et al, 1998. The GUESS model form is based upon the state machine concept, where each individual process is modeled as an individual module of code. The state machine concept was offered by Gijzen et al (1998) as an alternative to the "traditional" mean of model development and is identical to the form of model organization found in Simulink. Instead of machines or modules, in Simulink, the basic unit is the block.

In a block diagram, each machine or block is described by three sets of variables: the inputs, the state variables which describe the condition of the machine, and the output which depend directly upon the state. At each time step, the machine or block can be called upon perform to following commands:

1. Initialize/reset outputs and states
2. Calculate state derivatives
3. Integrate state derivatives to calculate future state
4. Calculate outputs based upon current state.

This robust yet simple methodology can be applied to a wide array of processes particularly those involving lumped parameter mass/energy balances or transfer functions, and is thus quite amenable to crop models, which are based on "black box" theory. The main impetus for using Simulink is that this methodology is built into the structure of the program, allowing the user to focus on the block diagram side of the model.

Source Code Organization

The GUESS source code is composed of the following programs:

1. guessinit.m: user-tunable model parameters.
2. guessread.m: weather data file processor and lookup table generator.
3. initplant.m: photosynthesis and growth lookup table generator
4. guesssim.mdl: Simulink model of GUESS processes
5. guessmodel.m: shell for guesssim.mdl
6. guessoutput.m: post-processing and output graphs
7. Subfunction Directories: 3 toolboxes of different helper functions
    a. Psychrometrics
    b. Solar Radiation
    c. Unit Conversions
    d. Plant Growth Processes

Each successive execution of the GUESS simulation involves the following steps:

1. Initialization with guessinit.m
2. Data processing with guessread.m
3. Model simulation with guesssim.mdl
4. Post-processing with guessoutput.m

Data Processing

A word or two should also be said about the data processing used in GUESS (both pre & post). Pre-processing are the data processing steps used to convert a raw data set into a series of parameters that can be used to determine boundary & initial condition for a model. Likewise post-processing are the data processing steps used to convert a models results (state and outputs) into useful visual output (data files and graphs and statistical analyses). In GUESS, pre-processing is handled by guessread.m, and post-processing is handled by guessoutput.m.

The raw data for the GUESS model consists of a 4 column *.CSV which contains hourly measurements of the following weather parameters: global solar radiation, temperature, relative humidity and wind speed. The program, guessread.m uses these hourly values to determine the following derived parameters: wet bulb temperature, humidity ratio, wind pressure coefficient, humidity ratio at wet-bulb, clearness index, fraction of diffuse radiation, and sky temperature.

Next, the original and derived weather parameters are interpolated to smaller intervals of n minutes, where n is the size of the time step, using cubic Hermite splines. Hermite splines offer advantage over other interpolation techniques in that monotonicity is preserved, and the spurious oscillations often seen with other spline techniques are dampened (Fritsch 1980). Finally, the interpolated data is then imputed to the guesssim model as a series of column vectors.

Post-processing is handled by guessoutput.m. Post-processing in GUESS is a rather simplistic affair. Data from the Simulink is given in a per time step format. However, a per the hour or per the day is far more useful for the end user. "guessouput" converts the Simulink from a by-n minute format to a more useful daily or hourly format, which is then graphed using MATLAB plotting functions or exported to an Excel spreadsheet for further processing.

Chapter 6

SIMULATION AND MODEL VALIDATION

The GUESS model was run to simulate a years worth (Jan 1 to Dec 31) of production for a nursery greenhouse located in Corvallis, OR. The goal was determine the effects of supplemental lighting and CO_2 on tree growth and production cost. Based upon personal communications with different growers, it was hypothesized that supplemental lighting would prove profitable at low levels only. The effects of CO_2 enrichment are not widely known and are the subject of current research.

Setup & Parameterization

The GUESS greenhouse was parameterized for a 34m by 17m single span greenhouse of single pane glass construction. Height to ridge is 7.66m, and height to eave is 5.11m. Infiltration rate is 1.1 air changes per hour, and a U-Value of 5.76 W/m^2-K was used. The plant model was parameterized for Douglas-fir seedlings. Plants were started at 0.57 g dry weight, and harvested at 1.67 g dry weight; a new growing season was recorded at each harvest.

An hourly data set for the year 2000 (Jan. 1 to Dec. 31) from the Corvallis, OR Agrimet weather station was used (address: http://www.usbr.gov/pn/agrimet/). Heating, ventilation and fogging (where applicable) schedules are presented in the tables at the end of this section.

Parameters for the model were obtained from a variety of sources which are listed in the third column of the *Parameter Table* in Appendix A). To examine the simultaneous effects of CO_2 enrichment and supplemental lighting, the following test cases were run:

Lighting Parameters			
Intensity	Set point	Band-width	CO2 enrichment
No Lights	--	--	Yes
25	50	15	Yes
75	88	40	Yes
75	88	40	No
100	100	55	No
100	100	55	Yes
250	250	130	Yes

Table 2 List of lighting scenarios tested.

During each simulation run, the following data was recorded and used to compare scenarios:

- length of first growing seasons
- number of growing seasons
- the presence of any significant period of negative carbon balance (failure in the model, indicating the onset of dormancy)
- and year-end cost breakdown
    - total cost
    - electrical cost
    - fuel cost.

Utility costs are assumed to be: water: $0/1000 gal (water), 11.87¢/kWh (electricity) and $3.88/MBTU (fuel [natural gas]). Since a strong dependency of growth rate upon the μmoles $\cdot s^{-1}$ per W sunlight to PAR conversion factor was observed, two separate conversion factors were tested: 2.1 μmoles$\cdot s^{-1}$ per W, which was designed to reach 2000 μmoles$\cdot m^{-2} \cdot s^{-1}$ at full sun [950 W/m^2]) and 2.35 μmoles$\cdot s^{-1}$ per W, which was based upon data from Langhans 1990, assuming sunlight is 50% PAR. Due to problems with algebraic loops, parts of an otherwise dynamic model that require an iterative solution, differences with air and leaf temperature were ignored. An explicit heat balance for the canopy might work serve as a workaround to this problem. In this simple test, heaters, fans, and foggers were assumed to respond instantaneously to control signal, and dynamic effects like slewing, and delaying are ignored.

CONTROL SCHEDULES

Stage	Setpoint °C	Bandwidth ± °C	Heating Rate W/m^2 floor Area
1	18.3	1	60
2	15.6	2.5	120
3	12.8	2.5	200

Table 3 Heating Schedule

Stage	Setpoint °C	Bandwidth ± °C	Ventilation Rate (Air Changes/ Minute)
1	26.7	2	0.22
2	29.4	2	0.55
3	31	2	1.1

Table 4 Ventilation Schedule

Stage	Setpoint °C	Bandwidth ± °C	Fogging Rate (L/min)	Conductance
1	23.9	2	3.9	49.7
2	26.7	2	9.75	124.3
3	29.44	2	19.5	248.6

Table 5 Fogging Schedule

Chapter 7

RESULTS OF SIMULATION

The GUESS model was encoded using the full Windows version of MATLAB 7.1 Release 14 with Simulink. The simulation was run on a SONY VAIO laptop, model № VGN-FS660. The laptop was equipped with a 1.73 GHz Intel Pentium® Centrino™ processor, 70GB hard drive, and 1GB of RAM. The Simulink portions of the model were run in "Accelerator" mode: which first generates compact C code representation of the block diagram, and then compiles and executes the resulting C program. Execution of the simulation took approximately 150 seconds. The results of the simulation are presented on the next page.

Lighting Parameters					Growth			System Cost		
Intensity	Sun: PAR Conversion Factor	Set point	Band-width	CO2 enrich-ment	# of growing seasons	Length of first season	Negative growth in last season	Total	Fuel	Electric
No Lights	2.1			Yes	2	142	Yes	$ 3,448.93	$3,148.37	$ 300.55
No Lights	2.35			No	3	141	Unclear	$ 3,448.93	$3,148.37	$ 300.55
75	2.1	88	40	Yes	3	136	No	$ 7,373.77	$2,871.37	$4,502.40
75	2.1	88	40	No	2	136	Yes	$ 7,373.77	$2,871.37	$4,502.40
75	2.35	88	40	No	3	135	No	$ 7,373.77	$2,871.37	$4,502.40
100	2.35	100	55	Yes	3	126	No	$ 8,774.14	$2,797.07	$5,977.06
200	2.35	250	130	Yes	4	101	No	$21,957.59	$2,472.23	$19,485.35

Table 6 Comparison of different lighting schedules. Ventilation & heating were achieved as per schedules outlined in previous section. Photoperiod was kept to a minimum of 12 hours for all light settings.

Evaporative Cooling Comparison		
	Pads & Fans	Foggers & Nat. Vent
Mean	68.55 °F	68.87 °F
Std. Dev.	6.30 °F	6.79 °F
Elect. Cost	$ 300.00	$ 179.93
F-Statistic	**3.267E-315**	

Table 7 Comparison of two different evaporative cooling methods. No supplemental lighting was provided. In the first run, pads were enabled, and fans were staged according to the schedule presented before. In the second run, pads were disabled, foggers were enabled and the first fan stage was replaced with natural ventilation. Statistically, definitively better control was achieved using pads & fans. However, the difference was slight and outweighed by the cost savings with foggers. F-Statistics were calculated using vartest2.m.

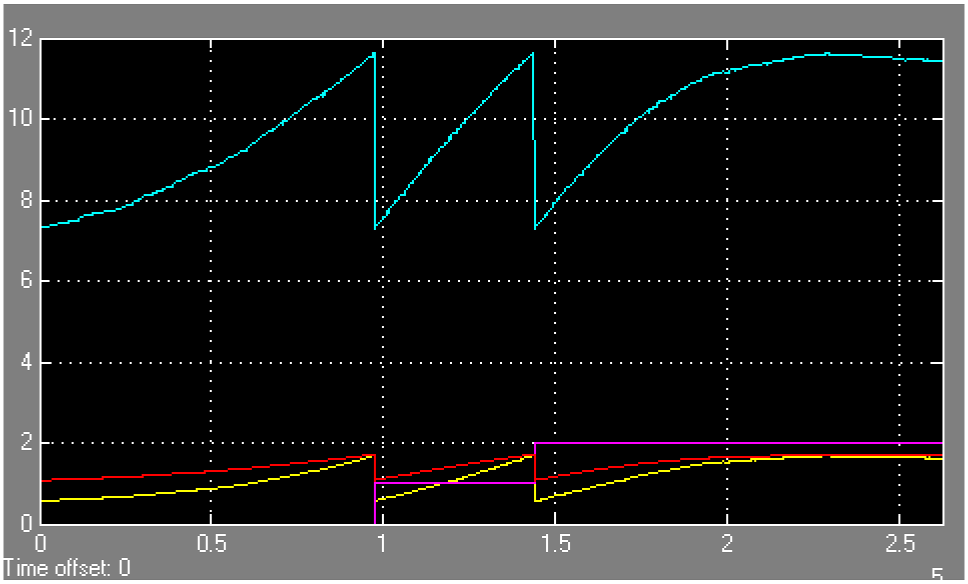


Figure 9 Simulink scope output for GUESS model. Blue line is height in cm, purple line is the cumulative number of growing season, red line is total biomass (dry weight), and yellow line is stem diameter in mm. Negative growth is caused by insufficient light integral during the winter and signifies a breakdown in the model. In reality, dormancy would be induced by declining light levels and photoperiod. Data taken from the no supplemental lighting trial with 2.1μmol/W.

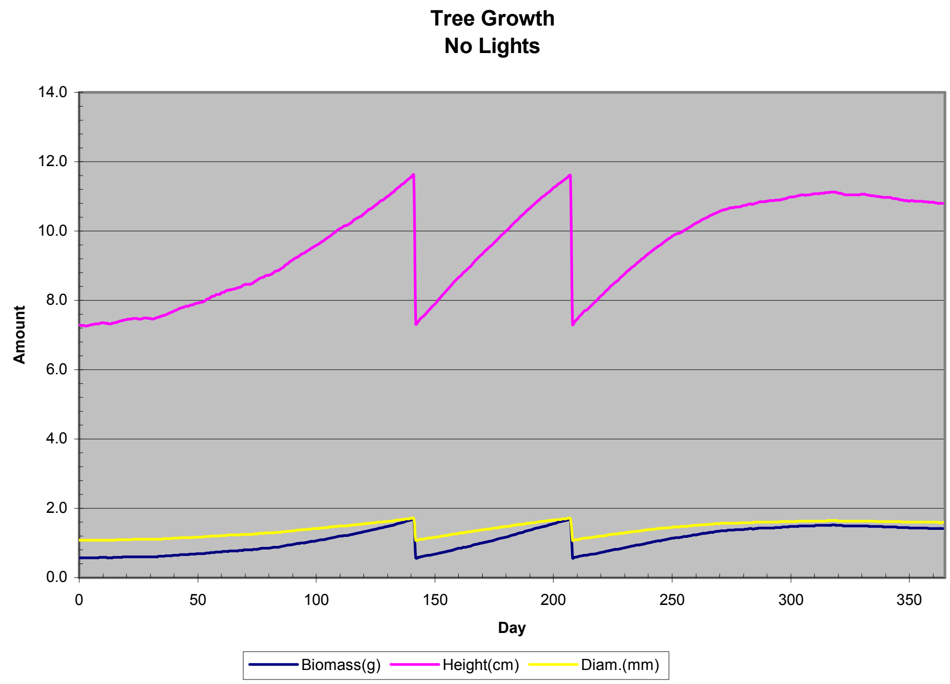


Figure 10 Growth Chart for CO_2 enrichment without supplemental lighting using 2.1 μmol/W.

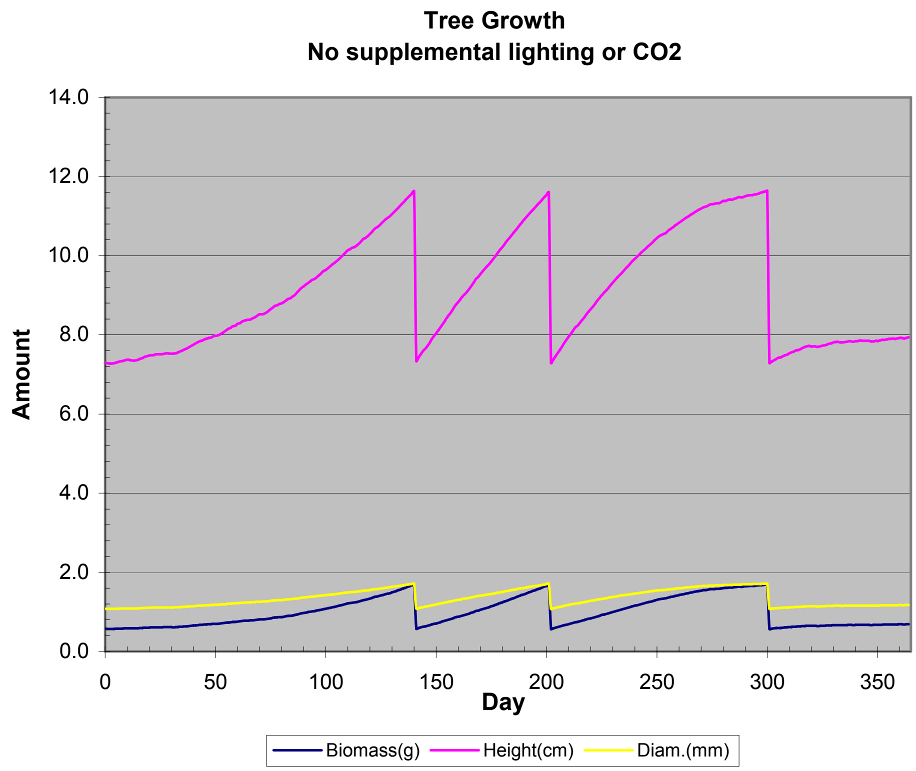


Figure 11 No supplemental lighting or CO_2 scenario with 2.3 µmoles PAR/W sunlight conversion factor

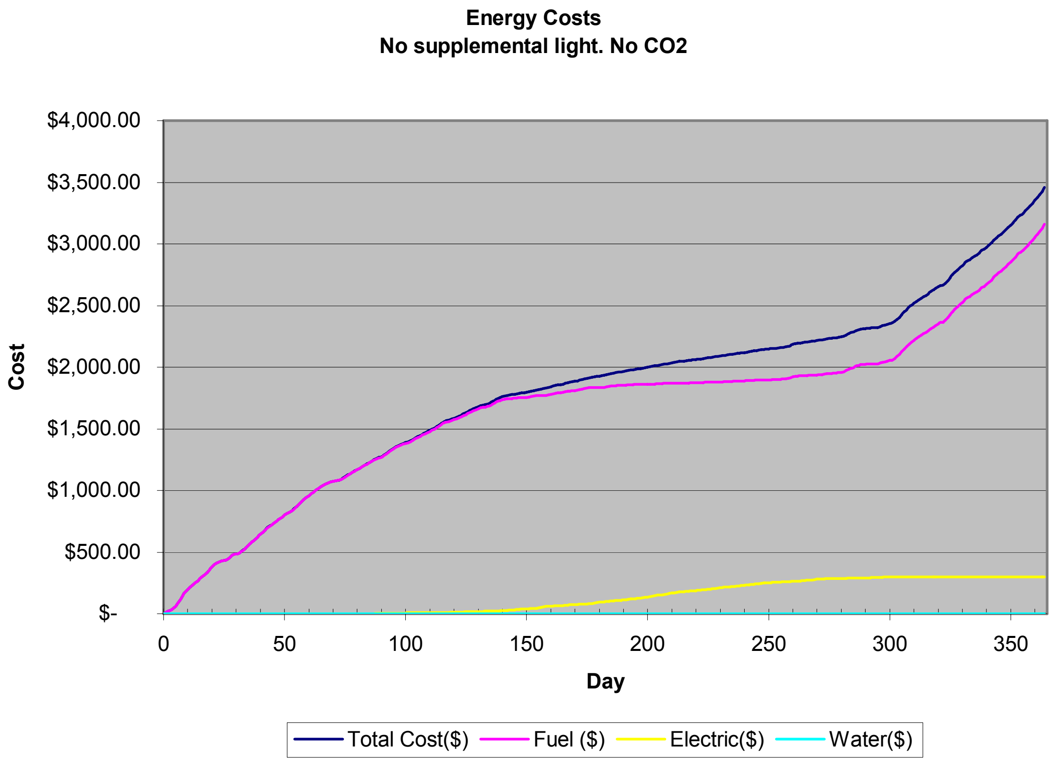


Figure 12 Energy costs for no lighting scenario

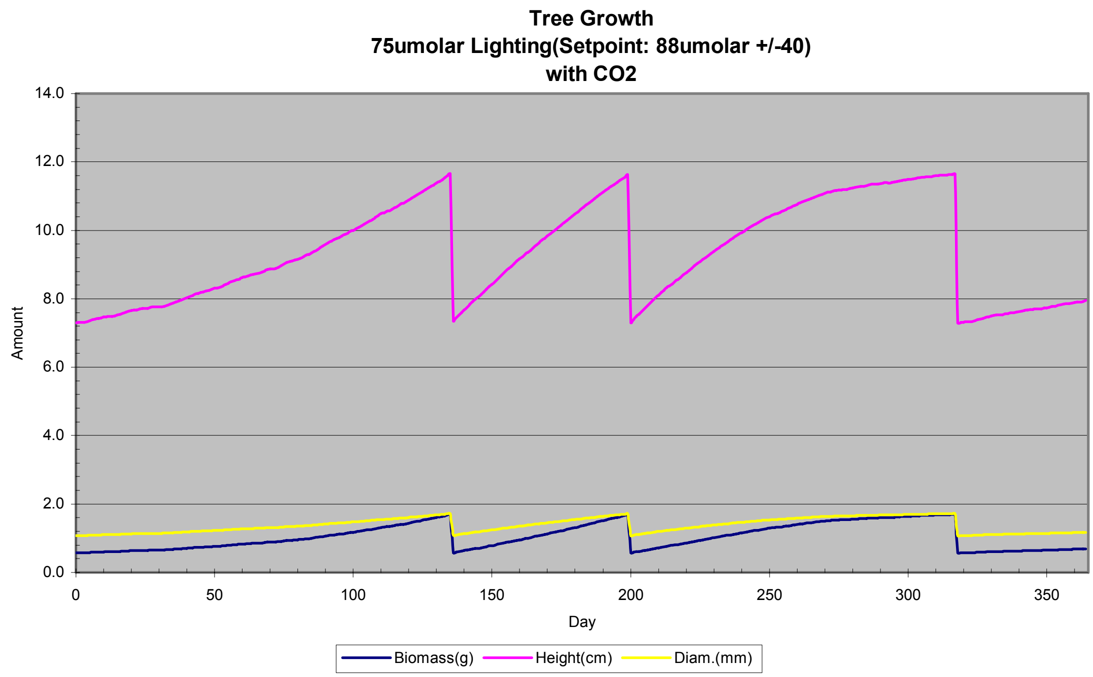


Figure 13 Growth chart for 75μmolar supplemental lighting with CO_2 enrichment and 2.1 μmol/W conversion factor.

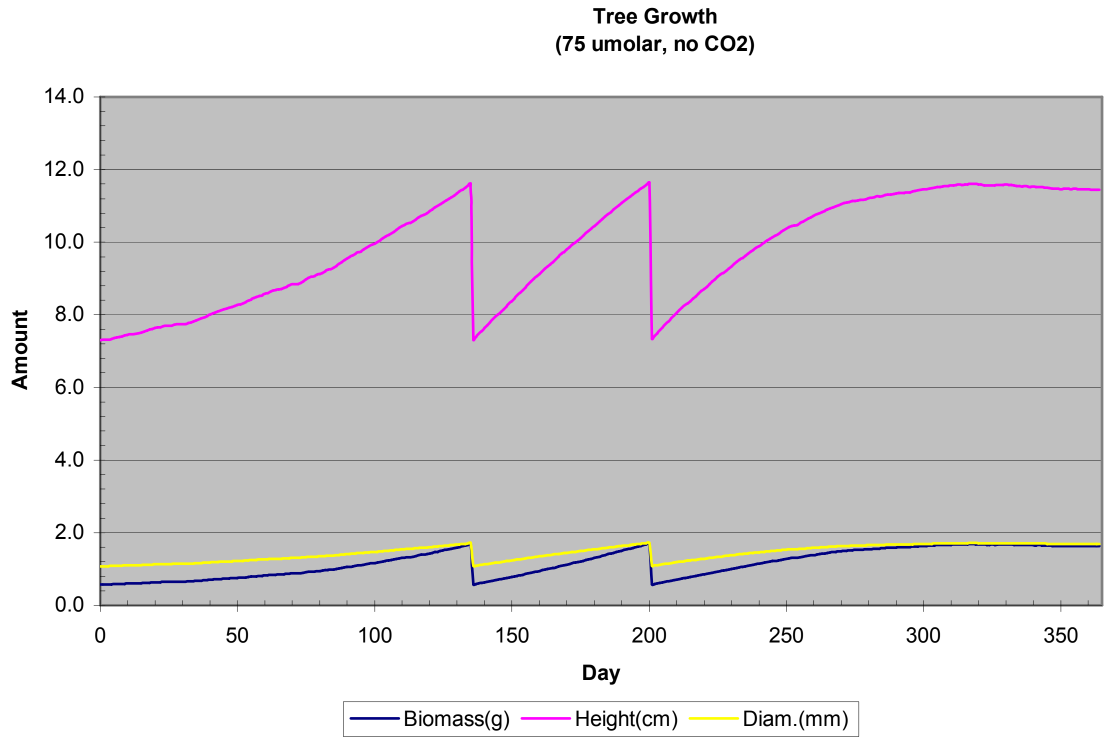


Figure 14 Growth chart for 75 μmolar supplemental lighting without CO_2 enrichment and 2.1 μmol/W. Notice negative growth rate in third season

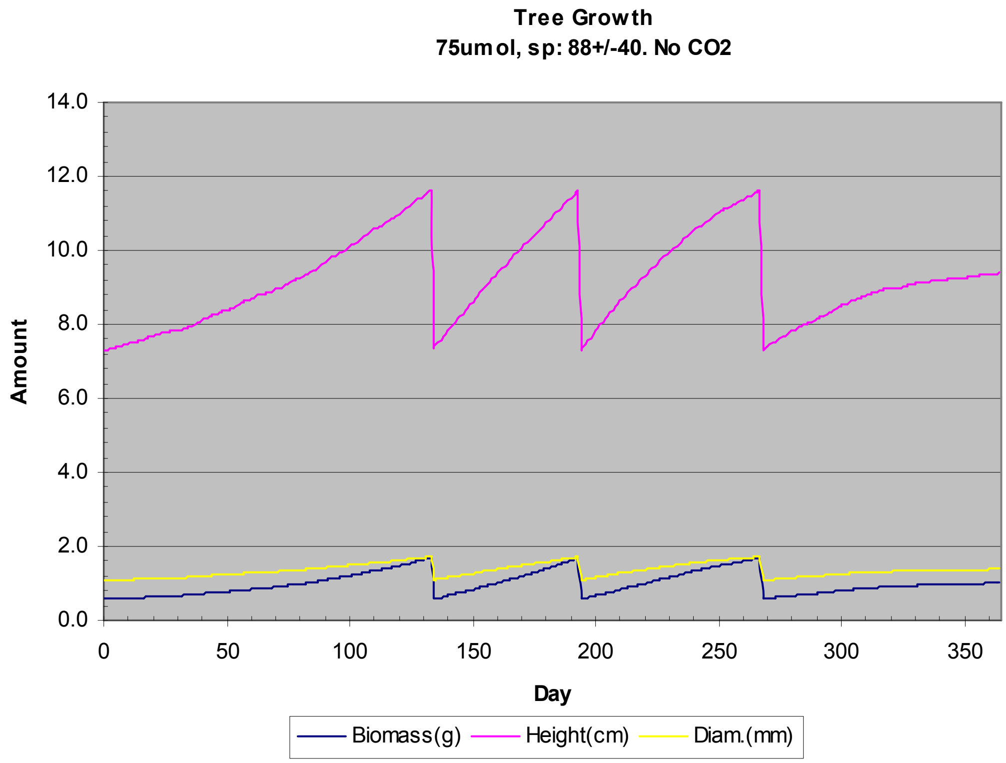


Figure 15 Growth rate for 75μmolar lighting without CO_2 enrichment and 2.35 μmol/W conversion factor

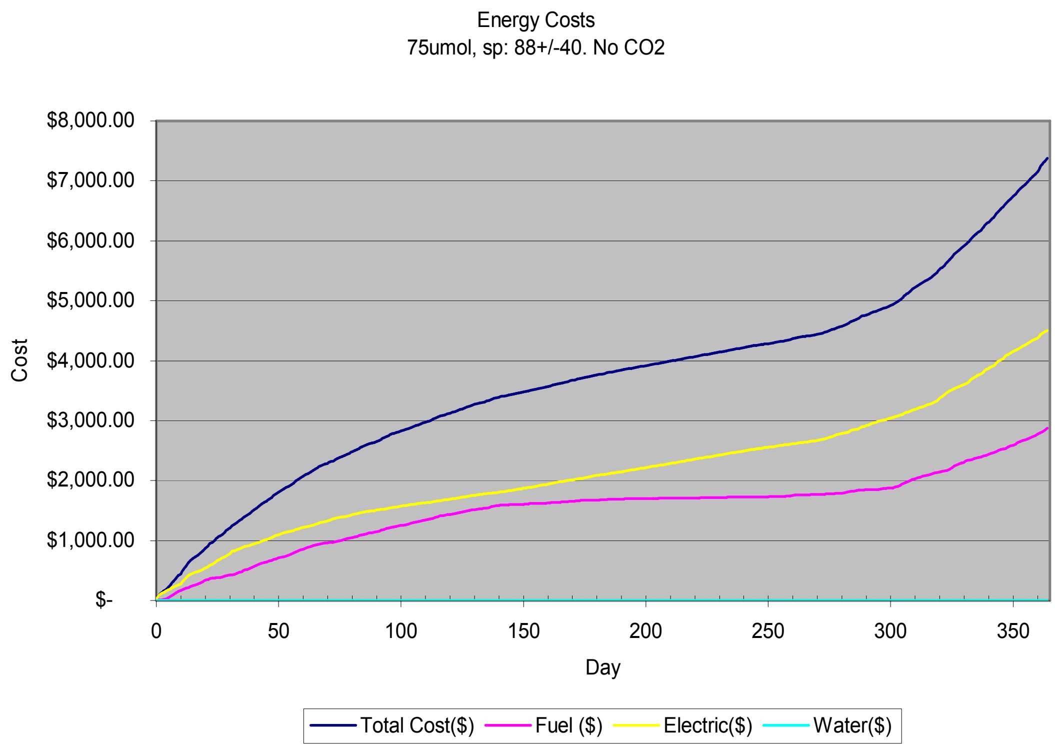


Figure 16 Energy cost chart for 75 μmolar supplemental lighting.

Figure 17 Growth chart for 100 μmolar supplemental lighting

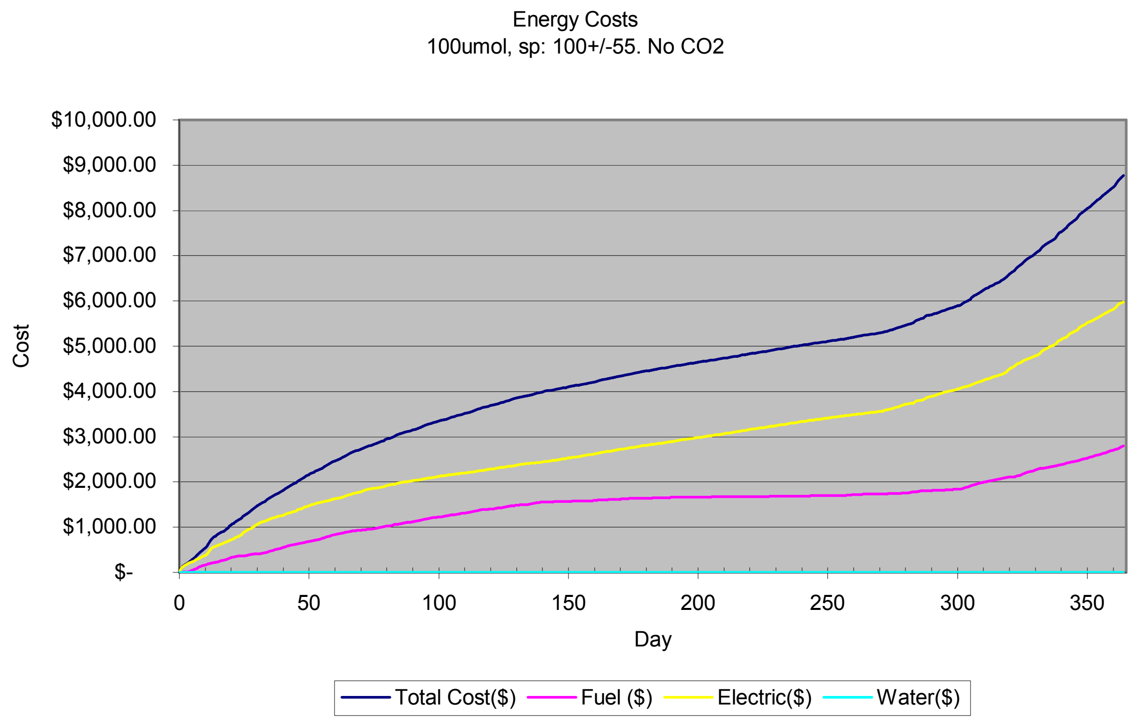


Figure 18 Energy Cost Diagram for 100 µmolar supplemental lighting with CO_2 enrichment

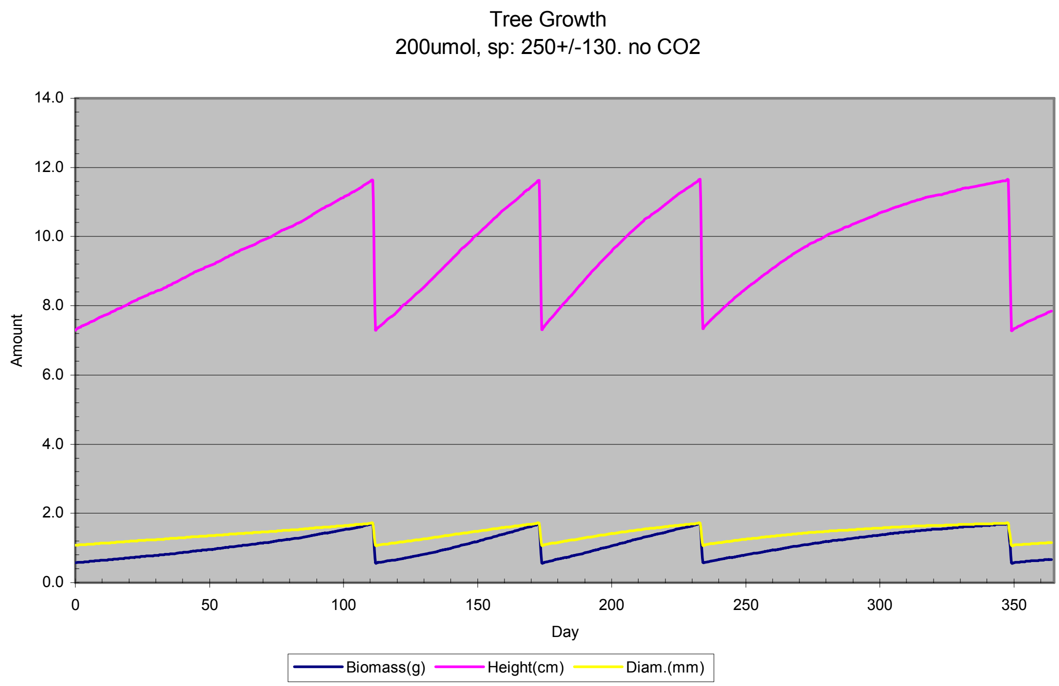


Figure 19 Growth chart for 250 μmolar supplemental lighting, without CO_2 & 2.35 conversion factor

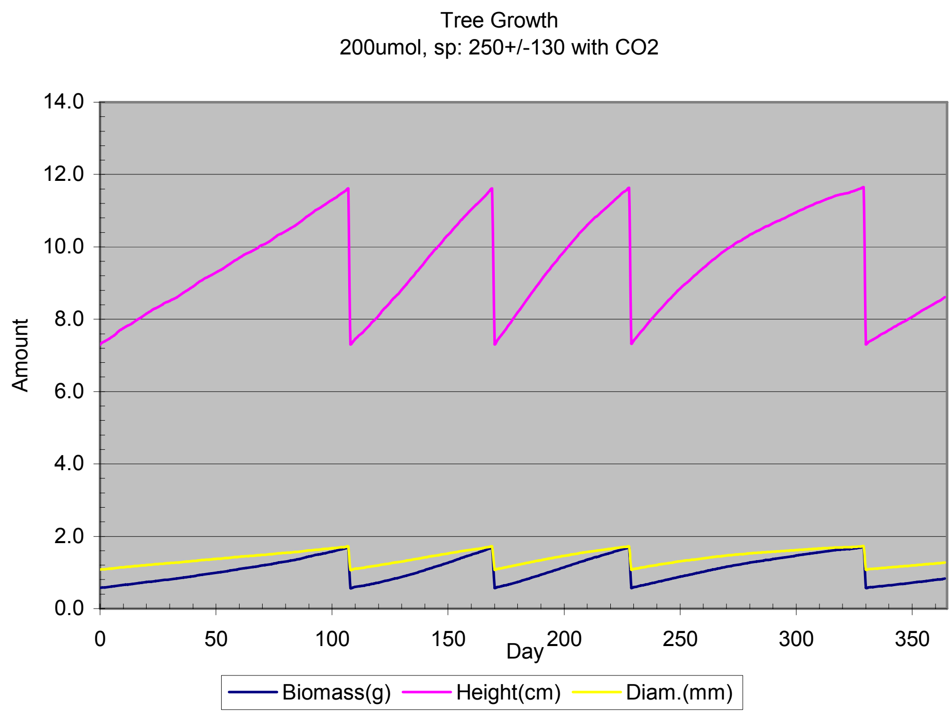


Figure 20 Growth chart with 250 μmolar supplemental lighting and CO_2. (2.35 conversion factor)

Luxuriant growth …

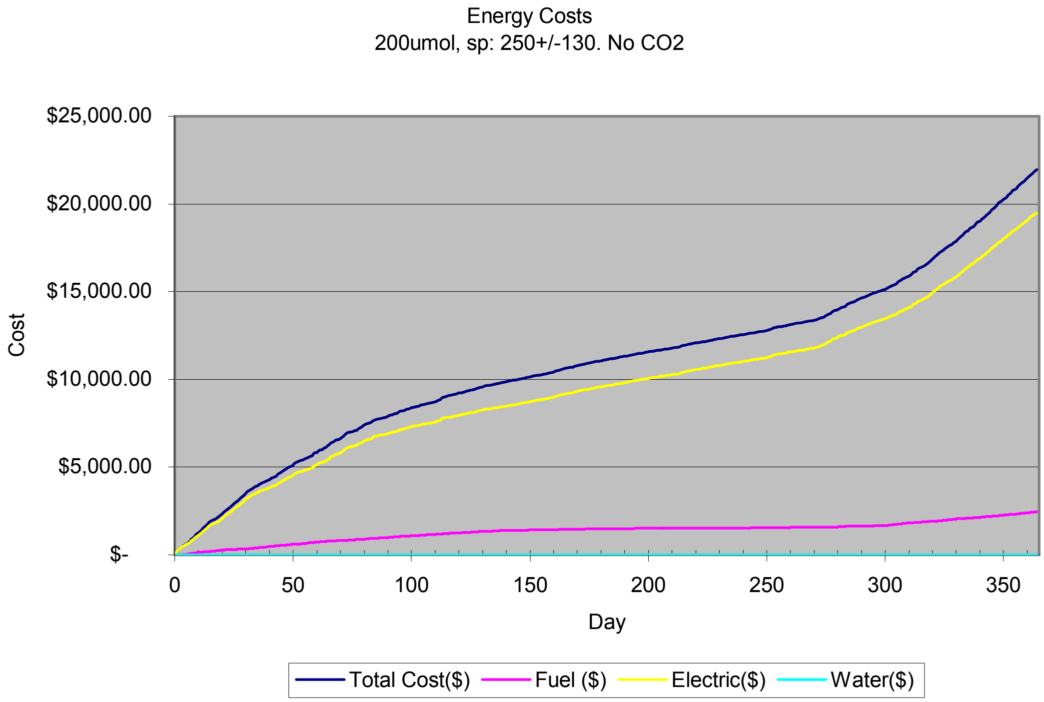


Figure 21 At an exorbitant price

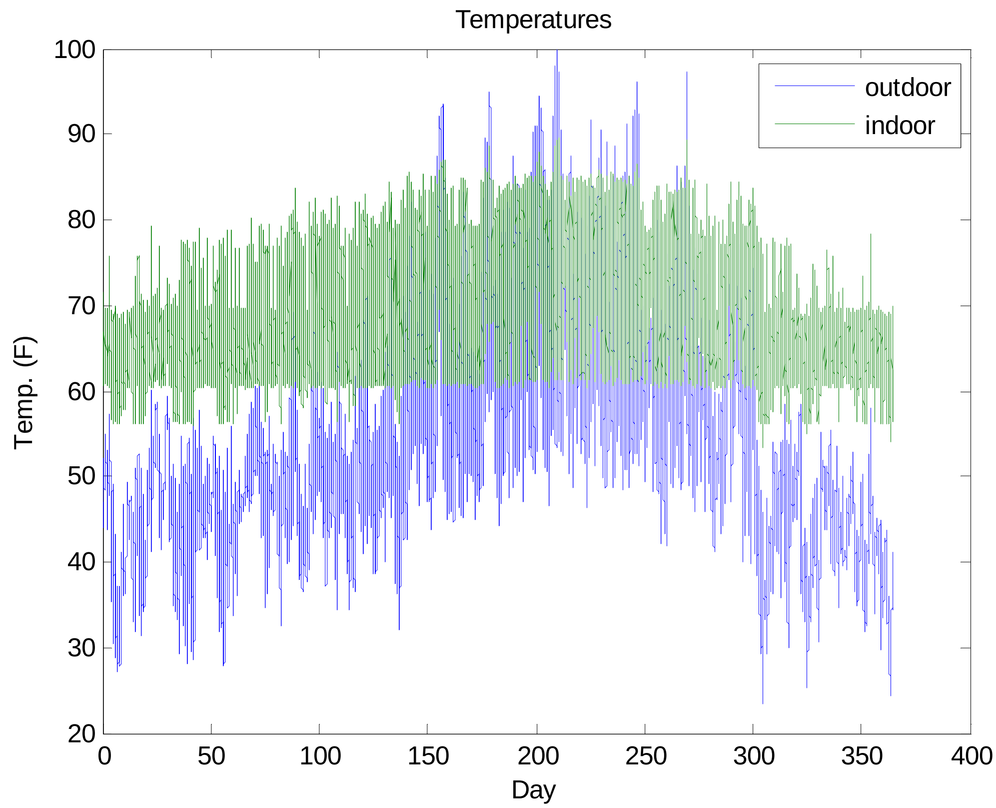


Figure 22 Indoor/Outdoor temperature trace of temperature, with foggers and natural ventilation enabled. Notice hard floors at 60°F and 55°F due to the constant heat flux from the heaters, and notice as well soft ceiling due to variable outdoor temperature and relative humidity.

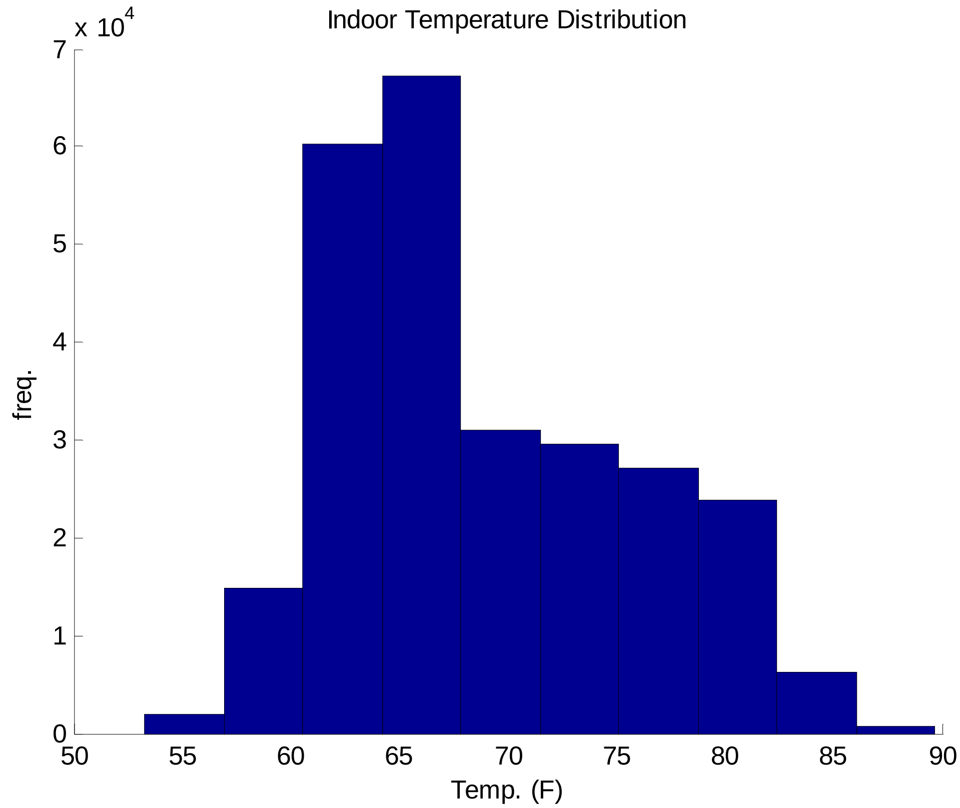


Figure 23 Temperature Histogram for foggers/natural ventilation

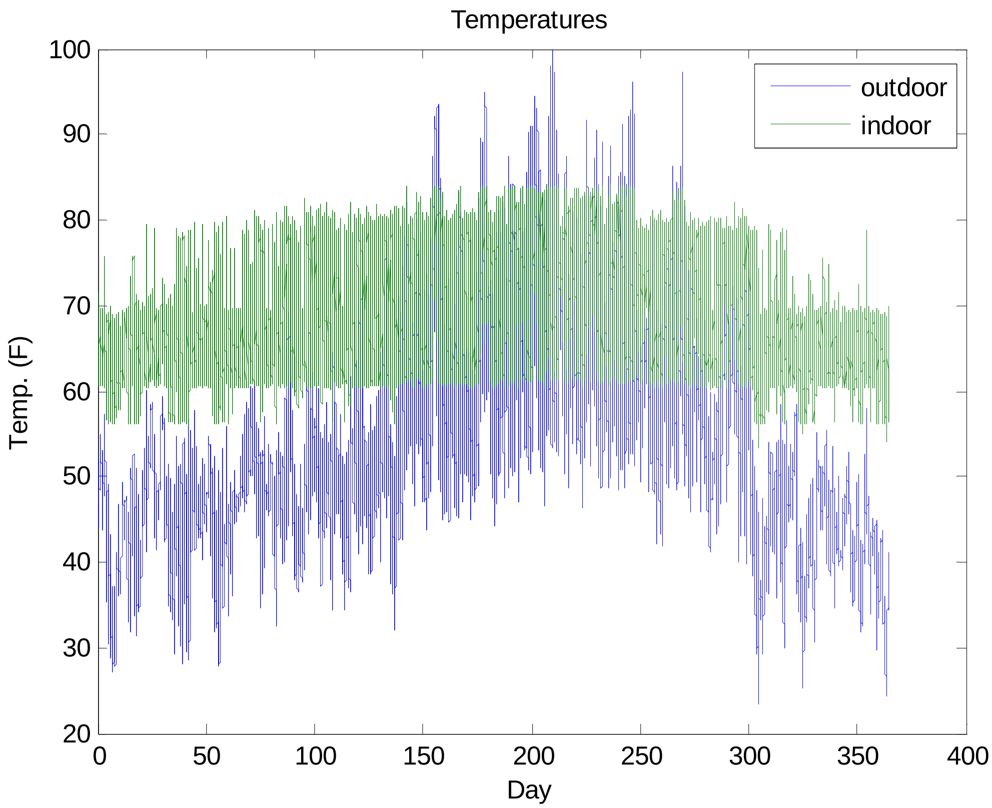


Figure 24 Temperature Trace for greenhouse with pads enabled, and natural ventilation and foggers disabled.

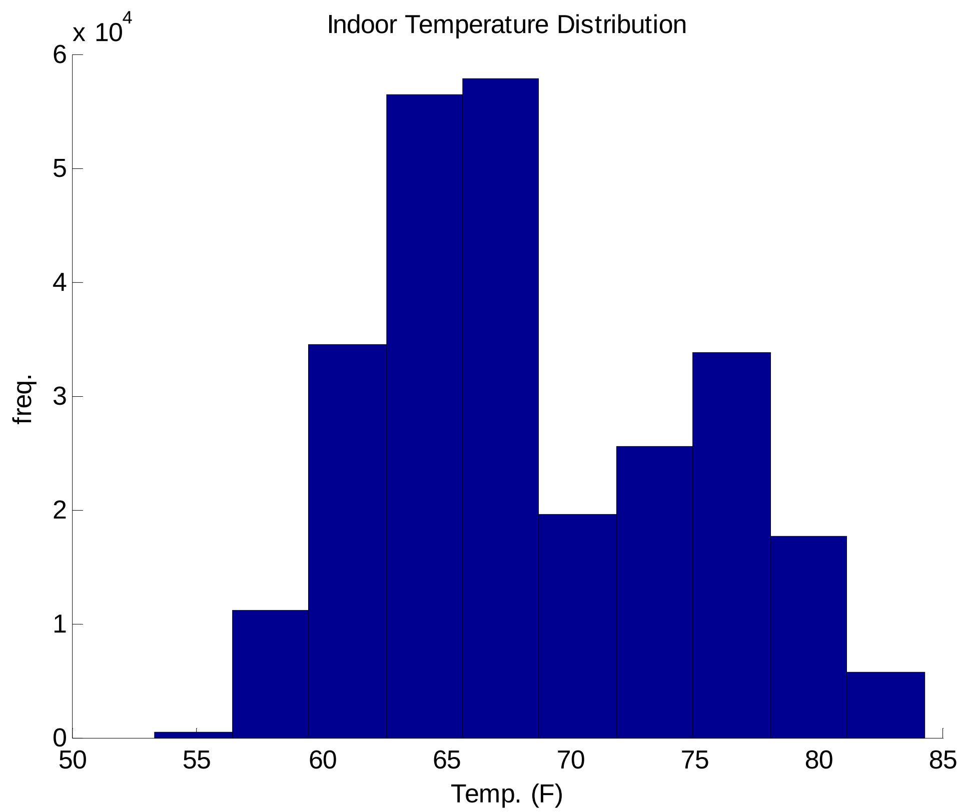


Figure 25 Temperature histogram for mechanical ventilation and cooling pads.

Chapter 8

DISCUSSION

Based upon conversations with growers (Mehlschau, Reagan, and Landis), it was hypothesized that supplemental lighting while necessary to prevent dormancy is uneconomical in quantities large enough to make practical impacts in production. To test this hypothesis, whole production years were simulated at the different lighting schedules indicated in table 5.

Assuming that GUESS accurately models the growth of greenhouse Douglas-fir seedlings, then given the relative relatively constant temperatures, ample supply of water and nutrients, and high daytime humidities inside the glasshouse; the limiting factor for the production of Douglas-fir in Corvallis, OR should be light. This conclusion is supported by the relatively slow growth rates observed during the first and last crop seasons, and by graphing daily growth rates obtained from GUESS in mg dry matter against daily light integral in moles photons.
Furthermore, a strong dependency of growth rate upon sunlight PAR content should be observed. Indeed that was the case.

Upon changing the original 2.1 μmoles$\cdot s^{-1}$ per W conversion factor to the new 2.35 μmoles$\cdot s^{-1}$ per W value a significant increase in winter growth was observed. Number of growing seasons per year increased from 2 to 3. According to Mehlschau, three growing seasons per year is typical.

For such a sensitive parameter, scant information is available about the molar PAR content of full spectrum sunlight. As with many lighting sources, information is provided only for the visible band, requiring the construction of assumptions about the size of the PAR and NIR wavebands in terms of energy content. The split between NIR (near-infrared) and PAR is highly dependent on local conditions. Values for molar PAR content vary from 2.1 μmoles·s^{-1} per W(California) to 2.9 μmoles·s^{-1} per W(Texas); see Monteith and Unsworth, 1990.

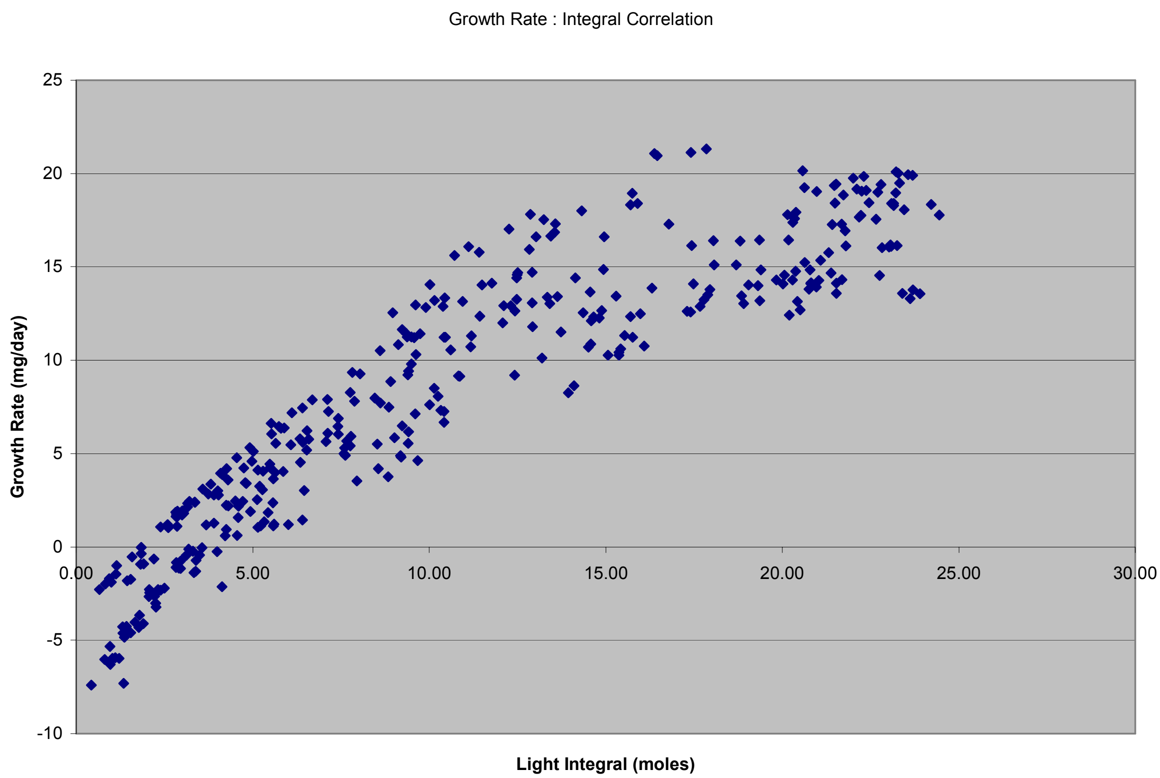


Figure 26 Daily growth rate vs. light integral. The data was taken from the no light simulation run.

Although there is some scatter, the trend of linear growth up to 15 moles per day, and saturation thereafter can easily be observed. It is also apparent from the graph that growth rate can become negative when light integral becomes too low, and this term manifests itself in the negative growth rates observed in the 3rd growing season for supplemental lighting levels less that 75 μmolar. Much of the scatter is due to leaf area index and biomass effects.

Based upon the results presented in table 6, the hypothesis that lighting is uneconomical in large quantities is a correct one. If the goals of lighting are to not only extend photoperiod, but also to extend the growth period into the night, and maintain a consistent light integral during cloudy days, then the cost of lighting will increase non-linearly with the amount light to be produced. Although seedling prices could not be obtained at the time of writing, it is safe to conclude based upon grower interviews that levels of lighting sufficient to achieve notable increases in the number of growing season and shortening in length thereof, (roughly 200+μmolar) are too expensive for most growers to use. Also, the need for lighting depends strongly upon the quantum content of sunlight esp. for dim and cloudy days, a simple increase in PAR content from 2.1 to 2.35 meant the difference between 2 growing seasons and 3 growing seasons under no supplemental lighting conditions.

As of yet, no known commercial greenhouses are implementing CO_2 enrichment for Douglas-fir production, so it would interesting to see if enrichment is a cost-effective means of enhancing production. Using the 2.1 μEinstein per $W \cdot m^{-2}$ conversion factor, uncontrolled enrichment by flue gas recirculation results a gain in production of 1 additional growing season, and a reduction in electricity costs of $1700 are obtained, since 75μmolar of lighting is required instead of 100 μmolar for the no enrichment scenario. With the 2.35 μEinstein per $W \cdot m^{-2}$ conversion factor, notable production gains occur in the last season, however they are not as dramatic.

Another way to save money while conserving energy is to replace the cooling pads with a fogger system, and to replace the first stage of ventilation with natural wind-induced ventilation. It is useful to see if the same level of temperature control can be maintained using natural ventilation. The 75 μmolar with enrichment case was rerun twice: once with stage one replaced by natural ventilation and foggers, the other with fans for all stages and pads. Foggers and natural ventilation performed slightly worse than fans and pads, with an average temperature variation of 6.15°F versus 5.84°F. While this difference was found to be statistically significant, for practical purposes, it is of no consequence, so natural ventilation is a viable option for cost reduction and energy conservation.

A Simple Expression for Net Photosynthesis

Earlier it was mentioned that much of the scatter could be accounted for by correcting for LAI or biomass. For rough design calculations, a simple model obtained from the simulation results would be most useful.

As LAI increases, average irradiances per leaf area decreases, however since assimilation is expressed as a flux, overall growth may increase per plant. Respiration load increases linearly with biomass for most plant tissues. Since LAI, given a constant container size, can be related to biomass by allometric equations; dividing both growth rate and light integral by biomass should correct for these two separate effects.

If we apply the following saturation transform to the light integral equation, a linear relationship between assimilation and net photosynthesis results (Chen & Klinka 1997).

$$x = \frac{LI_{mass}}{18 + 2 * LI_{mass}} \qquad (8.1)$$

Saturation Transform

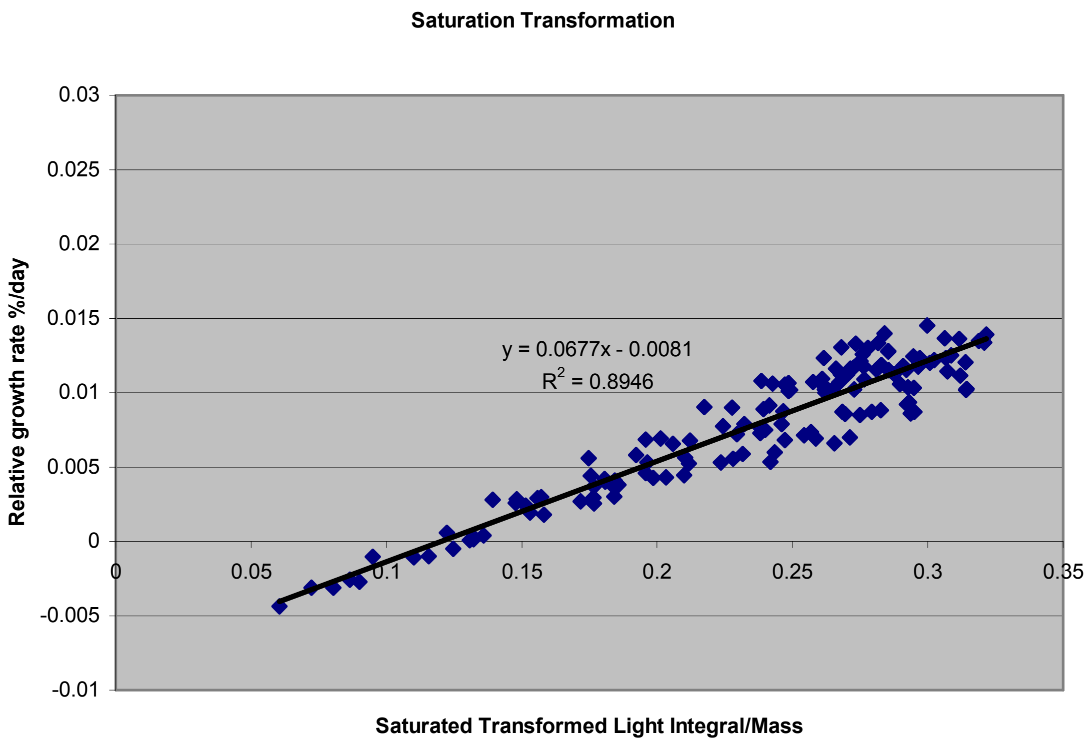


Figure 27 New Light Integral: Growth Rate Relationship

$$\frac{dM}{M} = .0677\left(\frac{LI/M}{2*LI/M+18}\right) - .0081 \tag{8.2}$$

Simplified Regression expression for Douglas Fir Growth Rate

Simple linear regressions, like the one above have the advantage of being able to be solved by hand as opposed to simulation. Although the regression cannot predict exactly the behavior of the plant under all conditions, for a given set of fixed parameters: CO_2 and temperature, the above model can be used for rough design calculations, or implementation in a model based control of light integral. It also serves to illustrate how complex simulations can be used to derive simpler single equation models.

Chapter 9

MODEL VALIDATION

A complete validation of the GUESS model would involve comparing crop yields, hourly indoor climatic data, and heating/lighting/cooling schedules of an actual greenhouse to the results of a GUESS simulation. Unfortunately time constraints prevented such a study from happening. However, the results of the GUESS model were checked for physical reasonableness. And a case study was conducted in Excel to test various assumptions in formulating the GUESS model.

Although a full validation could not have been performed at this time, the results of the GUESS were checked over to ensure there is as least qualitative agreement between the model predictions and regularly observed phenomena. Three commonly observed phenomena are an inverse relationship between indoor and outdoor relative humidities, a draw-down in CO_2 during the day and an increase at night; and hard floor and soft ceiling being imposed upon indoor temperature a 50-50 partitioning of incoming shortwave into latent and sensible heat.

Variations in rel. H

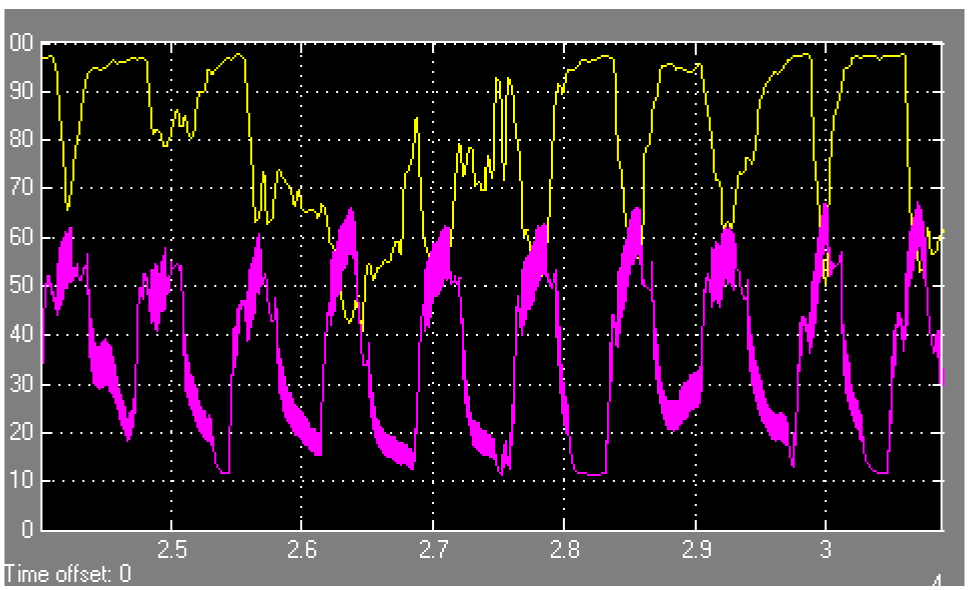


Figure 28 Relative humidity inside and outside greenhouse during the early winter. The yellow line is outdoor humidity, and the magenta line is indoor humidity

Greenhouse relative humidity is one half-cycle out of phase with outdoor humidity. This is to be expected. In the absence of significant weather events, outdoor humidity ratio (humidity ratio) remains constant and in temperate climates, almost equals saturation at night (personal communication, Riha). In a greenhouse the principal source of humidity is radiation driven evapotranspiration and evaporative cooling. As solar loading increases, evaporation will increase, the partitioning between latent heat and sensible being a function of temperature, see the Penman-Monteith equation (4.39). At high temperatures, the activation of evaporative cooling further acerbates this effect. Since the greenhouse envelope provides an effective barrier to mass transfer, relative humidity must continue, until condensation occurs on the inner surface of the glazing. The 60% relative humidity maximum could signify the onset of condensation. At nighttime, the air in the greenhouse is heated, and saturation vapor pressure rises exponentially with temperature. If radiation loads from lighting are small, then evapotranspiration should be negligible. Therefore, the only source of humidity should be from the outside air itself, and in short while, the atmosphere inside and out should equilibrate in terms of humidity ratios. So for a given humidity ratio, one should expect a significantly lower relative humidity inside in the greenhouse than out. The noise occurring in the indoor temperature signal is caused by the cyclical activation and deactivation of the heating and ventilation system.

Variations in CO_2 level

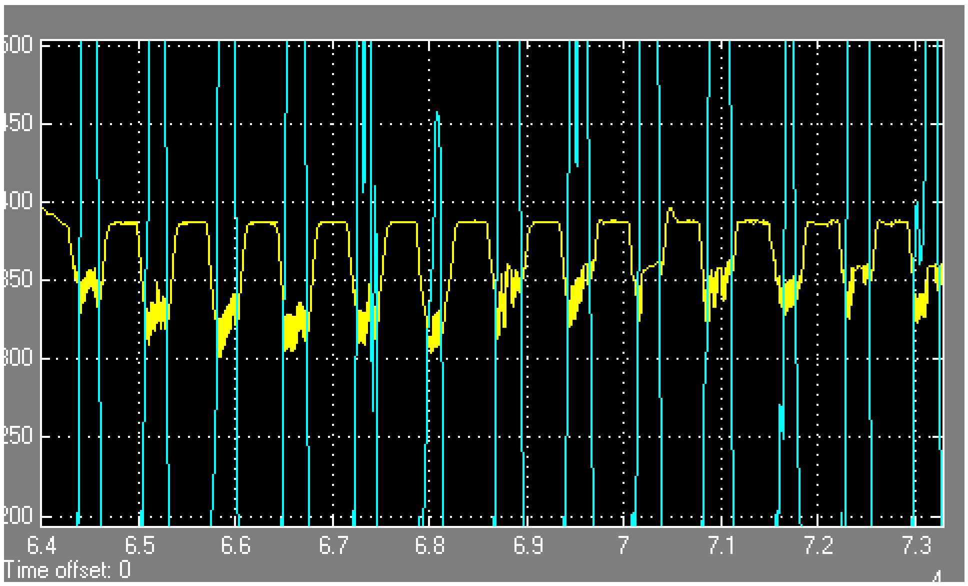


Figure 29 CO_2 levels inside the greenhouse without enrichment. Blue line indicates PAR load in μmol/m^2-s. Yellow line indicates CO_2 concentration(ppm) inside the greenhouse.

The previous graph is a plot of CO_2 concentration in ppm inside the greenhouse, and indicates the effects of photosynthesis and respiration upon CO_2 concentration in the absence of enrichment. These results were obtained from the 75μmol lighting trial. Ambient (outdoor) CO_2 levels are 370 ppm. Photosynthesis causes a depletion in CO_2 levels to as low as 300 ppm, as PAR levels increase. This is especially apparent during the winter or early morning hours as solar loading could be high, but lower outdoor temperatures and consequentially higher conduction losses prevent a temperature rise leading to the triggering of ventilation. At night, respiration and infiltration restore CO_2 levels to near ambient levels.

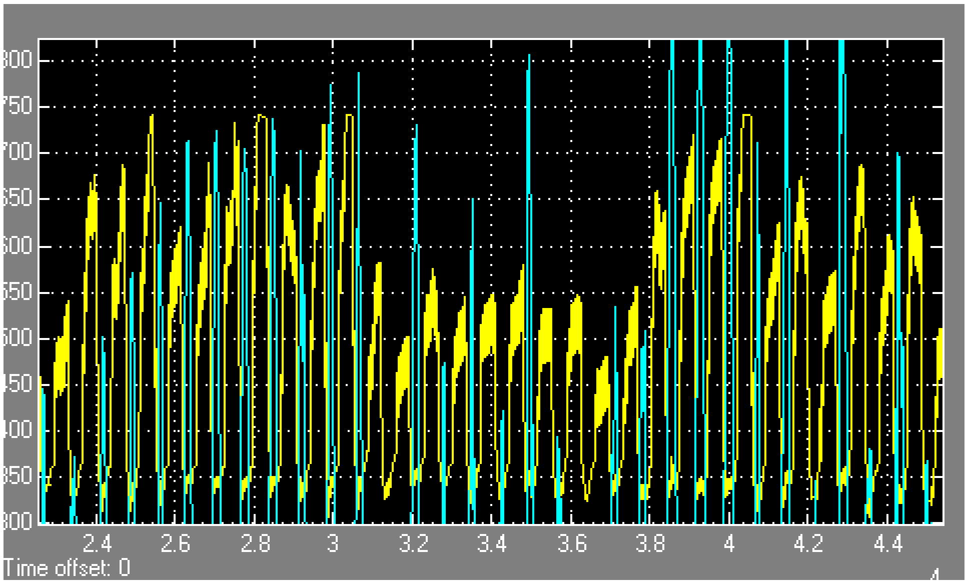


Figure 30 Indoor CO_2 concentration in a greenhouse with enrichment

In the above figure, we see another CO_2 concentration plot from the same greenhouse, the only difference being that CO_2 is being artificially supplied from flue gas recirculation. Return ratio is set at 0.333 (recirculated exhaust: total flue exhaust). Exhaust gas recirculation can provide a tremendous boost to indoor CO_2 levels. The problem is that the bulk of this gas is supplied at night, when its effects are minimal. The same shortwave energy which leads to increased rates of photosynthesis, also cause temperatures to rise, and heaters to turn off, and ventilators to turn on. If artificial lighting is used, nighttime enrichment becomes useful, since high CO2 levels lead to reduced photorespiration and more efficient usage of light energy, as illustrated by the Farquhar equations.

Variations in Temperature

The effects of the hard floor—soft ceiling on indoor temperature are evident in the temperature traces for both test cases: pads/fans and foggers/natural ventilation. The hard floor is caused by the constant heat flux from the heaters. Fluxes from heaters are independent of outdoor weather conditions. Assuming that the heaters produce enough energy to overcome nighttime conduction, then minimum temperature inside the greenhouse should be constant, hard floor. The fluxes from evaporative cooling and ventilation vary with temperature and relative humidity of both the indoor and outdoor airspace. Maximum temperature should not be a constant, and a soft ceiling is observed.

Case study

The case study is a simple model of envelope heat transfer created in Excel to test the following assumptions used in formulating the GUESS model.

1. Dynamic response of climate governed by air space
2. Constant cover conductance
3. Cover temperature is a constant linear function of indoor and outdoor temperatures

The case study also calculates time constants for the following surfaces based upon lumped parameter heat transfer theory:

1. Cover
2. Air Space
3. Soil
4. Canopy

In GUESS, the only heat transfer modes considered are conduction and advection. In the case study, all four principal modes are considered to determine whether a constant cover resistance, and a single time constant are valid assumptions or not.

The time constant is the amount of time needed for the system to dissipate approx 63% of its initial difference from steady state. For a linear first order system, the constant is equal to CR, where C & R are the thermal capacitance and resistance.

$$\begin{aligned} R &= \frac{r_{total}}{A} \\ C &= \rho C_P V \end{aligned} \qquad (9.1)$$

Thermal resistance and capacitance

Although the system in the case study is non-linear, time constants provide important information concerning the relative importance of different components to overall system dynamics.

Aside from the time constant, the other important dynamic parameter is the steady state temperature. Steady state temperature is useful to determine whether components should be lumped together or not, and will be used to assess the validity of the constant cover conductance assumption.

$$T_{SS} = \frac{U_1}{U_{total}} T_1 + \frac{U_2}{U_{total}} T_2 + \ldots + \frac{U_n}{U_{total}} T_n \tag{9.2}$$

Steady state temperature

Assuming a constant conductance on both sides of the cover, the following result should be obtained for steady state cover/indoor air temperature difference:

$$\frac{1}{U_{cover}} = \frac{1}{U_{film,o}} + \frac{1}{U_{film,i}} + \frac{1}{U_{glass}} \tag{9.3}$$

Cover conductance

$$\frac{(T_{in} - T_{cover})}{(T_{in} - T_{out})} = \frac{U_{cover}}{U_{film,in}} = \frac{R_{film,in}}{R_{cover}} \tag{9.4}$$

Cover Temperature Relation for constant cover conductance and film coeff.

In the case study, this assumption of cover temperature as an average with constant weights with respect to indoor and outdoor temperatures will be tested.

The test case consists of a single layer glasshouse, whose area and volume are the same as the one used in the Model Simulation section. Conduction via framing and the floor are ignored as suggested by Bakker et al (1995), indoor and outdoor temperatures are assumed to be constant, and the temperatures of both faces of the glass are assumed to be equal. The average thermal conductivity of glass is 0.8 W/m-K, so for a 3.2 mm pane (standard size), the area based conductance should be 250 W/m^2-K. The U-Value for single pane glass (ASHRAE 2001) is 5.91 W/m^2-K. Thus the assumption of equal temperatures on both sides is valid, since the bulk of the resistance occurs in the boundary layers. Since boundary layer properties can vary greatly with temperature and wind speed, it becomes all the more important to see if the constant cover conductance assumption is correct.

Case Study Formulation

Radiative transfer occurs between the canopy and the cover, the cover and the sky, and the canopy and the sky. All radiative transfers are modeled using the Ohm's Law analogue. Radiation between the canopy and the cover is treated as if the canopy were isothermal with respect to the air. Cover emissivity is assumed to be 95% (glass), and thus radiation between the canopy and the sky is ignored. A view factor of 0.73(A_{floor} / A_{cover}) from Takakura (1989) is used for canopy to cover transfers, and view factor of 1 is used for sky to cover transfers. Sky temperature is modeled using the Swinbank model, see Campbell, 1998.

$$\begin{aligned} \varepsilon_{sky,clear} &= 9.2*10^{-6} T_a^{\ 2} \\ T_{sky} &= T_{air} \varepsilon_{sky}^{\ 1/4} \end{aligned} \tag{9.5}$$

Swinbank clear sky temperature and emissivity

The effects of cloud cover upon sky temperature are modeled using Monteith and Unsworth's (1990) correlation:

$$\varepsilon_{sky,cloud} = \left(1 - .84c\right) \varepsilon_{ac} + 0.84c \tag{9.6}$$

Cloudy sky emissivity

Convective heat transfer takes place between the cover and the atmosphere, between the cover and the greenhouse airspace and between the air and the canopy.

Because of the length and velocity scales involved, each convective exchange is modeled differently. Convection at the exterior is treated as a form of turbulent forced convection. A linear equation from Bot (1983) is used:

$$r = \frac{1}{2.8 + 1.2u_{wind}} \qquad (9.7)$$

Outdoor convection resistance

Due to the extremely low wind speeds, and extensive cover length, convection along the inner surface of the cover is assumed to be natural and turbulent. The expression for turbulent natural convection in air with two horizontal plates, hot plate at bottom, cold plate on top, expressed as a Nusselt # from Table 1.

$$Nu = 0.13Gr^{1/3} \qquad (9.8)$$

Same equation expression above written as a resistance:

$$r = \frac{\ell}{\rho C_p k Nu} \qquad (9.9)$$

We apply the definition of the Grashof # and find that length scale cancels for turbulent forced convection.

$$Gr = \frac{\beta \ell^3 \Delta T}{\nu^2}$$

$$Gr = \left(\frac{|T_s - T_{air}|}{T_{air}} \right) \frac{\ell^3}{\nu^2} \quad \text{(ideal gas)}$$

$$g = .13 \left| \Delta T^{1/3} \right| \rho C_P k \left(\frac{g}{\nu^2 T_{ref}} \right)^{1/3}$$

For air in standard conditions (295K) using metric units, r can be written as:

$$r = \frac{1}{1.85\left|\Delta T\right|^{1/3}} \quad \frac{\mathrm{W}}{\mathrm{K{\bullet}m^2}} \tag{9.10}$$

Natural convection at cover

Resistance to heat transfer by ventilation or infiltration is modeled using the following equation

$$r_{inf+vent} = \rho C_P \left(\frac{V_{greenhouse}}{A_{cover}}\right)\frac{ACH}{3600} \tag{9.11}$$

Ventilation resistance where ACH is air changes per hour

Case Study Results

The results of the case study are presented below. Cover conductivity is the effective conductance of the glazing to heat (radiation + boundary layer) neglecting infiltration and ventilation. Floor area, glazing area, and enclosed volume were the same as they were in the simulation.

T air	Tinside	Tcover	gcover	Tin-Tair	Tin-Tcover
250	290	264	5.6	40	26
260	295	271.3	5.87	35	23.7
260	290	268.7	5.94	30	21.3
265	295	273.7	6.05	30	21.3
270	295	276.24	6.28	25	18.76
275	295	278.9	6.6	20	16.1
280	295	281.6	7.1	15	13.4
285	295	284.4	7.28	10	10.6
285	298	285.8	7.42	13	12.2
290	298	288.8	6.77	8	9.2
280	298	283	6.81	18	15
300	305	298.2	6.05	5	6.8
290	305	292	7.15	15	13
295	305	295.1	7.93	10	9.9
Average			**6.711538**		

Table 8 Effect of clear sky upon cover conductances ($Wm^{-2} \cdot K^{-1}$), 3 ms^{-1} outdoor wind.

T air	Tinside	Tcover	gcover
260	295	275	4.85
270	295	280	4.87
280	295	285	4.91
280	298	287	4.96
290	295	291	5.24
290	298	292	5.08
300	310	303	5.224
		Average	**5.019143**

Table 9 Effect of full cloud cover upon cover conductance, 3 ms^{-1} wind.

wind speed	T air	T inside	gcover
0	270	295	5.79
5	270	295	6.52
10	270	295	6.96
		Mean	6.423333

Table 10 Effect of wind upon cover conductance (clear sky)

wind speed	T air	T inside	gcover
0	270	295	4.1
5	270	295	5.26
10	270	295	5.94
		Mean	5.1

Table 11 Effect of wind upon cover conductance (cloudy sky)

Surface	Exchange Rate: AC/H	Resistance	Capacitance	Time Constant sec	Time Constant min
Cover		3.80E-05	6.E+06	219	3.6
Cover (through conduction only)		5.06E-06	6.E+06	29	0.5
Air Space	0	1.98E-04	4.50E+06	889	14.8
	1	1.59E-04	4.50E+06	713	11.9
	2	1.32E-04	4.50E+06	595	9.9
	3	1.14E-04	4.50E+06	511	8.5
	5	8.85E-05	4.50E+06	398	6.6
	7	7.25E-05	4.50E+06	326	5.4
Plant Canopy		7.57E-05	5.27E+06	400	6.7
Soil(styrofoam blocks)		2.27E-04	6.96E+07	15820	264
Soil(individual containers)		4.56E-05	6.96E+07	3177	52.9
Soil conduction(lengthwise)		9.391E-04	6.96E+07	65386	1090
Soil conduction(traverse)		2.424E-05	6.96E+07	1688	28.1

Table 12 Example time constants inside greenhouse

Parameter	Value	Unit	Source
Pane thickness	3.2	mm	ASHRAE
Area-based glass heat capacity	7284.5	Ws/m2-K	
Resistance Glass single pane (pane only)	0.004	m2/W-K	
Total Resistance Glass single pane	0.591	W/m^2K	ASHRAE Fundamentals 2001
2 cm Flat Plate Resistance	240	sm^{-1}	Bakker et al.
Soil Thermal Conductivity	0.4	W/m-K	Campbell
Air Thermal Conductivity	0.0257	W/m-K	Jones
Glass Thermal Conductivity	0.8	W/m-K	Campbell
Glazing Conductance	250	W/m^2K	Calculated
Glass Density	2710	kg/m^3	Campbell
Specific Heat glass	840	Ws/kg-K	Campbell
Specific Heat (Soil)	3000	Ws/kg-K	Campbell
Specific Heat(Soil) volumetric	3000	kJ/m^3	
Specific Heat (biomass--volumetric)	3000	Ws/kg-K	Bakker et al
Air Specific Heat(volumetric) 22C	1212	J/m^3-K	Jones
Canopy B.L Resistance	0.033	m^2-K/W	Bakker et al:
Soil B.L Resistance	0.099	m^2-K/W	rLeaf*LAI
Indoor Air Speed	3.000	cm/s	
Cell Diameter	3.000	cm	
Pot Film Resistance	0.248	m^2-K/W	Bakker
Soil Surface BL Resistance	0.124	m^2-K/W	
Pot Surface Area	88.000	cm^2	
Surface Area/Container Area Ratio	12.449		
Pot Film Resistance (Normalized)	0.020	m^2-K/W	
Greenhouse Volume	3710	m^3	From GUESS parameters
Glazing Area	790	m^2	From GUESS parameters
Glazing Volume	2.528	m^3	Calculated
Floor Area	581	m^2	From GUESS parameters
Plant Size	5	g	Fresh wt, assume 40% dry matter
LAI	3	Unitless	
Container volume	66	cm^3	Container nursery manual vol 2
Container Area	7.07	cm2	Container nursery manual vol 3
Cell density	807	$cells/m^2$	Container nursery manual vol 2
Cell equiv length	9.34	cm	estimated
Floor Utilization	75%	Growing area	Container Tree Manual vol 1
Container Utilization	43%	Asoil/Afloor	
U-Value (Glazing Total)	6.4	W/m^2-K	ASHRAE Fund. Used for air space
U-Value (Bidirectional) Heat leaves both sides	33.3	W/m^2-K	used for glazing time constant
U-Value Vent+Infiltration per ACH	1249.033	Wh/K	equiv. conductance for vent/inf
Indoor Film Coeff	8.6	W/m^2-K	Calculated using Model
Outdoor Film Coeff	24.7	W/m^2-K	Calculated using Model
Thermal Conductance (Single Pot-lengthwise)	1.8328	W/m^2-K	
Thermal Conductance (Single Pot-traverse)	5.7043	W/m^2-K	

Table 13 Parameters used for case study

Parameter	Value	Unit
Tinside	295	K
Toutside	275	K
VF(cover:sky)	1	unitless
VF(cover:floor)	0.8	unitless
wind speed	2.5	m/s
cloud cover	0.25	fraction

Table 14 Parameters used to calculate Indoor and Outdoor Film Coeff. Assuming negligible glass resistance

Case Study Discussion

Cover Conductance

Under clear sky conditions, both cover conductance and temperature showed much variability. Cover temperature evens drops below air temperature occurring at T_{in}-$T_{out} \leq$ about 5K. Cover conductance tended to increase as outdoor temperature increased. However under cloudy skies, cover conductance assumed a nearly constant albeit lower value on average due to the lower air: sky temperature difference.

As to be expected from eq. (9.7) cover conductance showed an inverse relationship with wind speed, due to declining resistance at the outdoor boundary layer.

Longwave losses tend to be greater and more variable under clear sky conditions due to the greater difference in temperature between air and sky, owing much in part to the temperature dependent sky emissivity.

Since free convective resistances are directly proportional to temperature difference to the 1/3 power, and temperature difference in turn is linearly proportional to flux, one would expect a higher cover conductance with greater longwave flux, and in times when T_{in}-T_{out} is small, subcooling of the cover below air temperature will occur until natural convection can balance longwave transfer to the sky. Indeed this was observed during the case study. In reality, it is unknown whether this subcooling would actually occur, as the effects of natural convection diminish other heat fluxes such as conduction from the frame or floor, condensation, or absorption of solar radiation by the glazing would become noticeable. Since natural convection is dynamic process, where h_{conv} is proportional to $\Delta T^{1/3}$, it is not fully clear whether the steady state values of conductance and cover temperature predicted would occur in real life.

The cover conductance under cloudy skies averages about 5.1 $W \cdot m^{-2} \cdot s^{-1}$ (all wind speed and temperature combinations tested). Under clear conditions, cover conductance averages about 6.7 $W \cdot m^{-2} \cdot K^{-1}$, close to the value recommended by ASHRAE (2001) for single pane greenhouse construction which is 6.4 $W \cdot m^{-2} \cdot K^{-1}$. In GUESS, a cover conductance of 5.76 $W \cdot m^{-2} \cdot K^{-1}$ was used which is close to the average of the clear and cloudy conductances. The value of 5.91 $W \cdot m^{-2} \cdot K^{-1}$ recommended by ASHRAE for center of glazing single pane plate glass windows is exactly equal to the average of clear and cloudy sky conductances. Thus, the case study reasonably approximates the heat processes which occur at the cover.

Time Constants

Based on the results in table 12, it appears that the air space is not the only significant thermal storage device and that other parts of the greenhouse structure (primarily the soil) can have much longer time constants. At all the ventilation rates tested, the cover time constant is significantly shorter than that of the air, and only at the highest ventilation rates (AC/hr>5), does the canopy time constant become comparable to that of the air.

At long timescales, thermal storage from the soil is far more capable of contributing to the dynamic behavior of the greenhouse in response to temperature change than the air space. Also the time constant for soil varies greatly with the planting method used. The time constants for soil (both for surface temperature and for bulk conduction) are far greater when styrofoam cell trays are used versus individually isolated pots. Further research is needed to quantify some of these dynamic effects, and to see how they would affect model predictions of temperature, growth, and energy cost.

Nonetheless, the GUESS framework serves as a convenient starting point should such one chose to embark on a more detailed heat transfer study. For the purposes of climate control and energy modeling air temperature is of far greater concern than either the soil or canopy.

Yet, at the plant scale, it is the soil and canopy temperatures are paramount. As a simplifying assumption, in GUESS, the soil and canopy are taken to be isothermal with the air. Further simulation is needed to see the magnitude of the errors in growth rate that could result from such an approximation, e.g.: photosynthesis rate too slow, root respiration too high, etc. Such simulation research would be valuable for the creation of model-based control systems based around the plant microclimate as opposed to the air space, and could lead to increased energy savings in the future.

Cover Temperature

One result of assuming a constant cover resistance is that cover temperature is a linear function and should be the weighted average of indoor and outdoor temperatures(9.4). The difference between indoor and cover would be an intercept-less function of outdoor and cover

Under clear skies, see figure 32, this is a poor characterization, and sky temperature must be inserted into the weighing function, since net flux from the cover to the sky may still occur even if the indoor and outdoor temperatures are equal. For that reason, the concept of a cover conductance is somewhat ambiguous for clear sky conditions.

Nonetheless under clear skies, and constant wind speed, for all purposes, cover temperature/indoor temperature difference is a linear function of indoor and outdoor temperature difference, the intercept being an acknowledgement of the longwave flux to the sky.

$$\begin{aligned} T_{indoor} - T_{cover} &= .554(T_{in} - T_{out}) + 4.74 \qquad \text{best fit} \\ T_{cover} - T_{indoor} &= .765(T_{in} - T_{out}) \qquad \text{zero intercept} \end{aligned} \tag{9.12}$$

Temperature Regressions (linear)

A better procedure would be to calculate cover temperature based on prevailing conditions, and then calculate heat loss to the cover based on eq. (9.10). Assuming a steady state approximation for cover temperature is not an unreasonable assumption considering that cover time constant is exceeded by that of the air space at most ventilation rates. However, a dynamic heat balance would be preferable since the cover time constant is still greater than the model time step (2 minutes) and would eliminate the need for any iterative search techniques. A linear regression such as eq. (9.12) could be used.

The largest problem observed with constant cover conductances is that they tend to overestimate the rate of heat transfer, particularly when skies are cloudy and wind speeds are low, most glazing conductances were computed as design values, for use in extreme conditions, not necessarily average conditions. As said before, when indoor-outdoor temperature differences are low, constant cover conductances tend to underestimate the rate of heat transfer, since longwave transfer to the sky may occur even if indoor and outdoor temperatures are equal. The standard practice for greenhouse energy modeling is to use a constant cover conductance when estimating cover losses. A better procedure to estimate net losses to the outdoor would be to:

1. Calculate the cover temperature based on 3 variables: indoor, outdoor, and sky, or better yet use the linear regression presented in (9.12) to estimate cover temperature solely using indoor and outdoor temperatures, eliminating the need for a dynamic heat balance.
2. Calculate indoor film coefficient

$$h_{film} = 1.85 \left| T_{cover} - T_{in} \right|^{1/3} + 4 \varepsilon F \sigma \left(\frac{T_{cover} + T_{in}}{2} \right)^3 \tag{9.13}$$

3. Use this film coefficient and glazing pure conductance (if necessary) to calculate the apparent conduction heat flux to the outside.

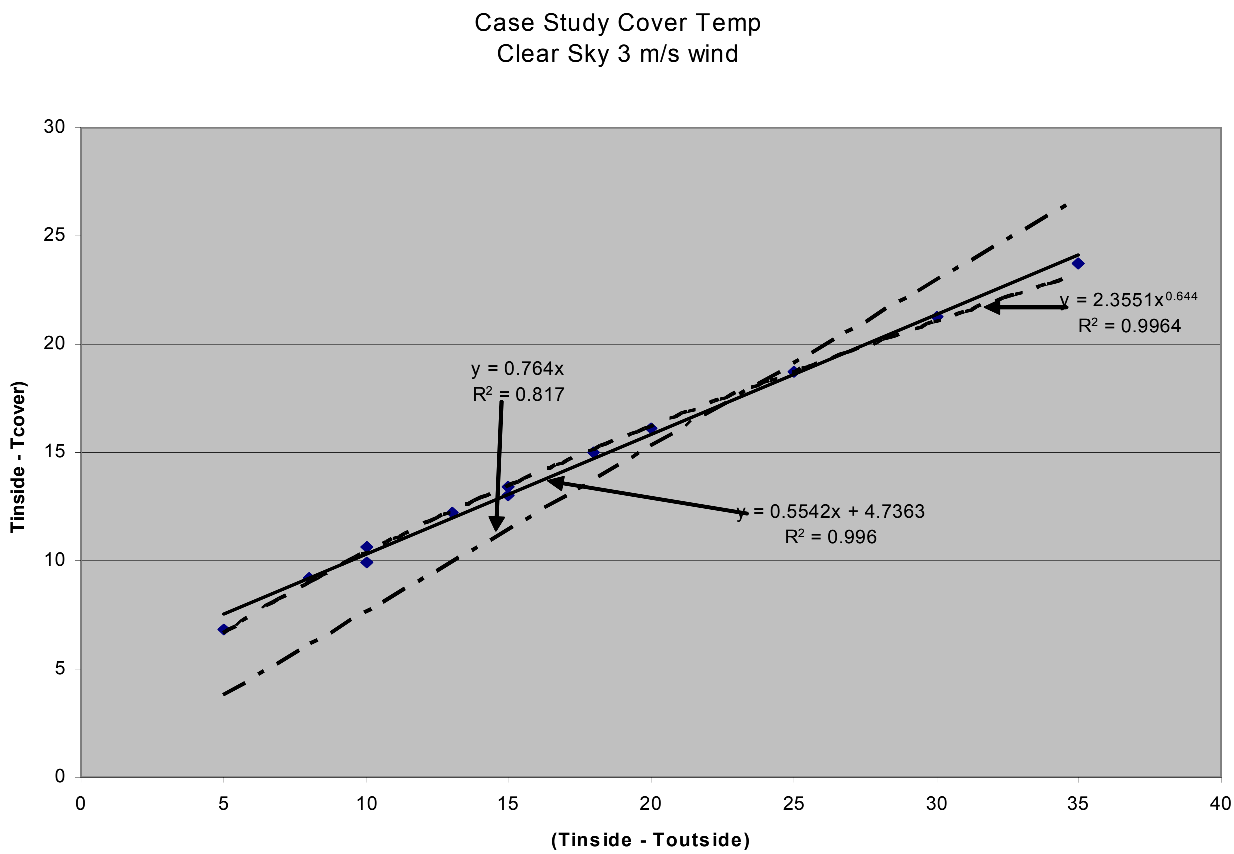


Figure 31 Linear regressions for constant cover (glazing) conductance. The best fit line has a non-zero intercept, which makes it difficult to apply Eq. (9.4) to calculate glazing temperature.

Sensitivity Analysis

Earlier in the case study it was shown that cover conductance can be quite sensitive to outdoor conditions esp. wind speed and cloud cover. Because of this, a sensitivity analysis was conducted around energy cost to see the relative magnitude of U-Value upon energy cost.

Glazing U-Value (W/m^2-K)	Yearly Gas Cost in $1000's	Relative dev. In U-Value	Relative dev. In Gas Cost
4	$ 2.58	-33%	-29%
5.1	$ 3.18	-15%	-13%
5.7	$ 3.49	-5%	-4%
6	$ 3.64	AVG	AVG
6.24	$ 3.76	+4%	+3%
6.7	$ 3.96	+12%	+9%
7.7	$ 4.40	+28%	+21%

Table 15 Sensitivity Analysis of heating cost versus U-Value

As one can see in the above table, a change in magnitude of glazing U-Value corresponds to a slightly lower change in magnitude of heating cost. For the lower U-Values, this decrease in magnitude is due in part to infiltration effects, for the higher U-Values it is unclear, what is at work. Because heating costs are almost linearly sensitive to glazing U-Value, and because glazing U-Value is a function of outdoor conditions, it is important to use a U-Value which corresponds to the local climate.

Chapter 10

CONCLUSION

The growth rate of any greenhouse crop is a function of the following state variables:

1. Light
2. Temperature
3. CO_2
4. relative humidity
5. Soil(texture, water content, and nutrition)
6. Genetics

Precise control over relative humidity is usually impractical, (Albright, personal communication). Genetics and soil, with the exception of nutrition and water content, are fixed at the time of planting. Assuming that the soil remains well-watered and well-fertilized, only three state variables (light, CO_2, and temperature) remain available for real time control, and it is the interaction of these variables with the crop that is modeled in GUESS.

These variables, listed in order of increasing cost are:

1. CO_2(uncontrolled enrichment)
2. Temperature
3. Light
4. CO_2(controlled enrichment)

CO_2 is mentioned twice since it is a byproduct of combustion of any heating fuel. Uncontrolled winter and night enrichment can be obtained essentially free of charge. More precise control requires the use of a liquefied CO_2, catalytic combustion units, or dry ice (see Landis 1990).

Based upon the results of the simulation, it appears as if light is the limiting factor in crop production both physiologically and economically. In all iterations, low light levels occurred during the beginning of the year. Even with supplemental lighting (200 μmolar, setpoint: 150 μmolar, bandwidth ± 100), it is difficult to eliminate the deficit in light integral that occurs during the wintertime. Owing to the inefficiencies of most common HID lamps: i.e. only about 34% of the radiant energy produced falls within the PAR band compared to 50% for sunlight; and owing to the higher cost of electricity versus natural gas per unit energy consumed, lighting quickly dominates the cost of production for controlled environment systems. Hence the conclusion made by growers interviewed that high levels of supplemental lighting are too costly for the production of Douglas-fir seedlings, grown primarily for reforestation purposes).

The need for supplemental lighting depends strongly on the quantum content of sunlight. During the winter, extremely low peak light levels and short photoperiod result in a negative carbon balance on some days. Due to the lack of a photosynthate storage model in GUESS, this problem manifested itself as a negative growth rate. For 2.1 $\mu mol \cdot s^{-1}/W$ quantum content, either 75 μmolar (with CO_2) or 100 μmolar (without CO_2) supplemental lighting was required to achieve positive carbon gain in the 3rd growing season.

For 2.35 μmol quantum content, no supplemental lighting was required. Although, actual quantum data from Corvallis, Oregon was unavailable, these results agreed with Landis (1990) and personal communication with Mehlschau who suggest wintertime growth was possible with only photoperiodic lighting, which is typically $\leq 15\ \mu mol \cdot s^{-1} \cdot m^{-2}$ for most conifer seedlings.

CO_2 levels were mediated by the heaters and not independently controlled. Substantial enrichment occurred at dawn/dusk and night, when the rates of photosynthesis would be at their lowest, were no supplemental enrichment used. The effects of solar heating during the morning resulted in premature shutdown of enrichment and depletion of CO_2 by photosynthesis kept CO_2 levels below atmospheric until ventilation occurred.

However if CO_2 enrichment is combined with supplemental lighting then significant improvements in yield can occur over supplemental lighting without enrichment. This was demonstrated by Ho in lettuce (Ho 2003), and modeled in Douglas-fir in figures 20 & 21. Shorter growing seasons under supplemental lighting were observed, It remains to be seen whether actively controlled CO_2 enrichment is economically viable or not. Although CO_2 enrichment has a history of commercial success with Southern pine species (Landis 1990), it is not known if commercial nurseries are using CO_2 enrichment with Douglas-Fir crop (personal communications with Mehlschau and Landis). One thing worth pointing out is that due to the high mesophyll resistance and low photosynthetic temperature optimum of Douglas-Fir needles, long term enrichment of ambient CO_2 concentrations result in only small increases in dry matter production (Lewis et al. 2001).

Owing to the lack of any confirming studies, the results of the GUESS simulation should be considered preliminary, not suitable yet for detailed control design or research.

Nonetheless, based upon the numerical case study, many of the assumptions present in the GUESS are reasonable although not correct according to heat transfer theory. Improvements can be made adding additional subsystems to the model. Additional surface heat balances can be added to take in the dynamic effects of the cover, canopy and soil, and their inherit differences with air temperature. A constant cover conductance can be replaced with a more realistic model which includes longwave radiation with the floor and canopy and with the sky, natural(free) convection on the inside face and forced convection on the outside.

While the model used by Bot (1983) might be too advanced for practical use, a simple yet mechanistic cover light transmission model, such as the one in Wang and Boulard, 2000 could be implemented so that transmissivity values could stem naturally from material properties and greenhouse architecture.

Another area requiring further work lies in the plant model. So far, no provisions have been made for photoperiod effects. The effects of water status are ignored except at the stomatal level; nutrient levels are assumed optimum; maximum rates of photosynthesis and respiration are assumed constant for each tissue pool, and all biomass is assumed labile. As a subsequent refinement, a more realistic carbon balance including the effects of storage, photoperiodicity, N-content and water status could be incorporated, as a means to test the effects of stress upon growth. Also planting cell density dependent and red/far red dependent seedling allometry (see Timmis and Tanaka 1976) can be incorporated into GUESS to produce a more realistic plant simulation. Finally, stomatal capacitance and conductance terms used in the model were roughly estimated, and should be modified to reflect real data.

Nevertheless, based upon the numeric case study and the test case performed on a virtual Douglas-fir seedling nursery greenhouse, the model appears to produce reasonable results, and at that the model in its current form is qualitatively correct. Now what remains, is further refinement if deemed necessary, but more importantly, a validation study to determine if the model is quantitatively correct as well.

Chapter 11

WORKS CITED

Albright, Louis D. Environment Control for Animals and Plants. St. Joseph, MI: American Society of Agricultural Engineers, 1990.

Aldrich, Robert A., and John W. Bartok, Jr. Greenhouse Engineering. 3rd ed. Ithaca, NY: NRAES (Natural Resource, Agriculture and Engineering Service), 1994.

Anekonda, T S., and W T. Adams. "Genetics of Dark Respiration and Its Relation to Drought Hardiness in Coastal Douglas-Fir." Thermochimica Acta 349 (2000): 69+.

ASHRAE Handbook of Fundamentals. SI ed. Vol. 1. Atlanta, GA: American Society of Heating, Refrigeration and Air Conditioning Engineers, 2001.

Bakker, J C., G P A. Bot, H Challa, and N J. Van De Brakk, eds. Greenhouse Climate Control: an Integrated Approach. Wageningen, NL: Wageningen Pers, 1995.

Baldocchi, Dennis. "Lecture 10: Leaf Photosynthesis & Respiration." ESPM 228: Advanced Topics in Biometeorology. Dept. of Environmental Science, University of California, Berkeley, CA. Apr. 2004. <http://nature.berkeley.edu/biometlab/espm228/>.

Baldocchi, Dennis. "Lectures 36: Leaf Photosynthesis and Respiration, Models." ESPM 129: Biometeorology. Dept. of Environmental Science, University of California, Berkeley, CA. Nov. 2004. <http://nature.berkeley.edu/biometlab/espm129/>.

Ball, J T., I E. Woodrow, and J A. Berry. "A Model Predicting Stomatal Conductance and Its Contribution to the Control of Photosynthesis under Different Environmental Conditions." Progress in Photosynthetic Research: Proceedings of 7th International Congress 4 (1987).

Bot, G P A. "Greenhouse Climate: From Physical Processes to a Dynamic Model." Diss. Univ. of Wageningen, 1983.

P Burns, and Deru, M. Infiltration and Natural Ventilation Model for Whole-Building Energy Simulation of Residential Buildings. NREL(National Renewable Energy Laboratory), US Dept. of Energy. Golden, CO, 2003. pub: 33698.

Brutsaert, W. "On a Derivable Formula for Longwave Radiation from Clear Skies." Water Resources Research 2 (1975): 742-744.

Campbell, Gaylon S., and John M. Norman. An Introduction to Environmental Biophysics. 2nd ed. New York, NY: Springer-Verlag, 1998.

Carlson, William C., and D Elaine Miller. Target Seedling Root System Size, Hydraulic Conductivity, and Water Use During Seedling Establishment. Target Seedling Symposium, 13 Aug. 1990, Western Forest Nursery Association. <http://www.rngr.net/Publications/proceedings/1990/carlson.pdf>.

Chen, Han Y., and Karel Klinka. "Light Availability and Photosynthesis of *Pseudotsuga menziesii* Seedlings Grown in the Open and in the Forest Understory." Tree Physiology 17 (1997): 23-29.

dePury, D G G., and G D. Farquhar. "Simple Scaling of Photosynthesis From Leaves to Canopies Without the Errors of Big Leaf Models." Plant, Cell and Environment 20 (1997): 537-557.

D.G. Erbs, S.A. Klein and J.A. Duffie, Estimation of the diffuse radiation fraction for hourly, daily and monthly average global radiation, Solar Energy, 28(4), 293-304, 1982.

Farquhar, G D., and A Brooks. "Effect of Temperature on the CO_2/O_2 Specificity of Ribulose-1, 5-Bisphosphate Carboxylase/Oxygenase [Rubisco] and the Rate of Respiration in the Light." Planta 165 (1985): 397-406.

Fritsch, F. N. and R. E. Carlson, "Monotone Piecewise Cubic Interpolation," SIAM J. Numerical Analysis, Vol. 17, 1980, pp.238-246.

Farquhar, G D., S Von Caemmerer, and J A. Berry. "A Biochemical Model of CO_2 Assimilation in Leaves of C3 Species." Planta 149 (1980).

G, Allen R., M Smith, and L S. Pereira. "An Update for the Definition of Reference Evapotranspiration." ICID Bulletin 43 (1994): 1-92.

Gijzen, H, E Heuvelink, H Challa, L F M. Marcelis, E Dayan, S Cohen, and M Fuchs. "HORTISIM: A Model For Greenhouse Crops And Greenhouse Climate." Acta Horticulturae 456 (1998): 431-450.

Harley, P C., and J D. Tenhunen. "Modeling the Photosynthetic Response of C3 Leaves to Environmental Factors." Modeling Crop Photosynthesis-- From Biochemistry to Canopy. Madison, WI: Crop Science Society of America, 1991. 17-39.

Hyland, R W., and A Wexler. "Formulations for the Thermodynamic Properties of the Saturated Phases of H_2O From 173.15K to 473.15K." ASHRAE Transactions 89.2A (1983): 500-535.

Ho, Jeffrey. Optimum Control of Supplemental Lighting and CO2 Concentration for Controlled Environment Agriculture. Cornell University. 2003.

Jarvis, P G. "The Interpretation of the Variations in Leaf Water Potential and Stomatal Conductance Found in Canopies in the Field." Philosophical Transactions of the Royal Society 273 (1976): 593-610.

Jones, Hamlyn G. Plants and Microclimate. 2nd ed. New York, NY: Cambridge UP, 1991.

Jones, J W. "Model Integration and Simulation Tools." Acta Hort 456 (1998): 411-417.

Koskela, J. "A Process-Based Growth Model for Grass Stage Pine Seedlings." Silva Fennica 34 (200): 3-20.

Kurata, K. "Simulation of Inside Air Temperature, Humidity and Crop Temperature in an Energy Conserving Greenhouse." Acta Hort 245 (1989): 339-345.

Landis, Thomas D., ed. The Container Tree Nursery Manual. Vol. 1-6. Washington, DC: US Department of Agriculture, Forest Service, 1990. Reforestation, Nursery, and Genetic Resources. <http://www.rngr.net/Publications/ctnm>.

Landsberg, J J., and M M. Ludlow. "A Technique for Determining Resistance to Mass Transfer Through the Boundary Layers of Plants with Complex Structure." The Journal of Applied Ecology 7 (1970): 187-192.

Langhans, Robert W. Greenhouse Management : a Guide to Structures, Environmental Control, Materials Handling, Crop Programming, and Business Analysis. 3rd ed. Ithaca, NY: Halcyon Press of Ithaca, 1990.

Leuning, R. "A Critical Appraisal of a Combined Stomatal-Photosynthetic Model for C3 Plants." Plant, Cell, and Environment 18 (1995): 339-357.

Lewis, J D., M Lucash, D Olszyk, and D T. Tingey. "Seasonal Patterns of Photosynthesis in Douglas Fir Seedlings During the Third and Fourth Year of Exposure to Elevated CO_2 and Temperature." Plant, Cell and Environment 24 (2001): 539-548.

Marshall, B, and P V. Biscoe. "A Model for C_3 Leaves Describing the Dependence of Net Photosynthesis on Irradiance." Journal of Experimental Botany 31.120 (1980): 29-39.

Monteith, J L., and M Unsworth. Principles of Environmental Physics. 2nd ed. Oxford, UK: Butterworth Heinemann, 1990.

Parde, J., 1980. Forest Biomass. Forestry Abstracts 41:343-62.

Pruyn, Michele, Mark E. Harmon, and Barbara L. Gartner. "Within-Stem Variation of Respiration in *Pseudotsuga menziesii*(Douglas Fir) Trees." New Phytologist 154 (2002): 359-372.

Qi, Jingen, John Marshall, and Kim G. Mattson. "High Soil Carbon Dioxide Concentrations Inhibit Root Respiration of Douglas Fir." New Phytologist 128.3 (1994): 435-442.

Riha, Susan. Win GAPS ver. 1.1 User Manual. Dept of Earth and Atmospheric Sciences, Cornell University. Ithaca, NY: Cornell University, 2004. <http://environment.eas.cornell.edu/WinGapsMan.v1.1.pdf>.

Ripullone, Francesco, Giacomo Grassi, Marco Lauteri, and Marco Borghetti. "Photosynthesis-Nitrogen Relationships: Interpretation of Different Patterns Between *Pseudotsuga menziesii* and *Populus x euroamericana*." Tree Physiology 23 (2003): 137-144.

Shinozaki, K, K Yoka, K Hozumi, and T Kira. "A Quantitative Analysis of Plant Form: the Pipe Model Theory." Japanese Journal of Ecology 14 (1964): 133-139.

Sugita, Michiaki, and Wilfried Brutsaert. "Cloud Effect in the Estimation of Instantaneous Downward Longwave Radiation." Water Resources Research 29.3 (1993): 599-605.

Takakura, T. "Technical Models of the Greenhouse Environment." Acta Hort 248 (1989): 49-59.

Timmis, R, and Y Tanaka. "Effects of Container Density and Plant Water Stress on Growth and Cold Hardiness of Douglas-Fir Seedlings." Forest Science 22 (1976): 167-172.

Tjoelker, Mark G., Jacek Oleksyn, and Peter B. Reich. "Modeling Respiration of Vegetation: Evidence for a General Temperature-Dependent Q10." Global Change Biology 7 (2001): 223-230.

Walcroft, A S., D Whitehead, W B. Silvester, and F M. Kelliher. "The Response of Photosynthetic Model Parameters to Temperature and Nitrogen Concentration in *Pinus radiata* D. Don." Plant, Cell and Environment 20 (1997): 1338-1348.

Wang, S, and T Boulard. "Measurement and Prediction of Solar Radiation Distribution in Full-Scale Greenhouse Tunnels." Agronomie 20 (2000): 41-50.

APPENDIX A: SIMULATION PARAMETERS

MODEL PARAMETERS

Greenhouse			
Parameter	*Value*	*Unit/ Description*	*Source*
U-Value	5.1	W/m^2-K	cloudy skies
Infiltration Rate	1	air change per hr	Aldrich and Bartok 1994
Volume	3711	m^3	
Floor Area	581	m^2	
Glazing Area	790	m^2	
Perimeter	102	m	
Perimeter Loss	0.7931	W/m-K	Aldrich and Bartok 1994
Internal wind speed	3	cm/s	Bakker et al. 1995

General Plant Properties & Allometry			
Avg. Carbon Content	0.45	g carbon/g dry wt.	Riha 2004
Dry weight density	0.53	g dwt/cc of wood	http://www.simetric.co.uk/si_wood.htm
Specifc Leaf Area	76	g dwt/m^2 of leaf	Ripullone et al 2003
Stem Taper	6.78	cm height/mm dia	Timmis & Tanaka 1976
Canopy Structure	1	dimensionless	1=spherical, 0 = horizonntal, ∞ = vertical
Stem Mass: Basal Area Coeff, K	0.2169	g dry wt/A^2	Timmis & Tanaka 1976
Total Shoot Wt: "Trunk Wt"	5.35	g/g	Calculated
Pipe Model Coefficient	258	mm2 stem area/m2 leaf area	Koskela 2000
Area exponent (basal area: biomass)	1.5	exponent	dimensional analysis
Dia. exponent (dia: height : biomass)	2	exponent	dimensional analysis
Height. Exponent (dia: height: biomass)	1	exponent	dimensional analysis

Respiration			
Q10(root)	1.95	1/K	Qi, Marshall, Mattson 1994
Q10(stem)	1.9	1/K	Pruyn et al. 2002
Root(25C)	9.55	nmol CO_2/g biomass-s	Qi, Marshall, Mattson 1994
Stem(25C)	0.633	nmol CO_2/g biomass-s	Pruyn et al. 2002
Leaf(Dark Respiration[Rd])	0.85	μmol CO_2/m^2 leaf area	Anekonda&Adams 1999
Ha/R: Leaf Respiration	9410	1/K	Anekonda&Adams 1999
Growth Respiration	0.25	μmol respired/μmol avail for growth	Riha 2004

Stomata			
Closed(cuticular) conductance	17	mmol/m^2-s (vapor)	Campbell 1998
Open conductance	330	mmol/m^2-s (vapor)	Campbell 1998
Ball-Berry sensitivity coeff	480	mmol/m^2-s	Estimated
Stomatal capacitance	10	mmol/m^2	estimated
		*: if rel H in %, A_{net} in μmol/m^2-s, CO_2 in ppm	

Photosynthesis			
Quantum efficiency	0.22	μmol electrons/ μmol photons	Walcroft et al 1997, Baldocchi 2004
Jmax(25 C)	85.13	μmol electrons/m_2 leaf area	Ripullone et al 2003
Vcmax(25 C)	30.86	μmol CO_2/m_2 leaf area	Ripullone et al 2003
Ha: Jmax	46000	activation energy kJ/kmol-K	Walcroft et al 1997
Hd:Jmax	199000	deactivation energy kJ/kmol-K	Walcroft et al 1997
Sv	650	entropy term	Walcroft et al 1997
Ha: Vcmax	45000	activation energy kJ/kmol-K	Walcroft et al 1997
	203000	deactivation energy kJ/kmol-K	Walcroft et al 1997
Specificity constant(25 C)	2900	compensation point ppm CO_2/ppm O_2	Baldocchi 2004
Linear Temp Parameter	0.0451	1/K	Leuning 1995
Quadraric Temp Parameter	0.000347	1/K^2	Leuning 1995

Material Properties & Meteorological Data			
Atmospheric Pressure(sea level)	101.3	kPa	
Air Density(20 C)	1.1929	kg/m^3	Campbell 1995
Ideal Gas Constant	8.3145	J/mol-K	
Prandtl #(air) (20C)	0.71	dimensionless	Monteith & Unsworth 1990
Psychrometric Constant	0.000667	1/K, gamma = Patm*6.67E-4	
Latent Heat of Vaporization, Water	540	kcal/kg	Engineering Toolbox http://www.engineeringtoolbox.com
Specific Heat, Air	1006	J/kg	Engineering Toolbox
Specific Heat, Liquid Water	4184	J/kg	Engineering Toolbox
Specific Heat, Steam	2008	J/kg	Engineering Toolbox
Stefan-Boltzmann constant	5.67E-08	W/K^4-m^2	Monteith & Unsworth
viscosity(air, 20C)	0.0000222	m^2/s	Monteith & Unsworth
Thermal conductivity(air, 20C)	0.000015	m^2/s	Monteith & Unsworth
Solar Constant	1360	W/m^2	Monteith & Unsworth
Gravitational Acceleration	9.8	m/s^2	Monteith & Unsworth
Lewis # H2O	0.89	kThermal/D_x	Monteith & Unsworth
Lewis # CO2	1.46	kThermal/D_x	Monteith & Unsworth
Ambient CO2 conc	370	ppm	Mona Loa (2000)
Ambient O2 conc	210000	ppm	
Wind Speed Power Law exponent	0.2	gentle terrain, agricultural	Albright 1990

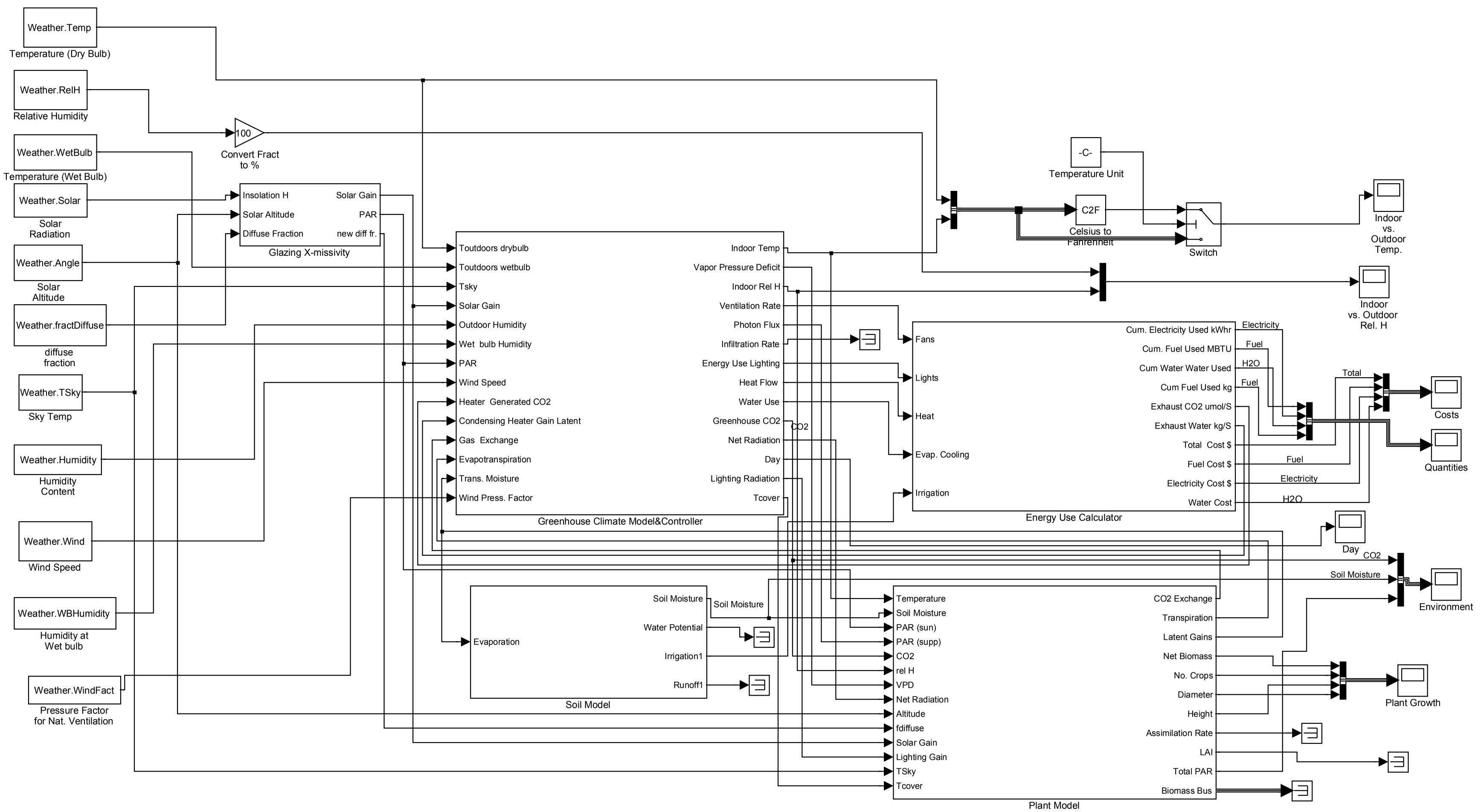
Locational Information			
City	Corvallis, OR		
Latitude	44.63	degrees	Agrimet station coord.
Longitude	123.19	degrees	Agrimet station coord.
Elevation	78	m	Agrimet station coord.

Planting Information			
Cell Size	dia:3.5, depth:12	cm	Container Tree Nursery Manual vol 2
Cell Density	807	cells/m^2	Container Tree Nursery Manual vol 2
Cell Volume	66	cm^3	Container Tree Nursery Manual vol 2
Maximum canopy area	12.4	cm^2	calculated
Floor Utilization	75%	m^2 bench/m^2 floor	Aldrich & Bartok ,1994

Soil Information			
Porosity	75%	vol voids/ total volume	based on Cornell mix
Saturated Water Content	4.5	vol H_2O/vol dry soil	based on Cornell mix
Field Capacity	3.5	vol H_2O/vol dry soil	based on Cornell mix
Permanent Wilting Point	0.75	vol H_2O/vol dry soil	based on Cornell mix

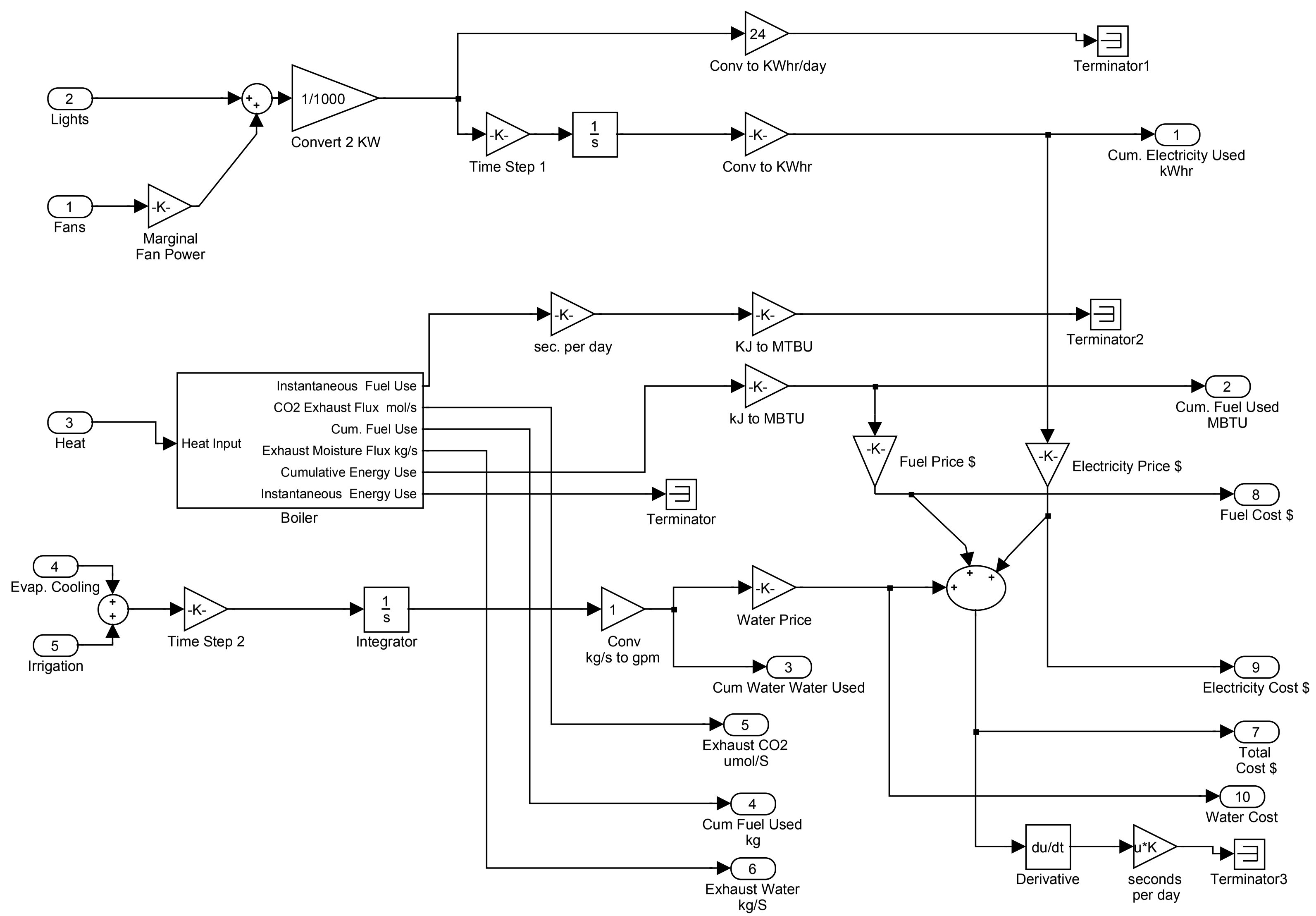
Energy Information			
Lighting	1000W HPS		
Marginal Fan Power Consumption	85	W/m^3-s	
Marginal Lighting Power Consumption	1.5	W/μmol PAR	
Fuel	natural gas		
Fuel LHV	55530	kJ/kg	www.engineeringtoolbox.com
Electricity Price	11.87	¢/kWhr	EIA:http://www.eia.doe.gov/cneaf/electricity/epm/table5_6_a.html
Fuel Price	3.88	$/MBTU	EIA:http://www.eia.doe.gov/emeu/inter-national/ngasprii.html

APPENDIX B: SOURCE CODE & SIMULINK BLOCK DIAGRAMS

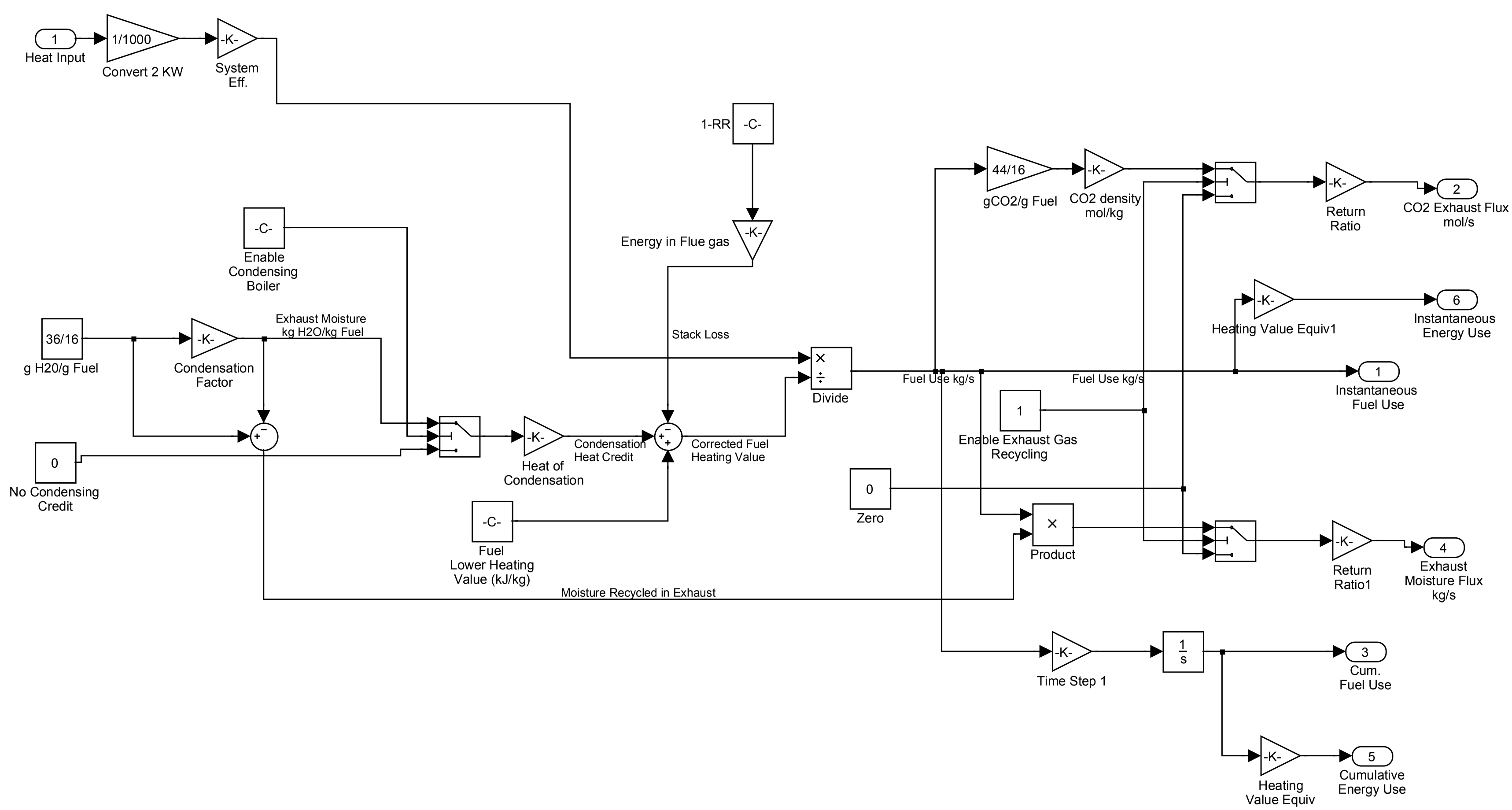


guesssim

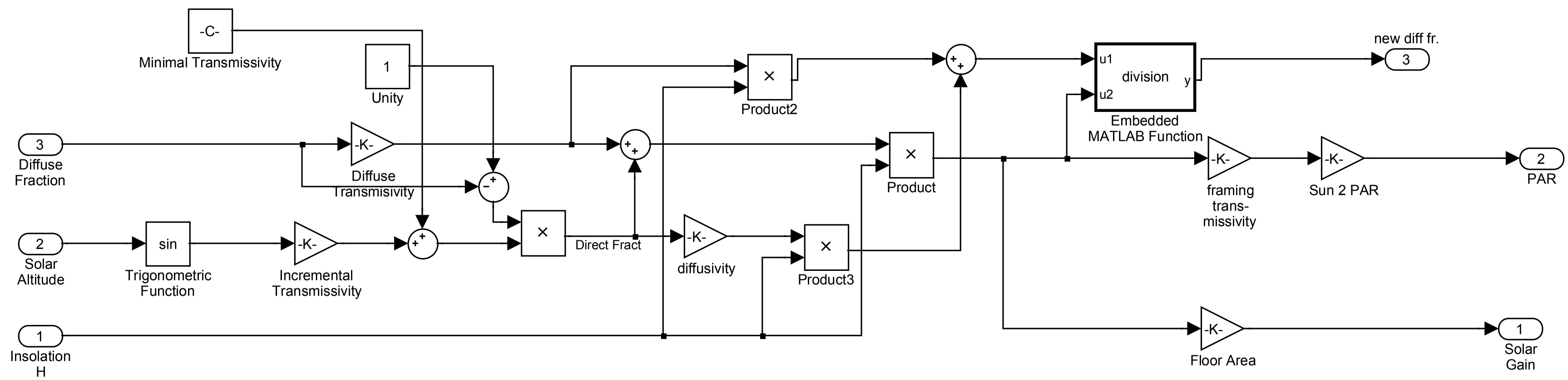
GUESS Model Block Diagrams | (C) 2006 Jamison Hill | 1



guesssim/Energy Use Calculator



guesssim/Energy Use Calculator/Boiler

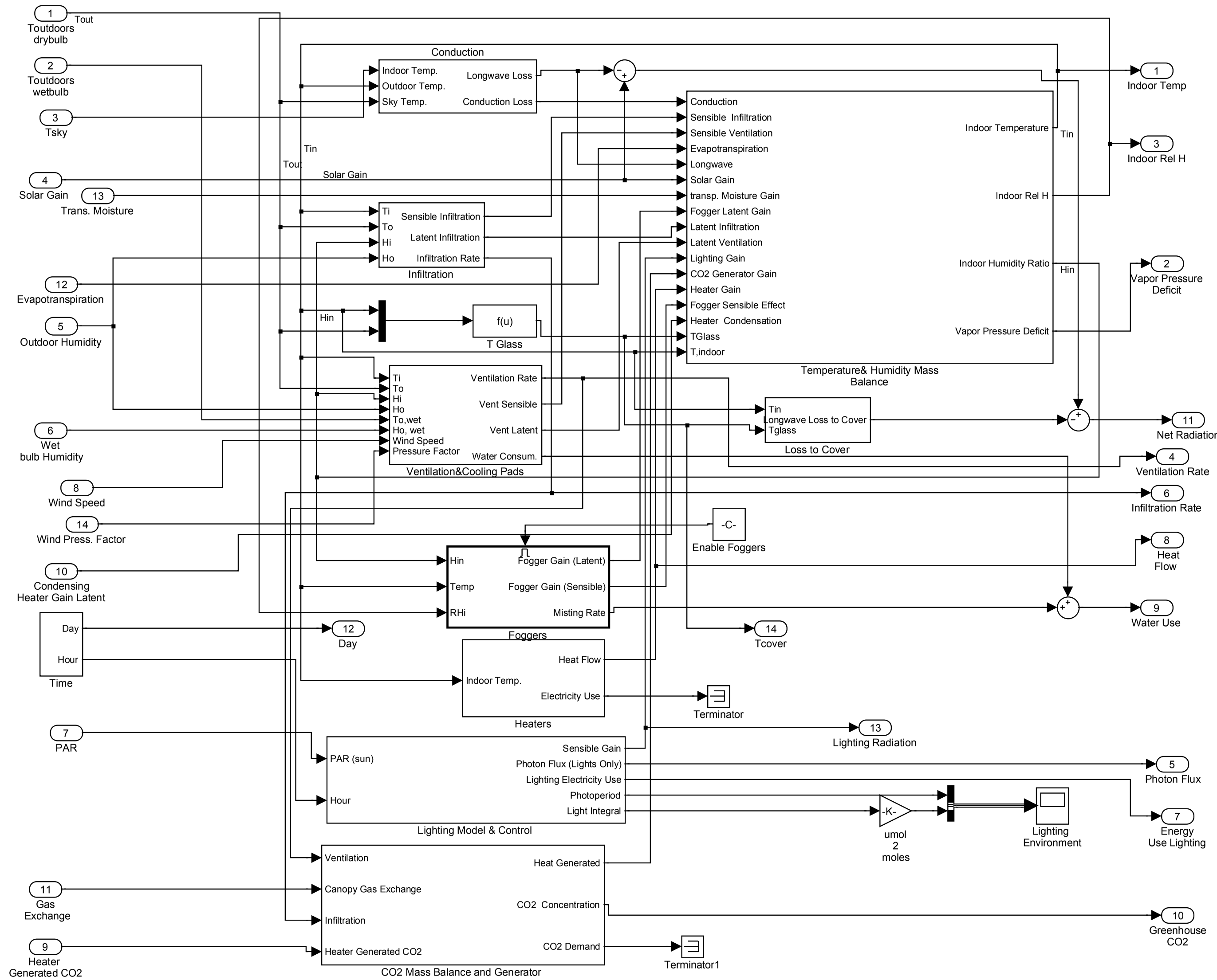


guesssim/Glazing X-missivity

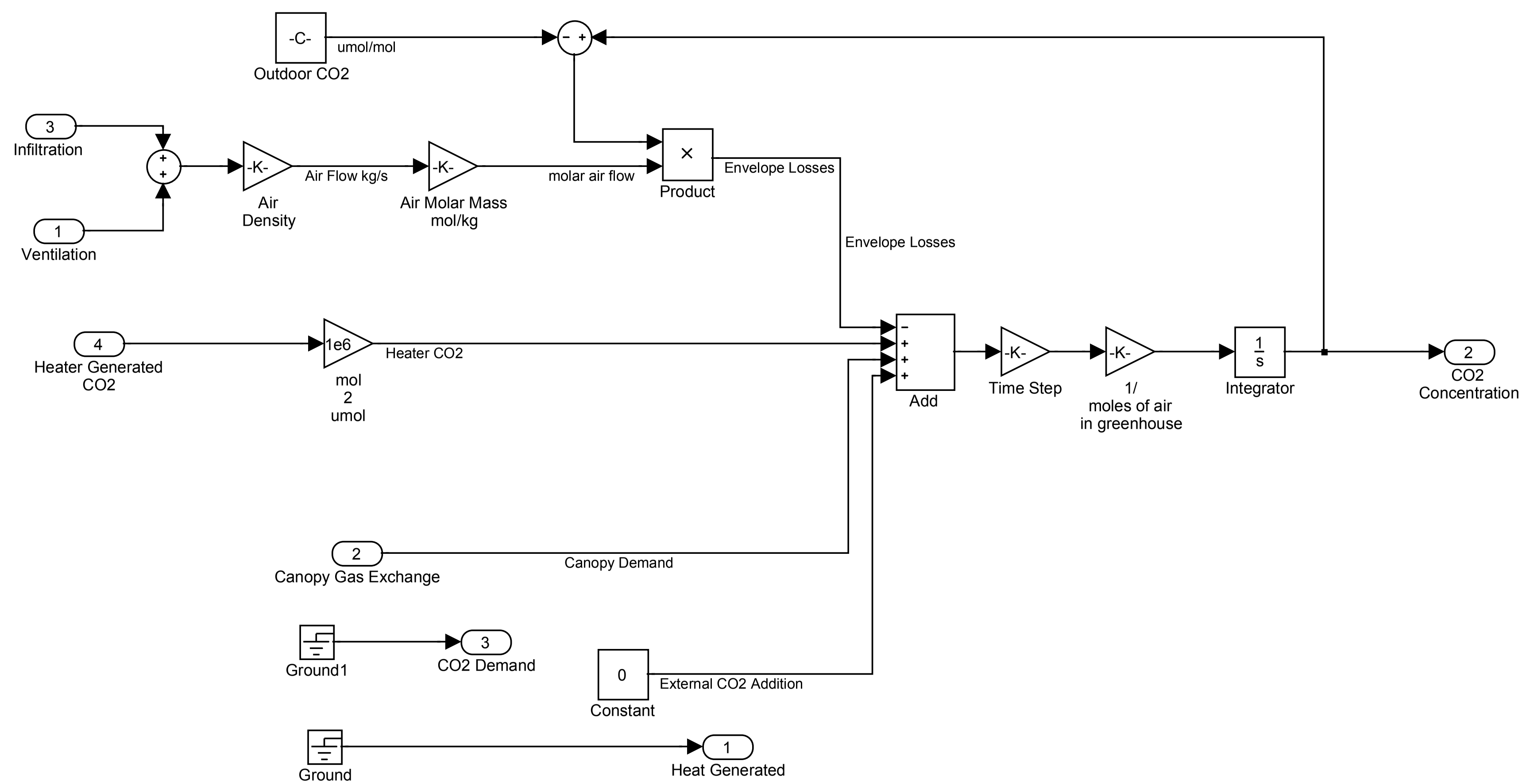
guesssim/Glazing X-missivity/Embedded MATLAB Function.eML_blk_kernel

```
function y = division(u1, u2)
% This block supports an embeddable subset of the MATLAB language
% See the help menu for details.
if u2 ~= 0
    y = u1./u2;
else y = 0;
end
```

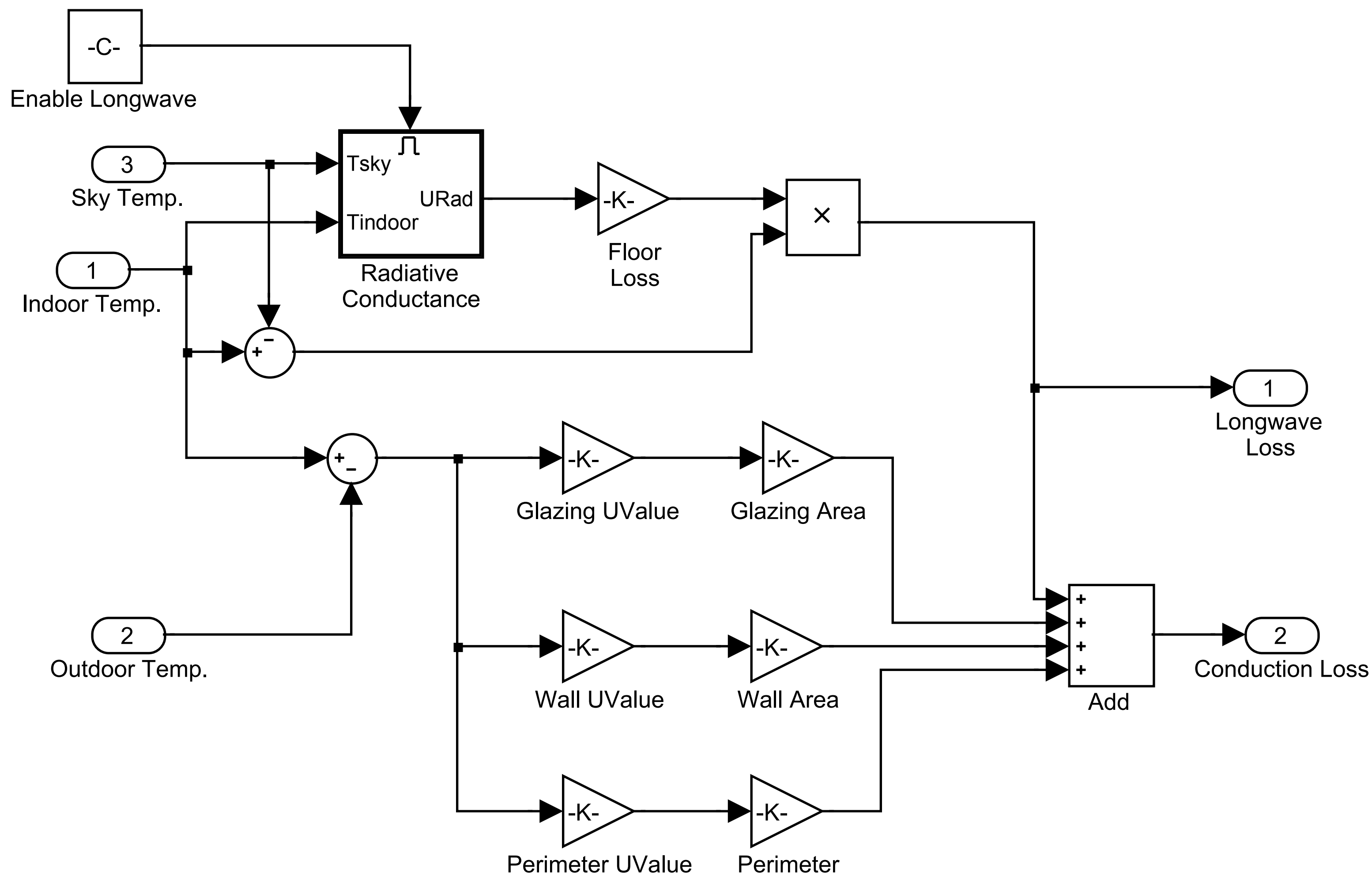
%<fullsystem>



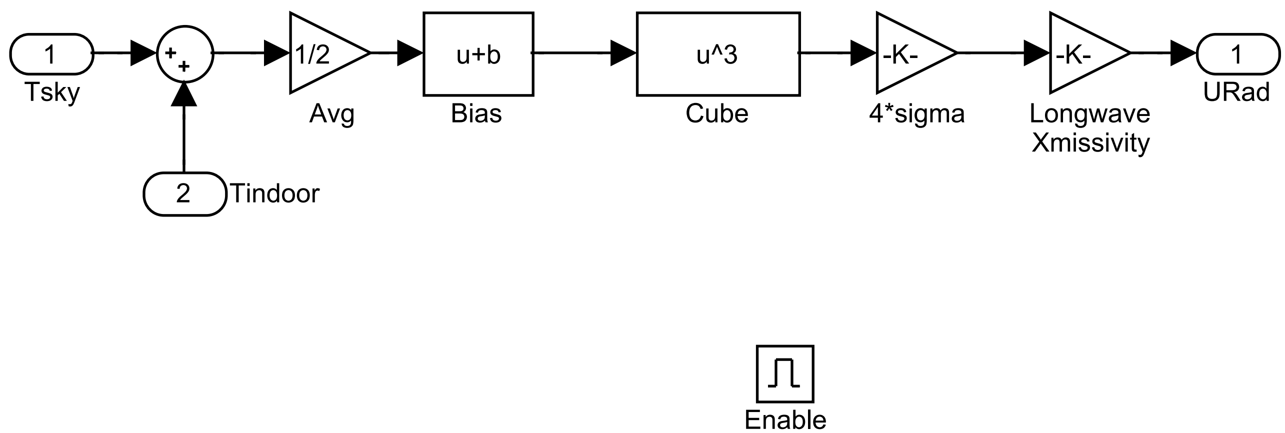
guesssim/Greenhouse Climate Model&Controller



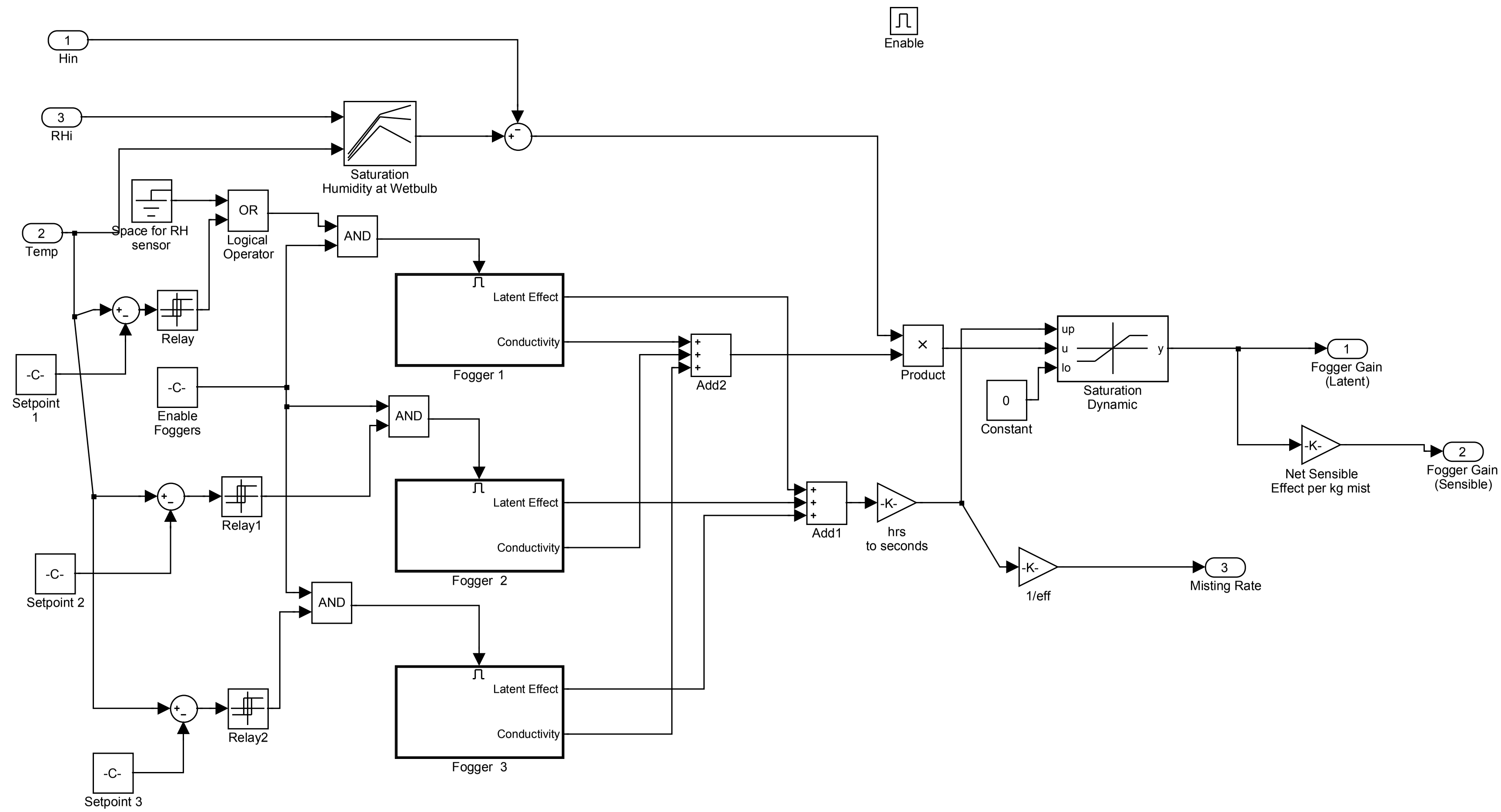
guesssim/Greenhouse Climate Model&Controller/CO2 Mass Balance and Generator



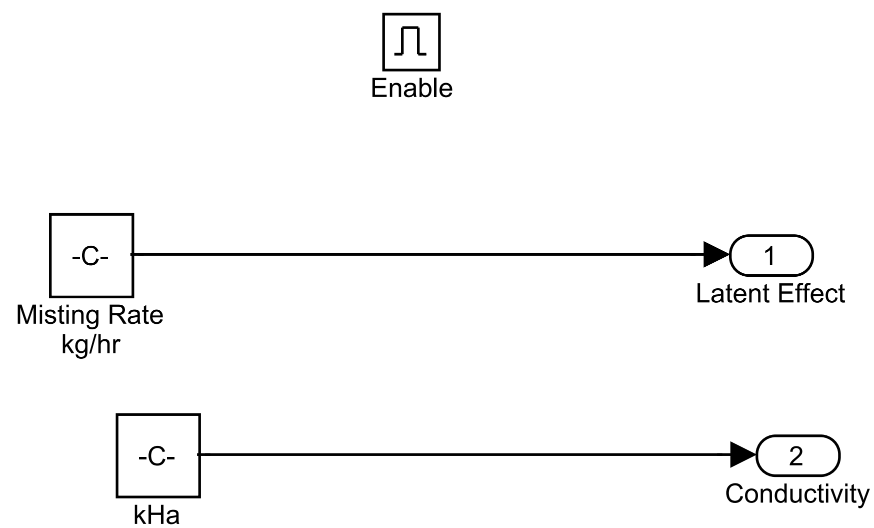
guesssim/Greenhouse Climate Model&Controller/Conduction



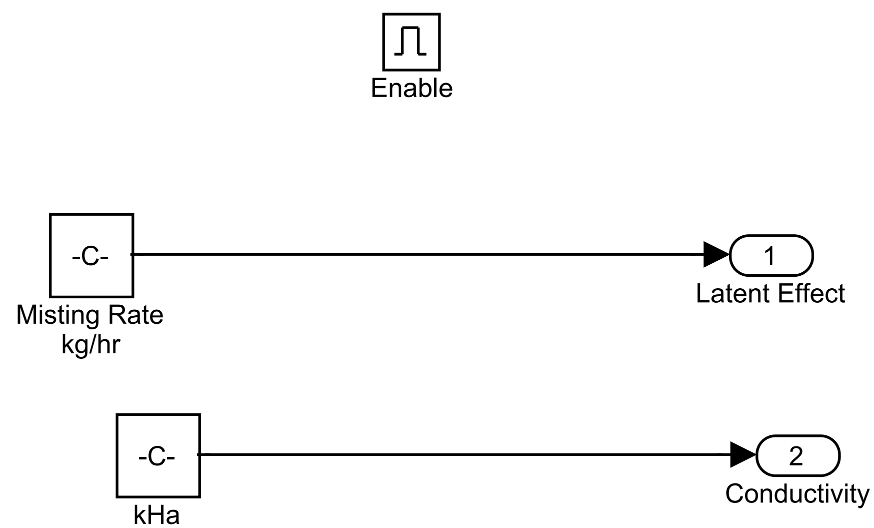
guesssim/Greenhouse Climate Model&Controller/Conduction/Radiative Conductance



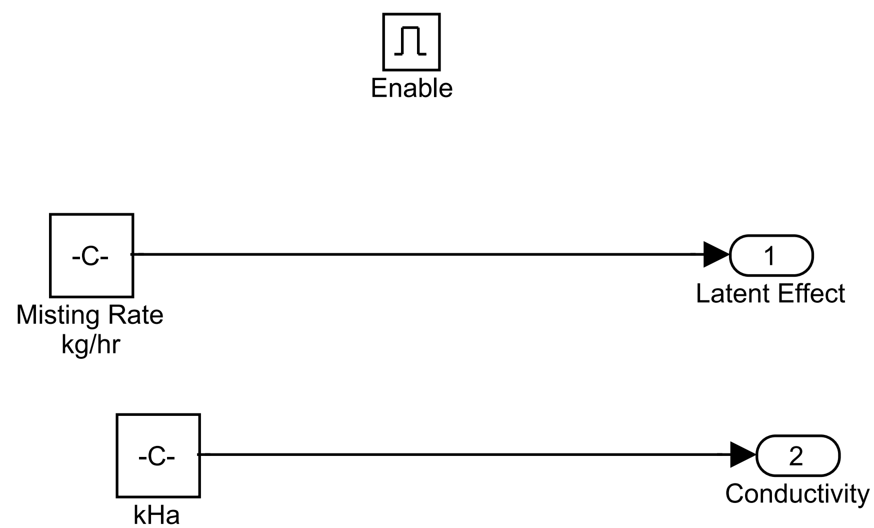
guesssim/Greenhouse Climate Model&Controller/Foggers



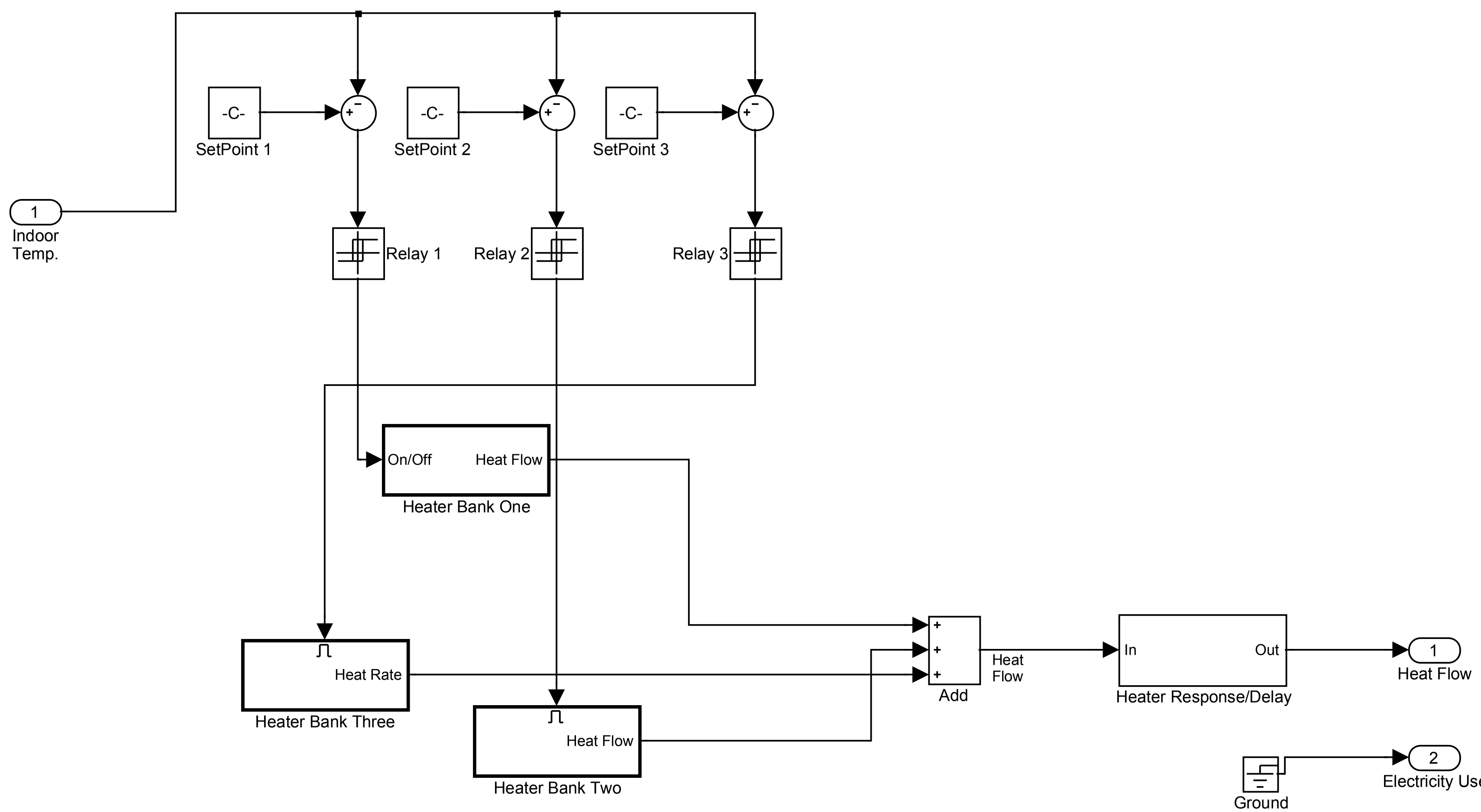
guesssim/Greenhouse Climate Model&Controller/Foggers/Fogger 2



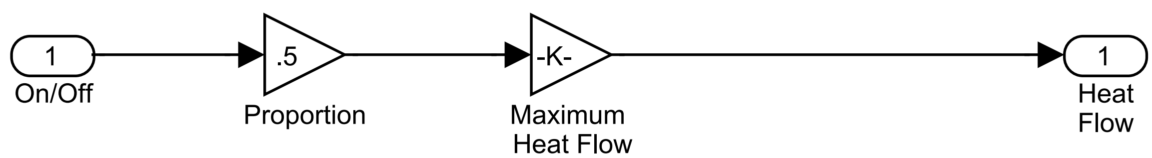
guesssim/Greenhouse Climate Model&Controller/Foggers/Fogger 3



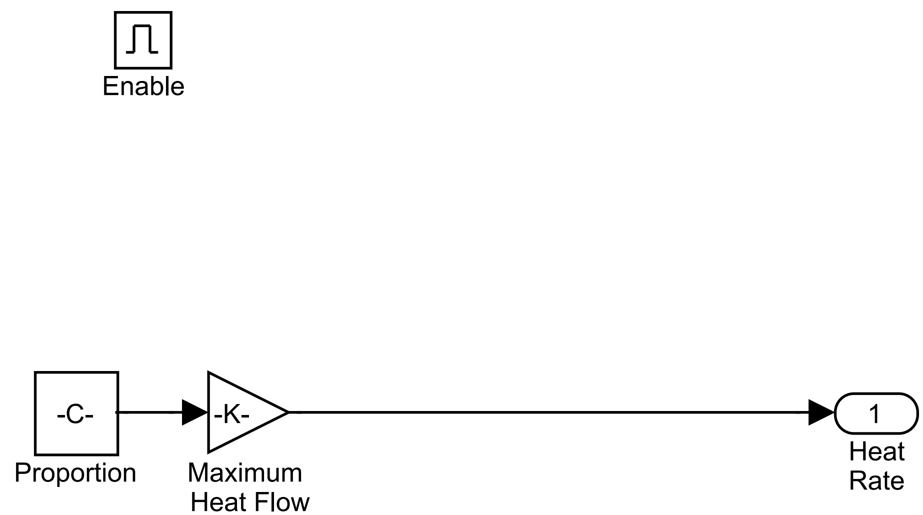
guesssim/Greenhouse Climate Model&Controller/Foggers/Fogger 1



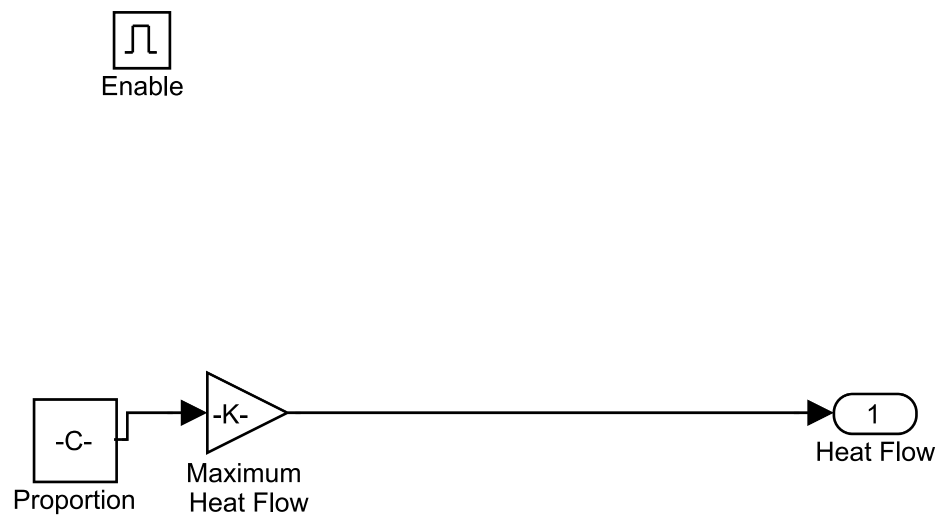
guesssim/Greenhouse Climate Model&Controller/Heaters



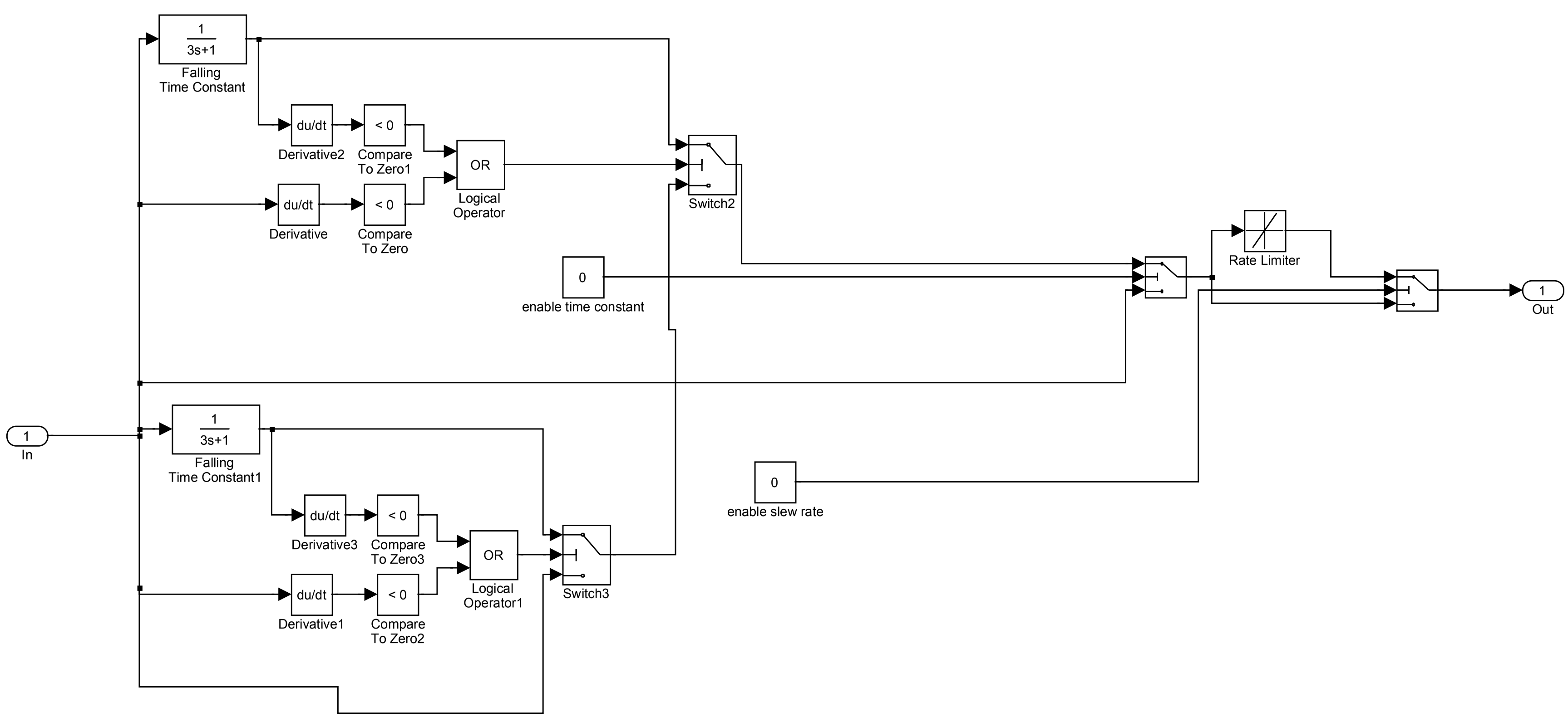
guesssim/Greenhouse Climate Model&Controller/Heaters/Heater Bank One



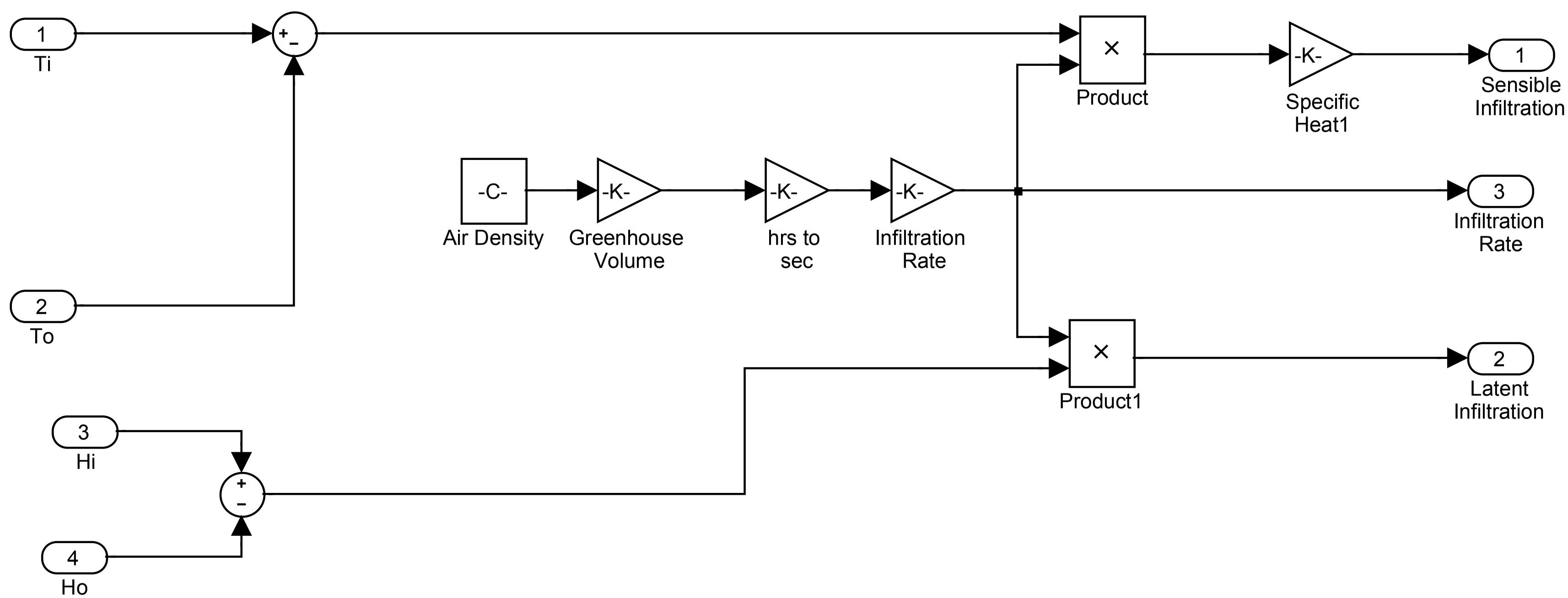
guesssim/Greenhouse Climate Model&Controller/Heaters/Heater Bank Three



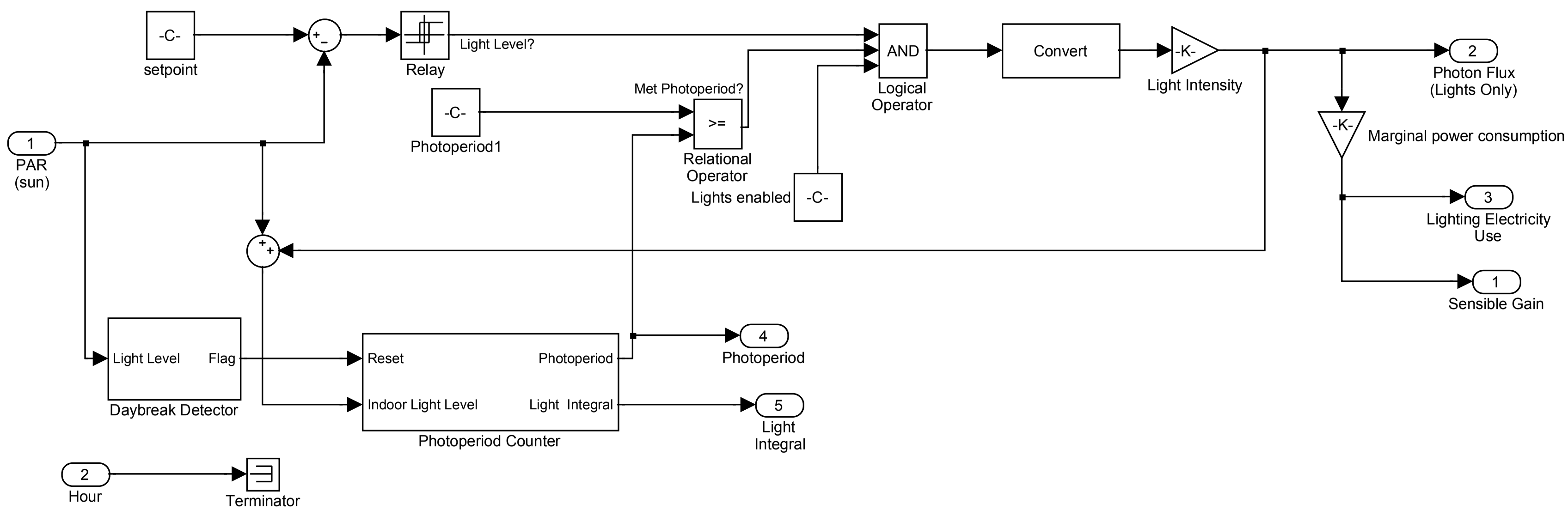
guesssim/Greenhouse Climate Model&Controller/Heaters/Heater Bank Two



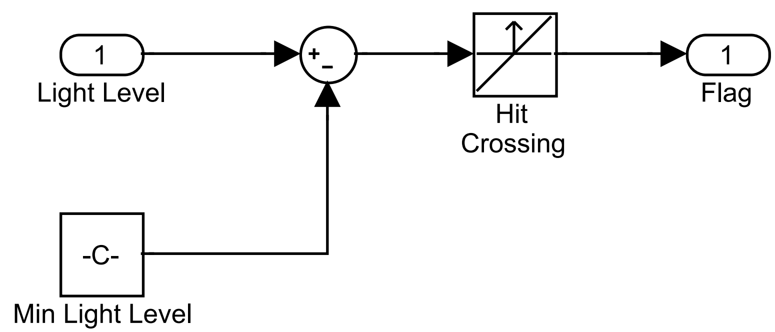
guesssim/Greenhouse Climate Model&Controller/Heaters/Heater Response//Delay



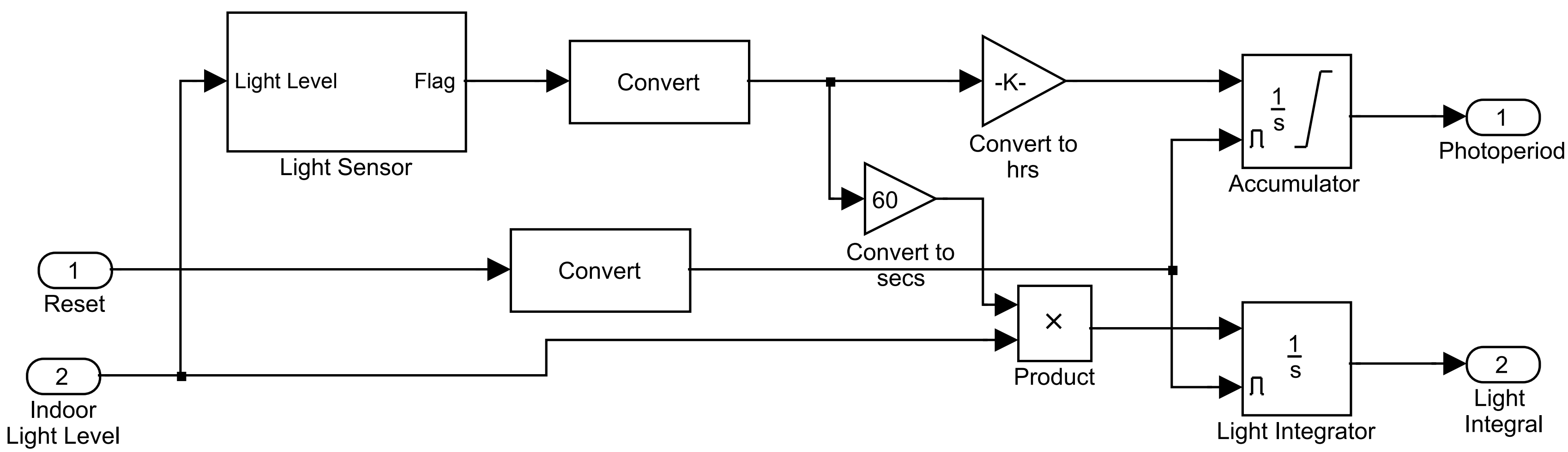
guesssim/Greenhouse Climate Model&Controller/Infiltration



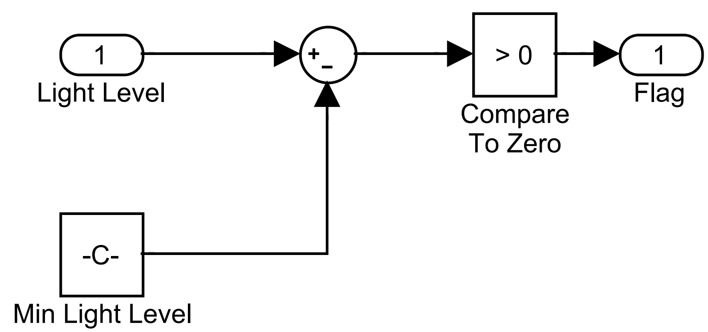
guesssim/Greenhouse Climate Model&Controller/Lighting Model & Control



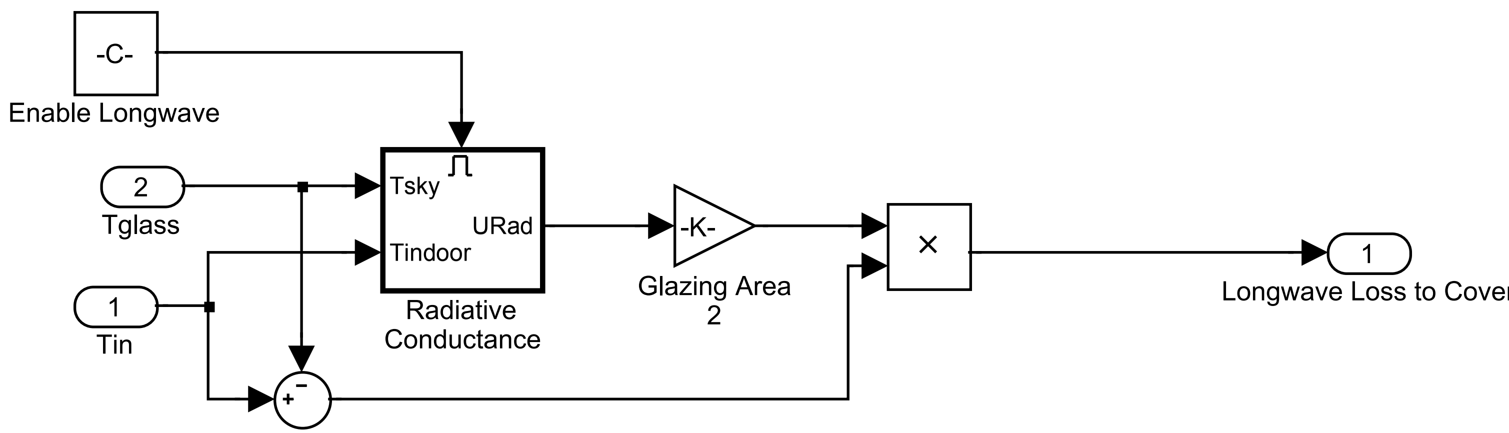
guesssim/Greenhouse Climate Model&Controller/Lighting Model & Control/Daybreak Detector



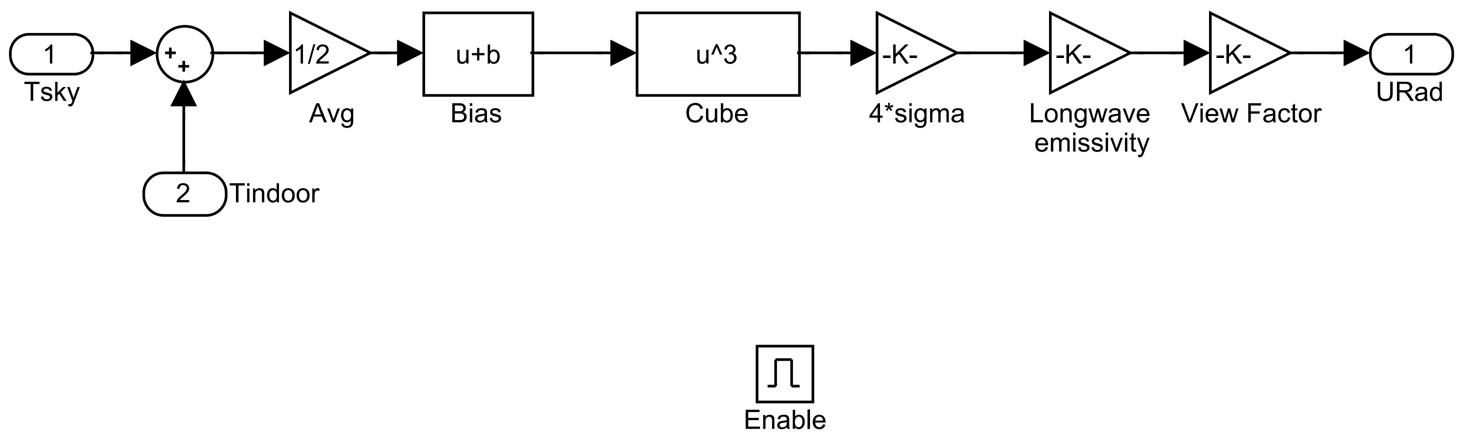
guesssim/Greenhouse Climate Model&Controller/Lighting Model & Control/Photoperiod Counter



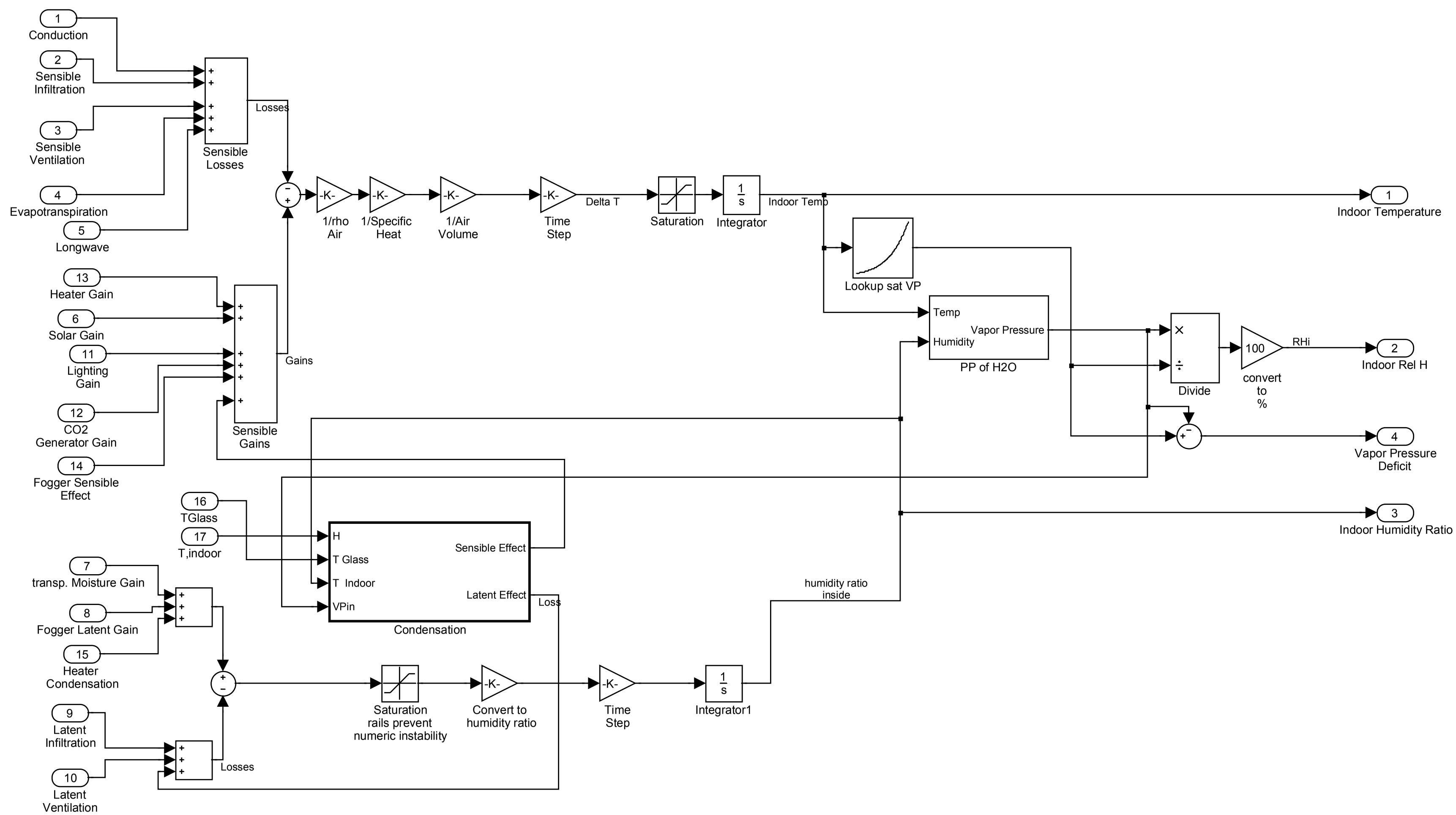
guesssim/Greenhouse Climate Model&Controller/Lighting Model & Control/Photoperiod Counter/Light Sensor



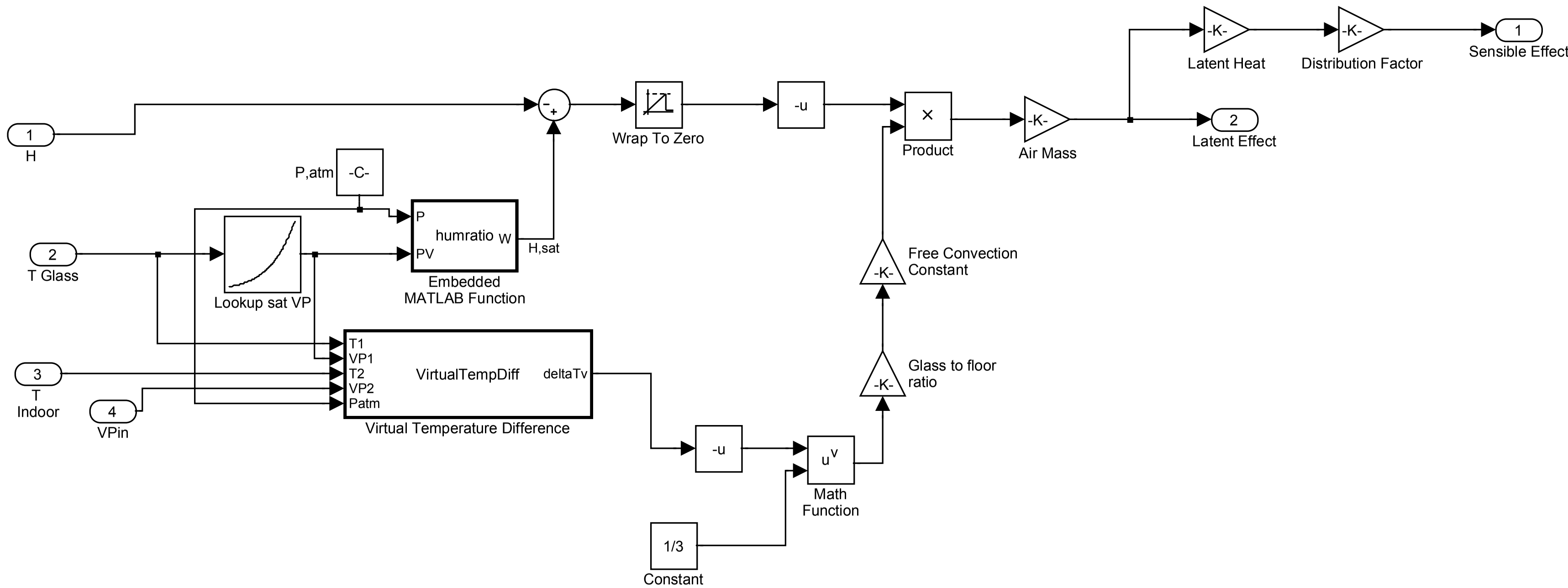
guesssim/Greenhouse Climate Model&Controller/Loss to Cover



guesssim/Greenhouse Climate Model&Controller/Loss to Cover/Radiative Conductance



guesssim/Greenhouse Climate Model&Controller/Temperature& Humidity Mass Balance



guesssim/Greenhouse Climate Model&Controller/Temperature& Humidity Mass Balance/Condensation guesssim/Greenhouse Climate Model&Controller/Temperature& Humidity Mass Balance/Condensation/Embedded MATLAB Function.eML_blk_kernel

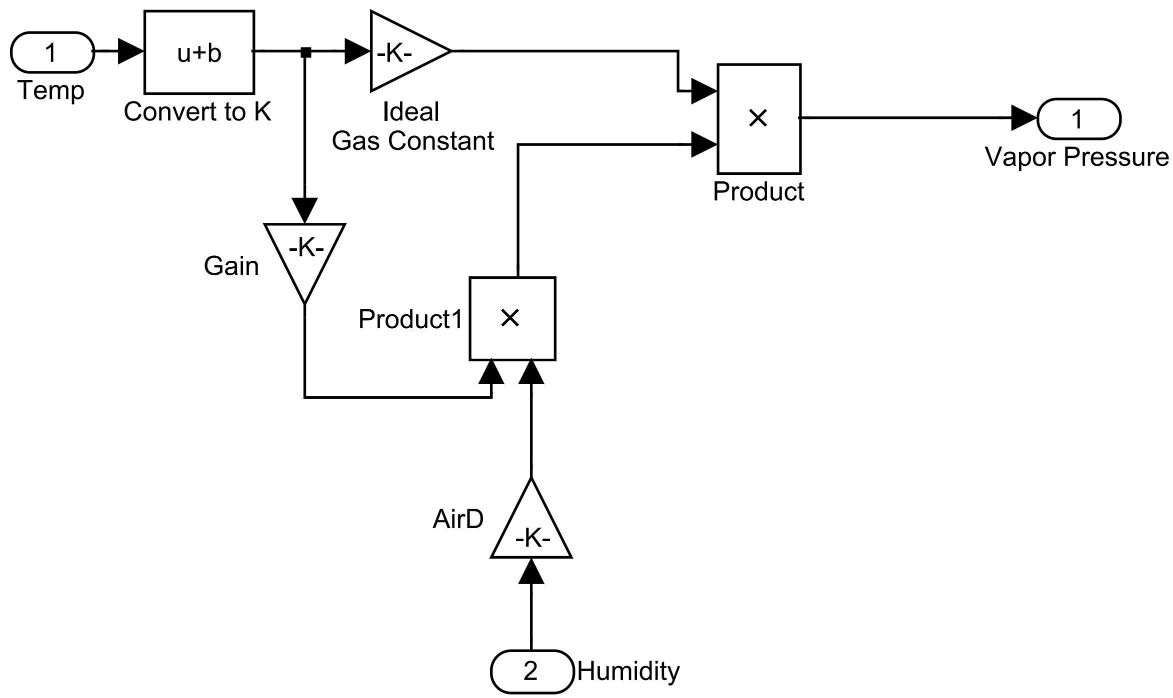
```
% HUMRATIO.m
% Calculates humidity ratio W (kg vapor/kg air)
% PV -- Partial pressure of water vapor
% P --  Atmospheric air pressure in kPa (P)
%
% W = HUMRATIO (P, PV)
function W  = humratio (P, PV)
% if (PV > P) | ((PV < 0) | (P < 0))
%    P
%    PV
%    error ('PSYCH03: FATAL Illegal or negative pressure values'
% end
W = 0.62198 * PV ./(P - PV);
return
```

%<fullsystem>

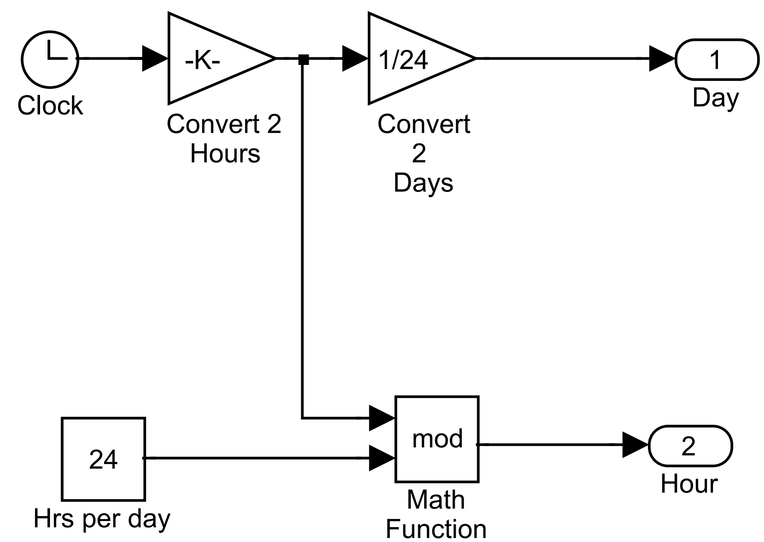
guesssim/Greenhouse Climate Model&Controller/Temperature& Humidity Mass Balance/Condensation/Virtual Temperature Difference.eML_blk_kernel

```
% Calculates the difference in virtual temperature between  T1 & T2
% Source: Monteith & Unsworth

function deltaTv = VirtualTempDiff(T1, VP1, T2, VP2, Patm)
% This block supports an embeddable subset of the MATLAB language
% See the help menu for details.

deltaTv = T1 - T2 + 0.38 * (VP1*T1 - VP2*T2)/Patm;
```

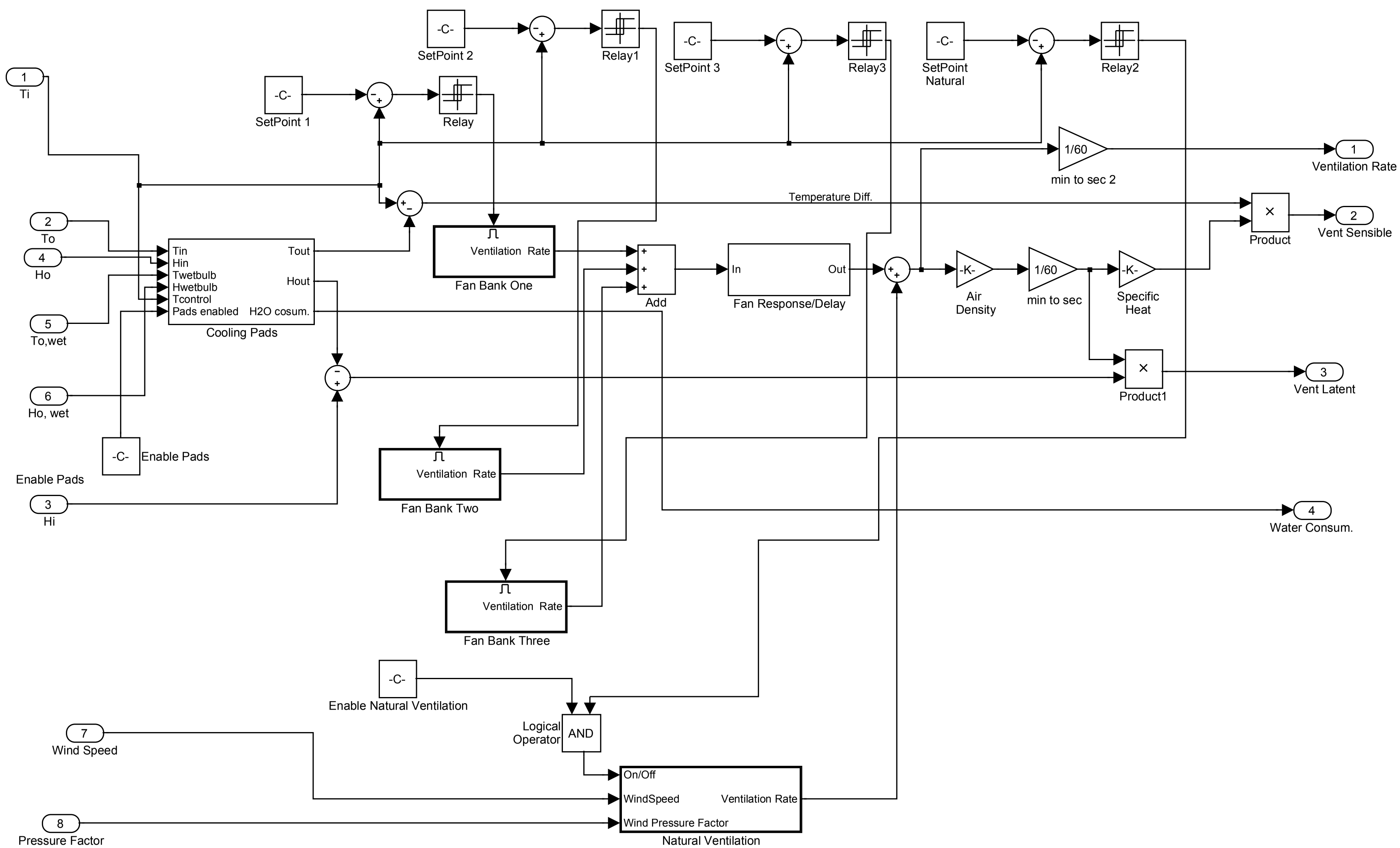
%<fullsystem>



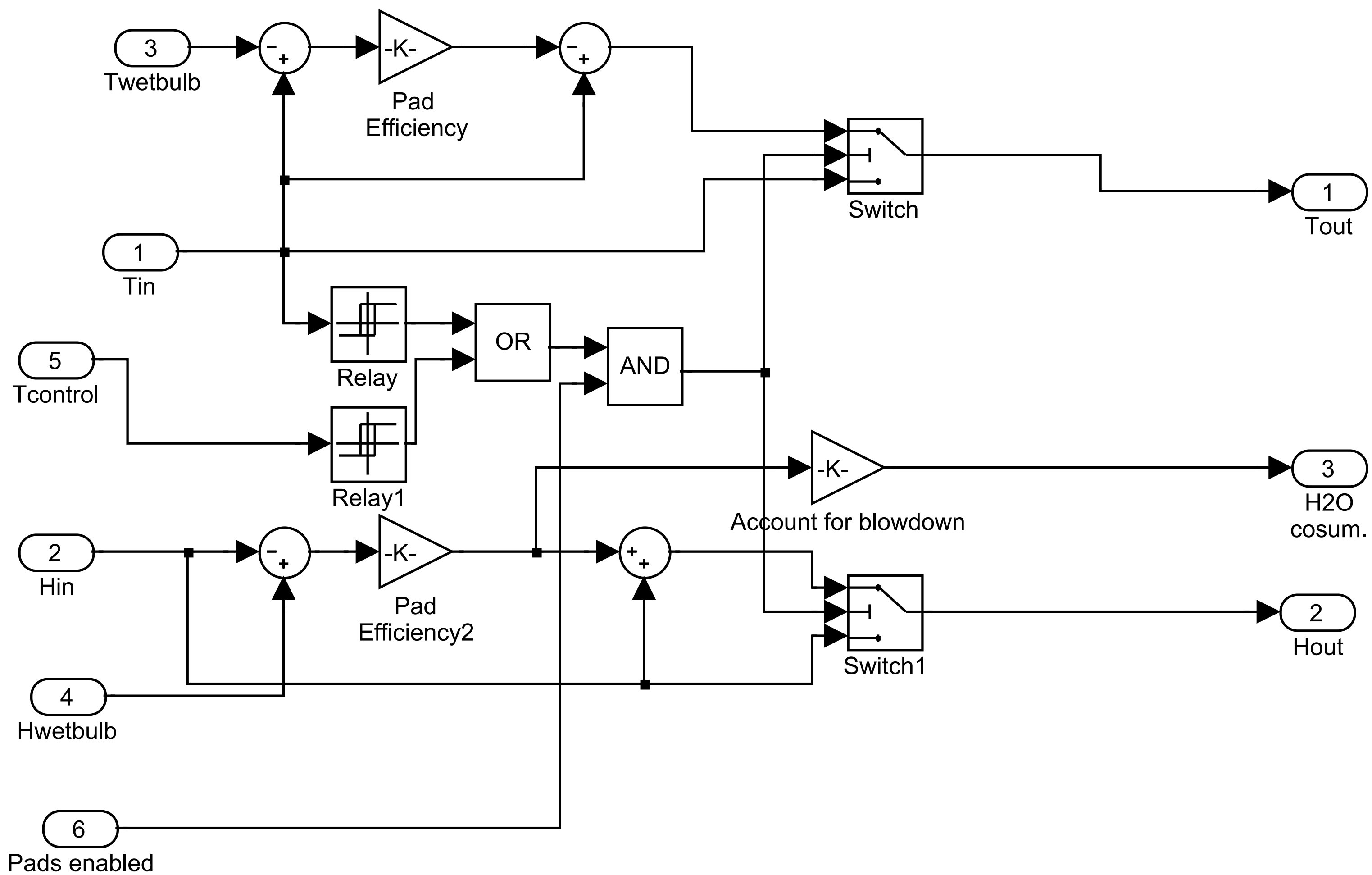
guesssim/Greenhouse Climate Model&Controller/Temperature& Humidity Mass Balance/PP of H2O



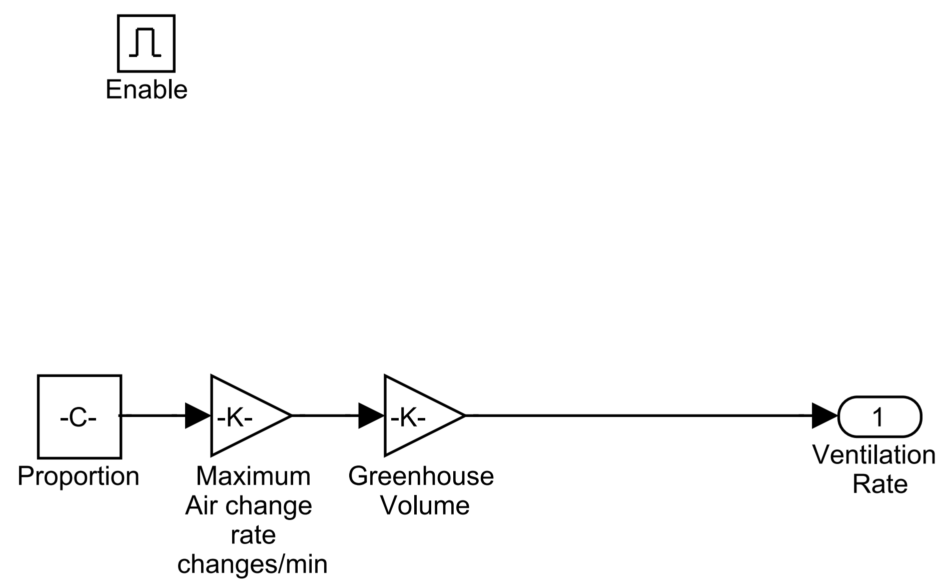
guesssim/Greenhouse Climate Model&Controller/Time



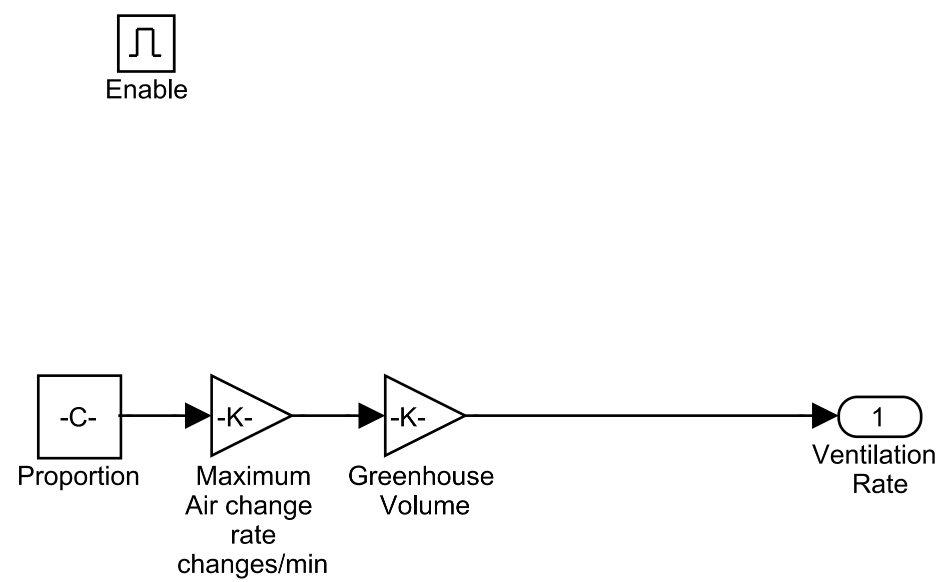
guesssim/Greenhouse Climate Model&Controller/Ventilation&Cooling Pads



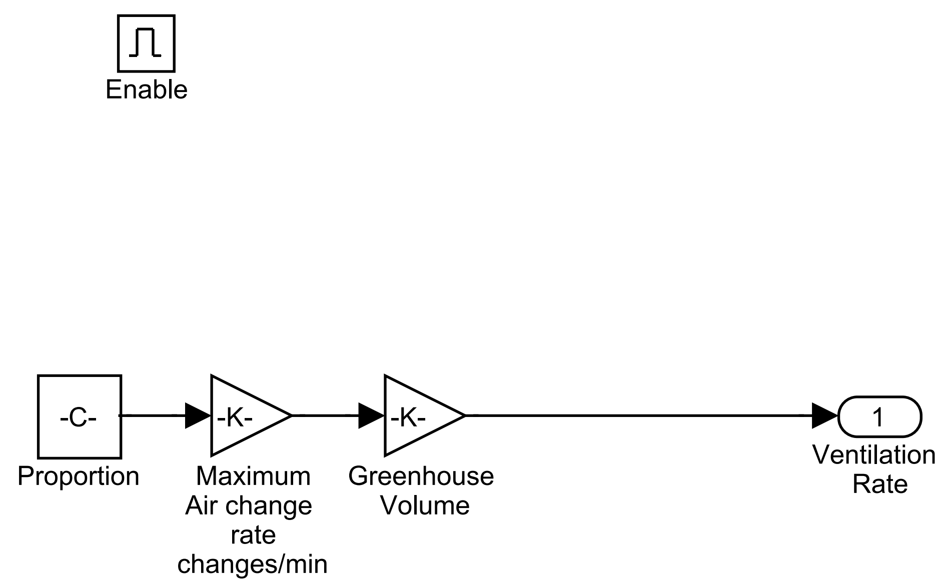
guesssim/Greenhouse Climate Model&Controller/Ventilation&Cooling Pads/Cooling Pads



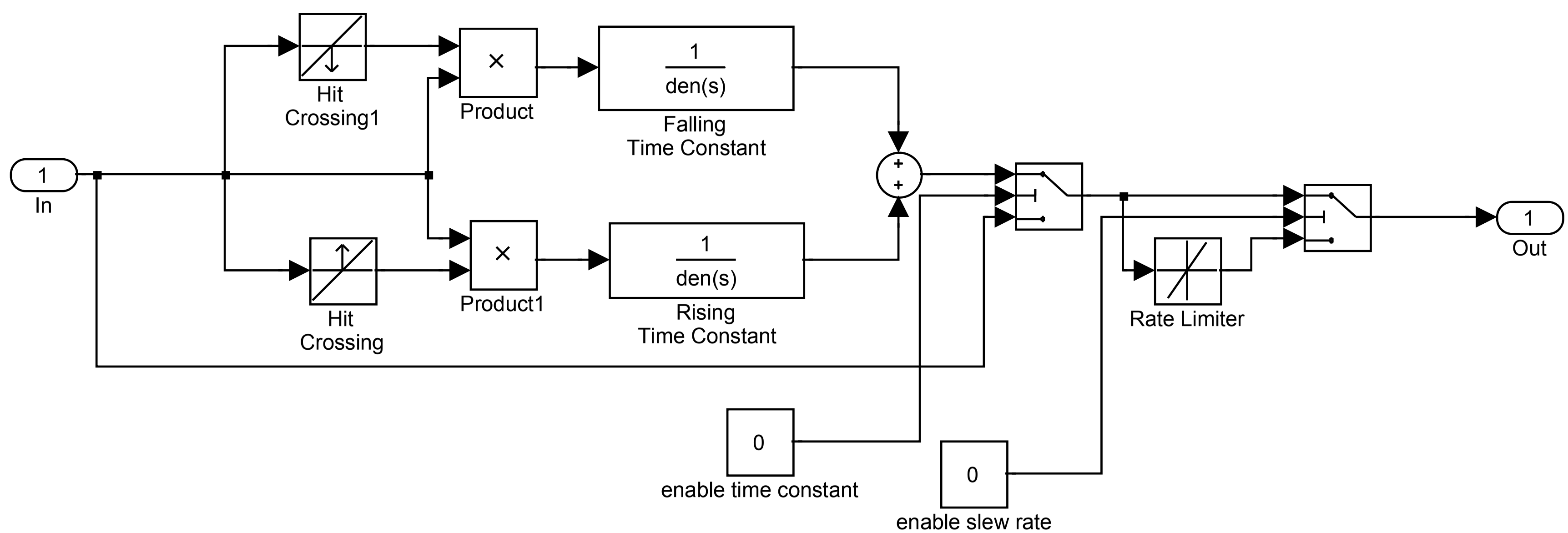
guesssim/Greenhouse Climate Model&Controller/Ventilation&Cooling Pads/Fan Bank One



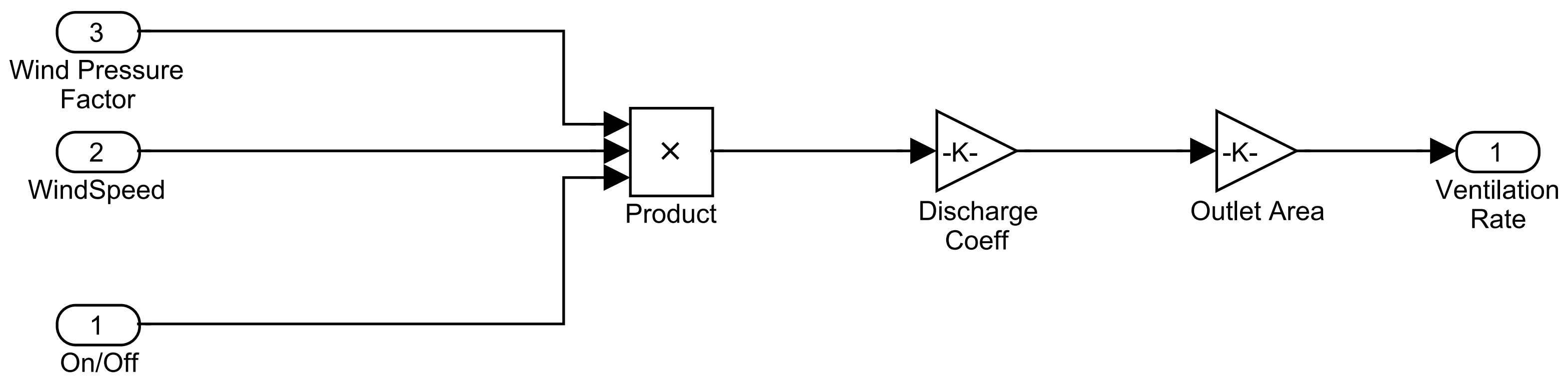
guesssim/Greenhouse Climate Model&Controller/Ventilation&Cooling Pads/Fan Bank Three



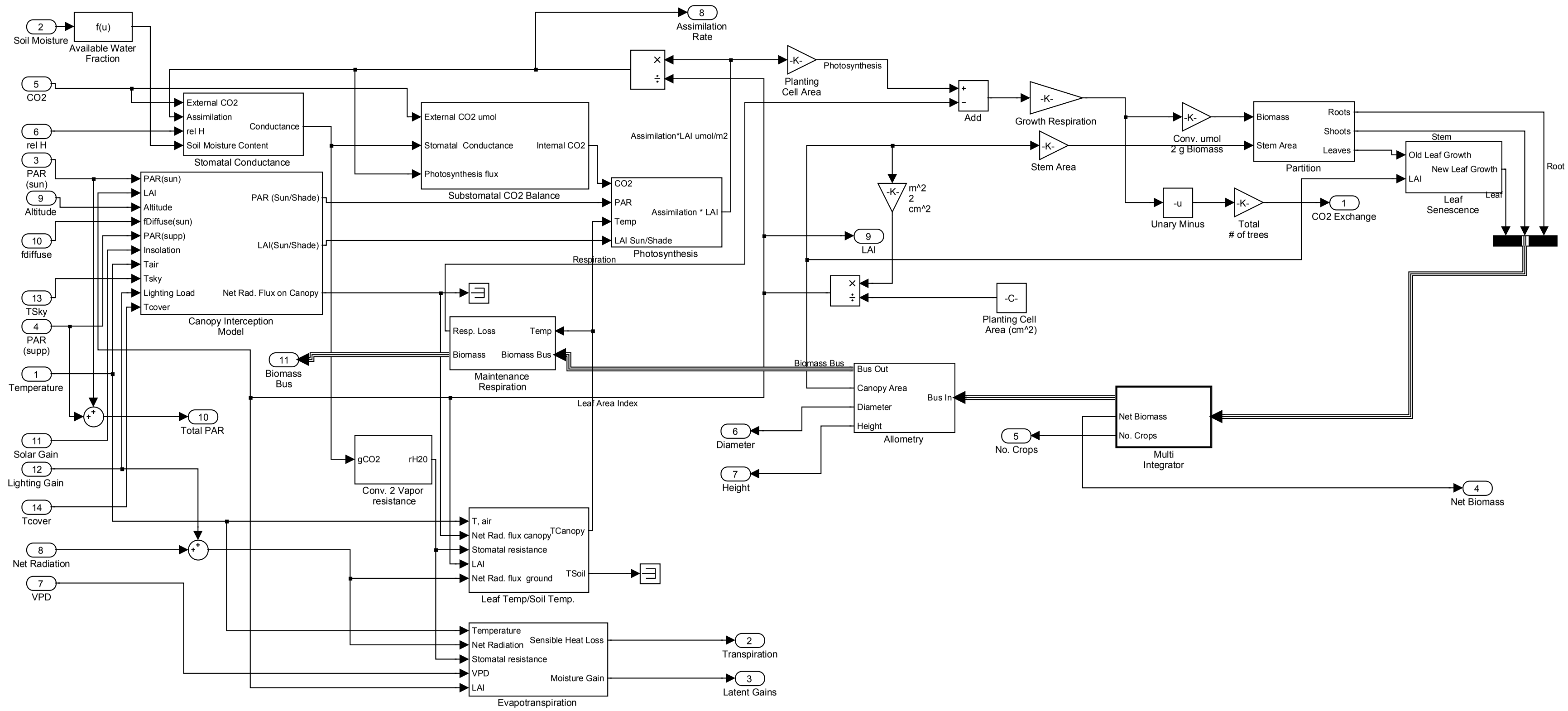
guesssim/Greenhouse Climate Model&Controller/Ventilation&Cooling Pads/Fan Bank Two



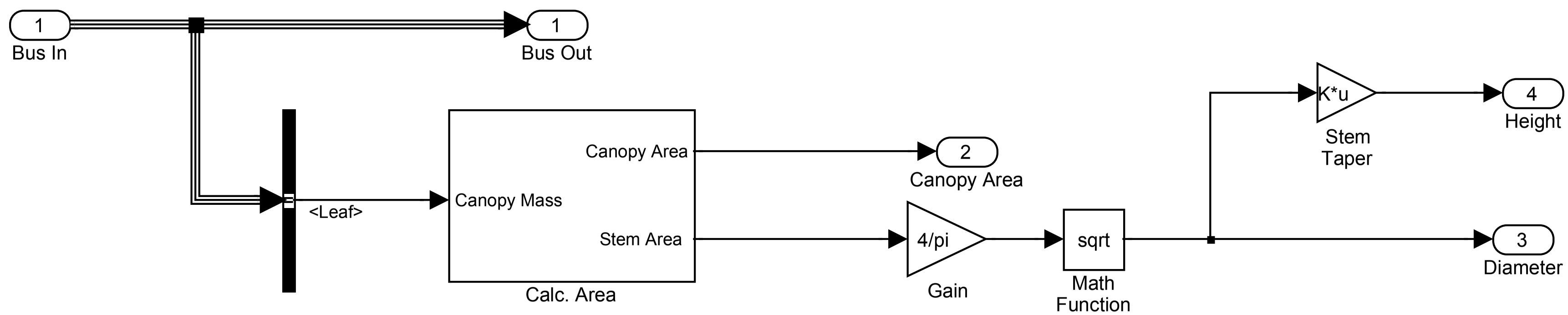
guesssim/Greenhouse Climate Model&Controller/Ventilation&Cooling Pads/Fan Response//Delay



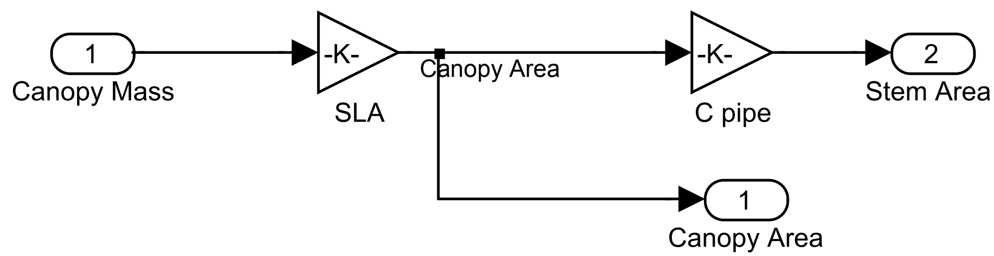
guesssim/Greenhouse Climate Model&Controller/Ventilation&Cooling Pads/Natural Ventilation



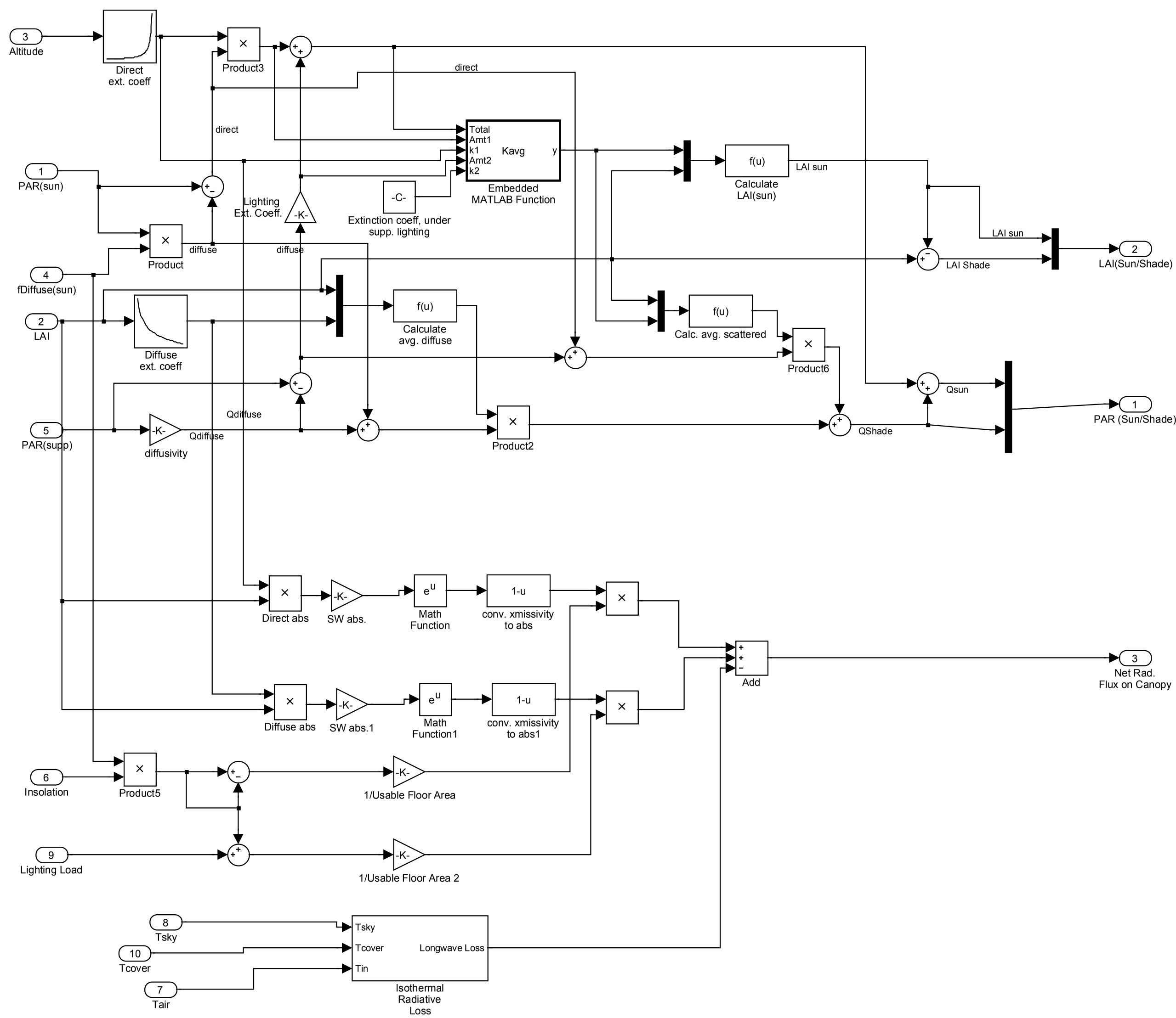
guesssim/Plant Model



guesssim/Plant Model/Allometry



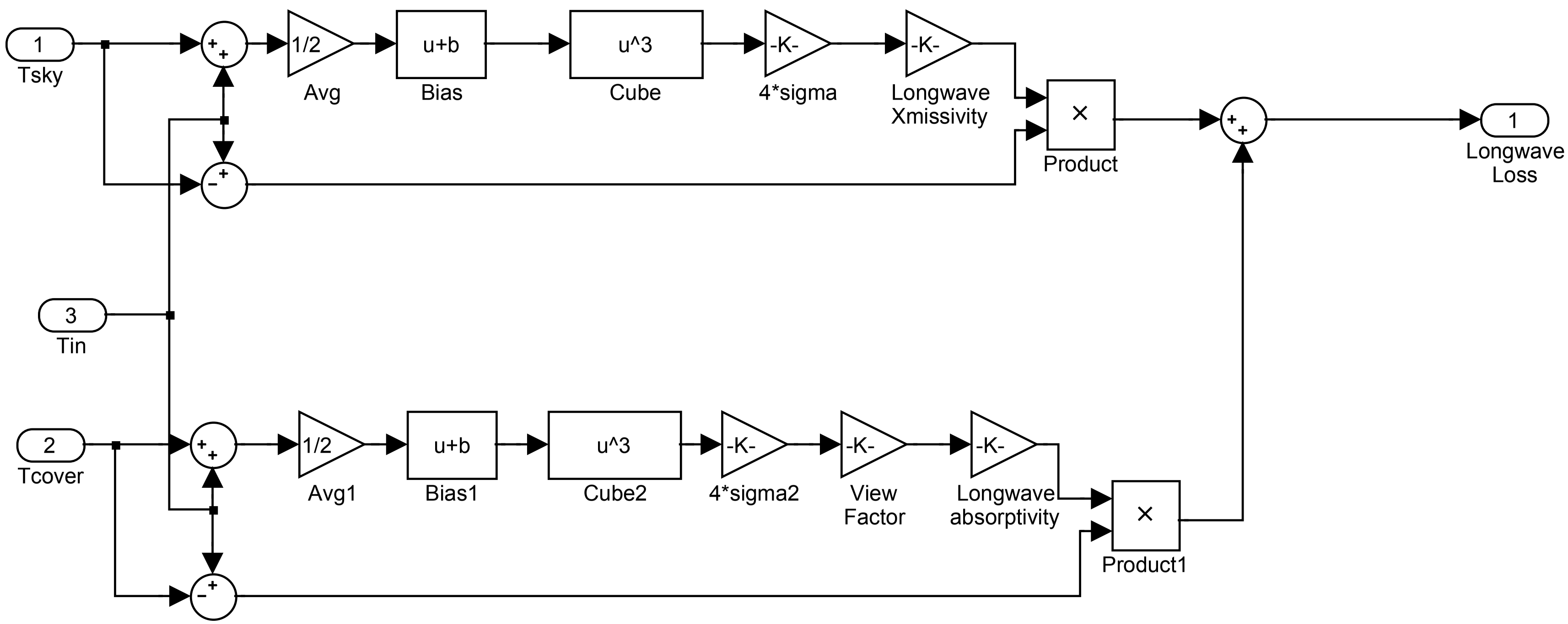
guesssim/Plant Model/Allometry/Calc. Area



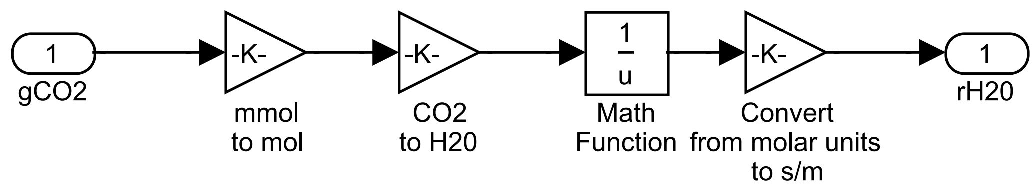
guesssim/Plant Model/Canopy Interception Model guesssim/Plant Model/Canopy Interception Model/Embedded MATLAB Function.eML_blk_kernel

```
function y = Kavg(Total,Amt1,k1,Amt2,k2)
% This block supports an embeddable subset of the MATLAB language
% See the help menu for details.
if Amt2 == 0
    y = k1;
elseif (Amt1 + Amt2) == 0
    y = k1;
else
    y = (k1 *Amt1 + k2*Amt2)/(Amt1 + Amt2);
end
```

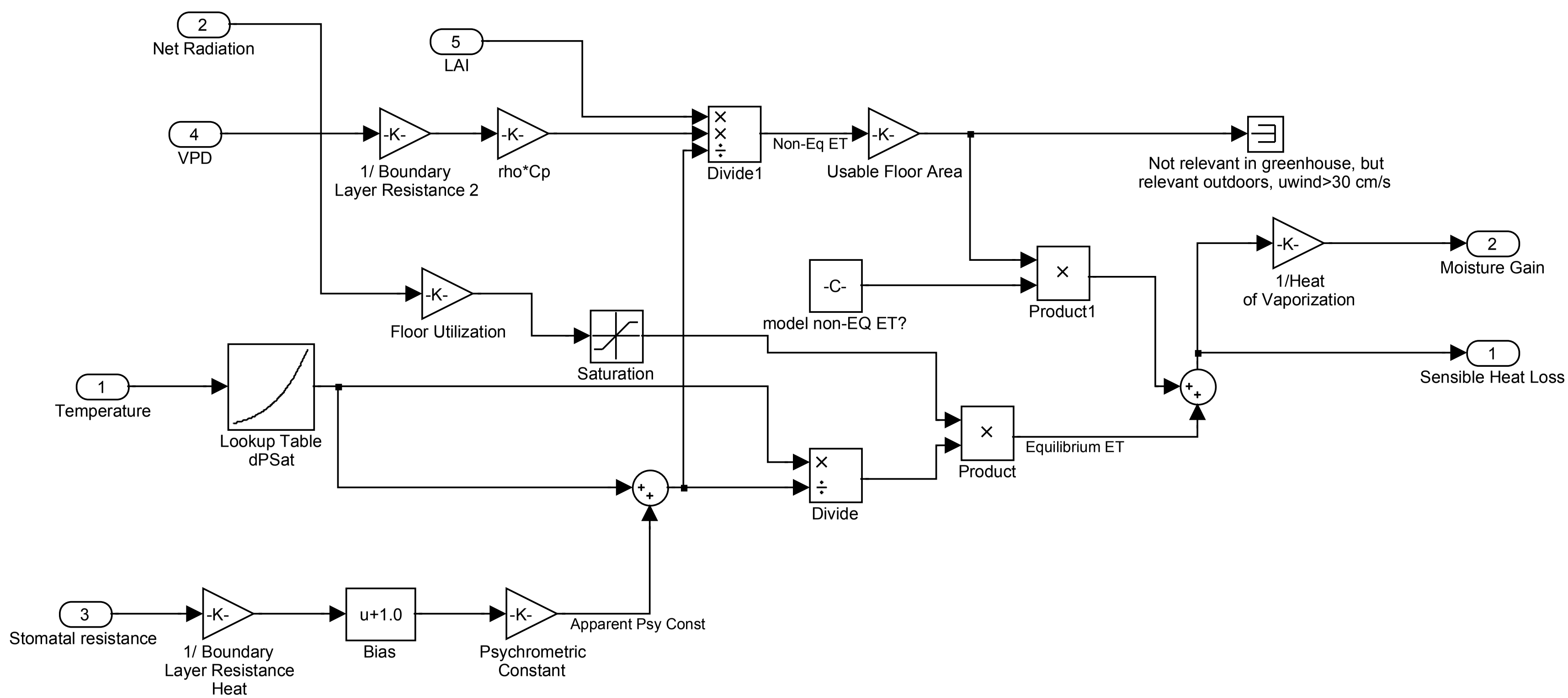
%<fullsystem>



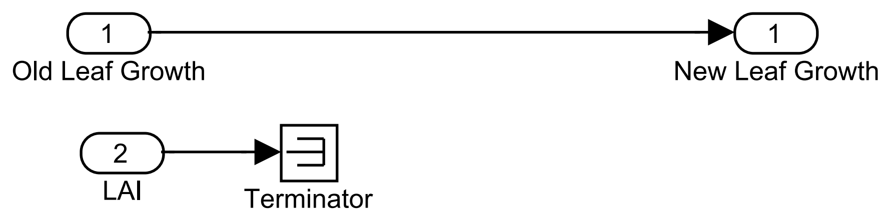
guesssim/Plant Model/Canopy Interception Model/Isothermal Radiative Loss



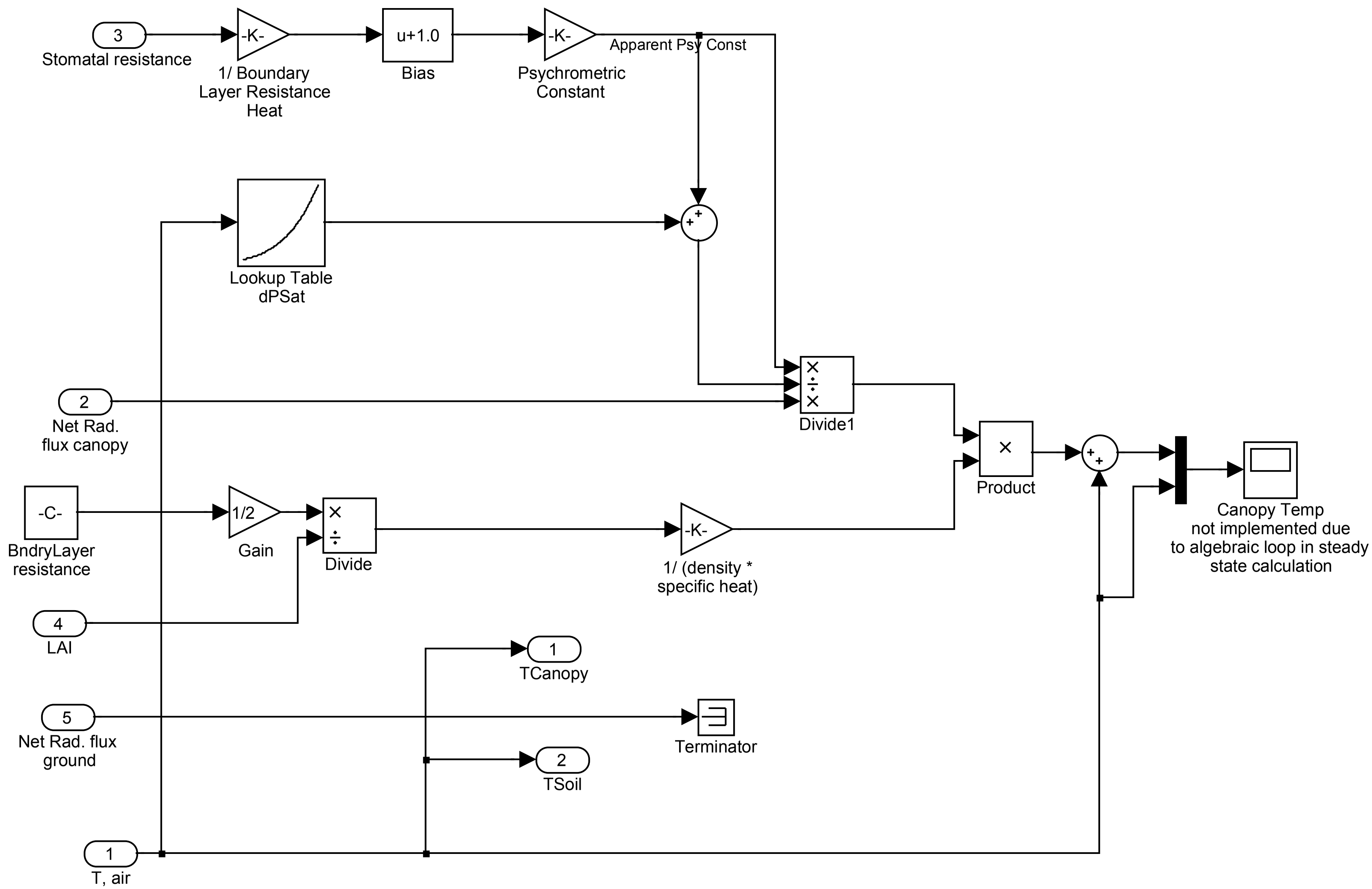
guesssim/Plant Model/Conv. 2 Vapor resistance



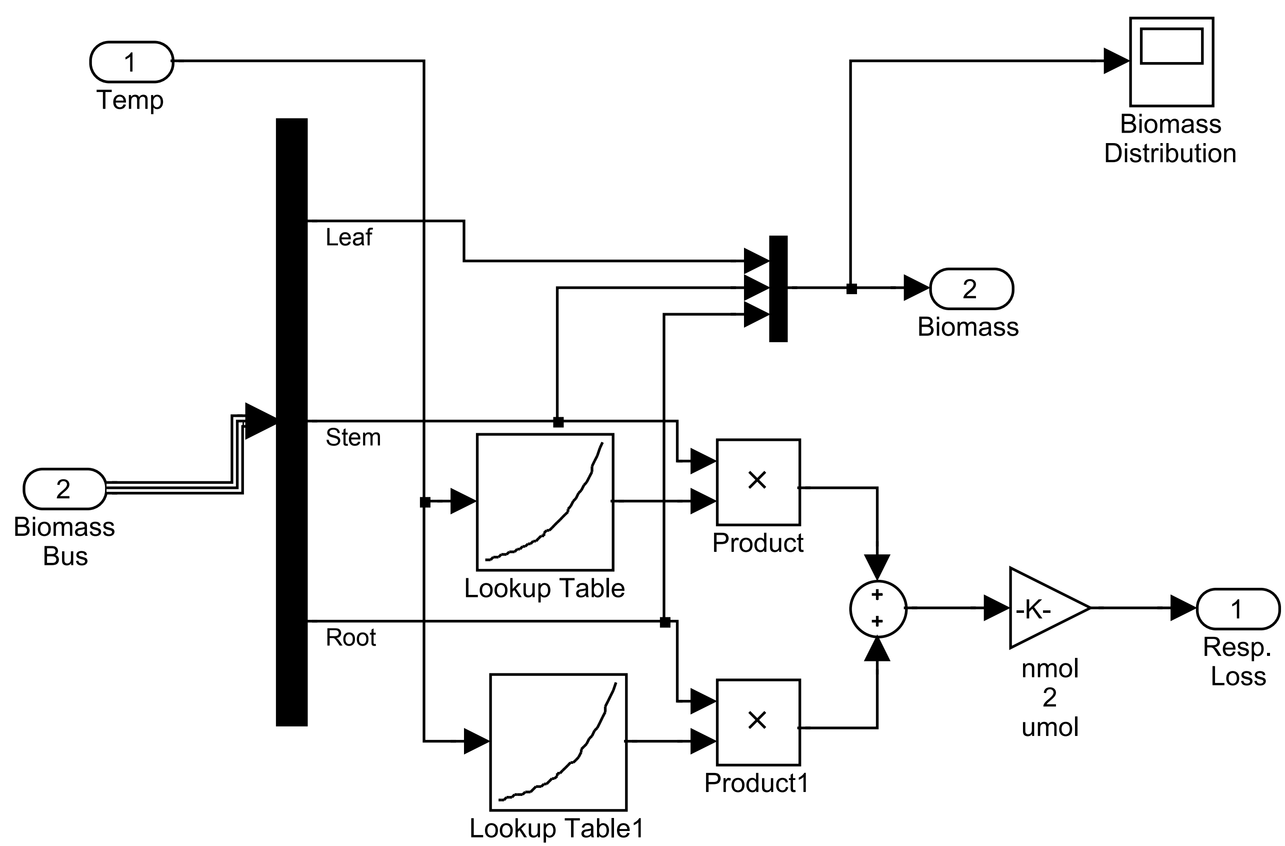
guesssim/Plant Model/Evapotranspiration



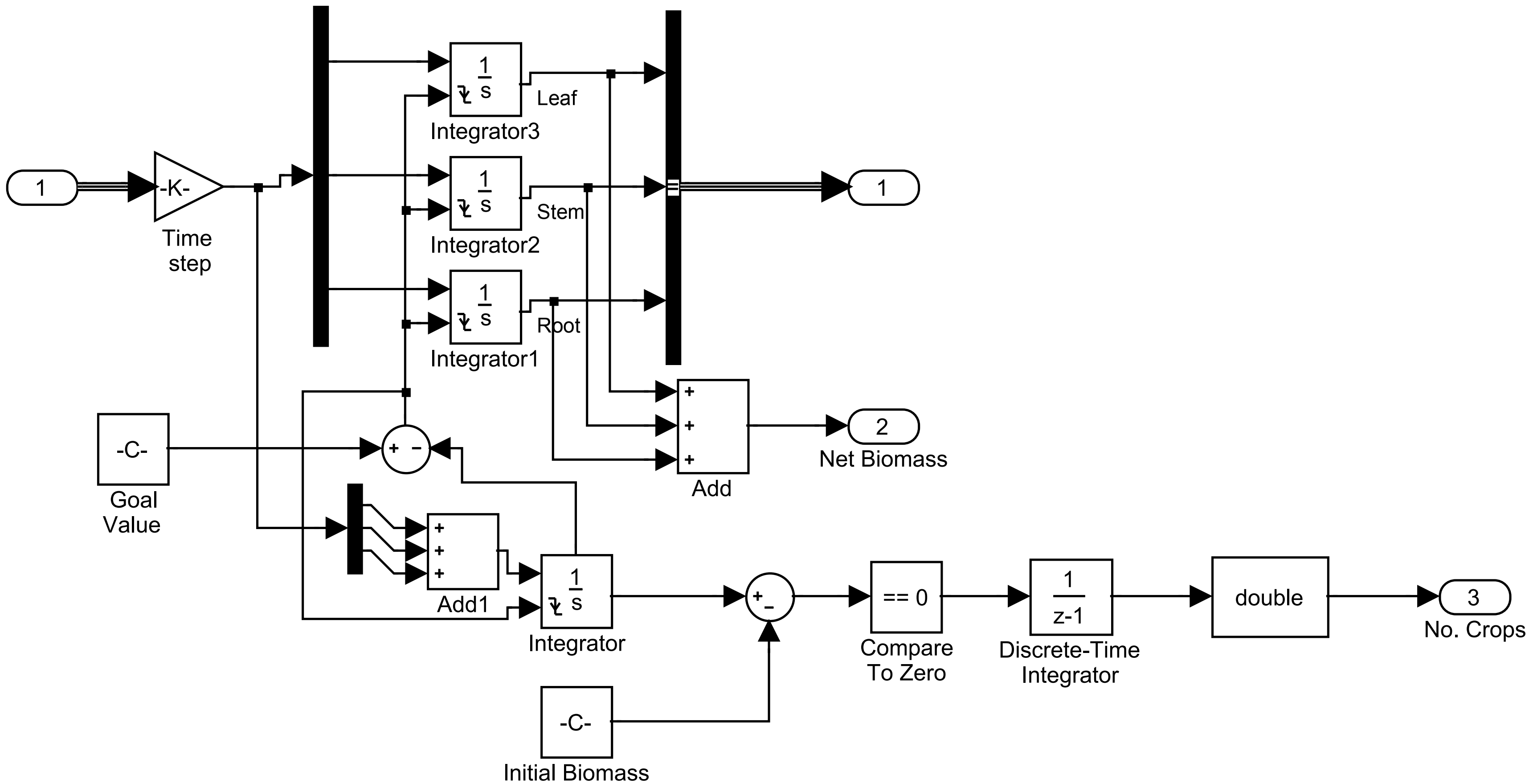
guesssim/Plant Model/Leaf Senescence



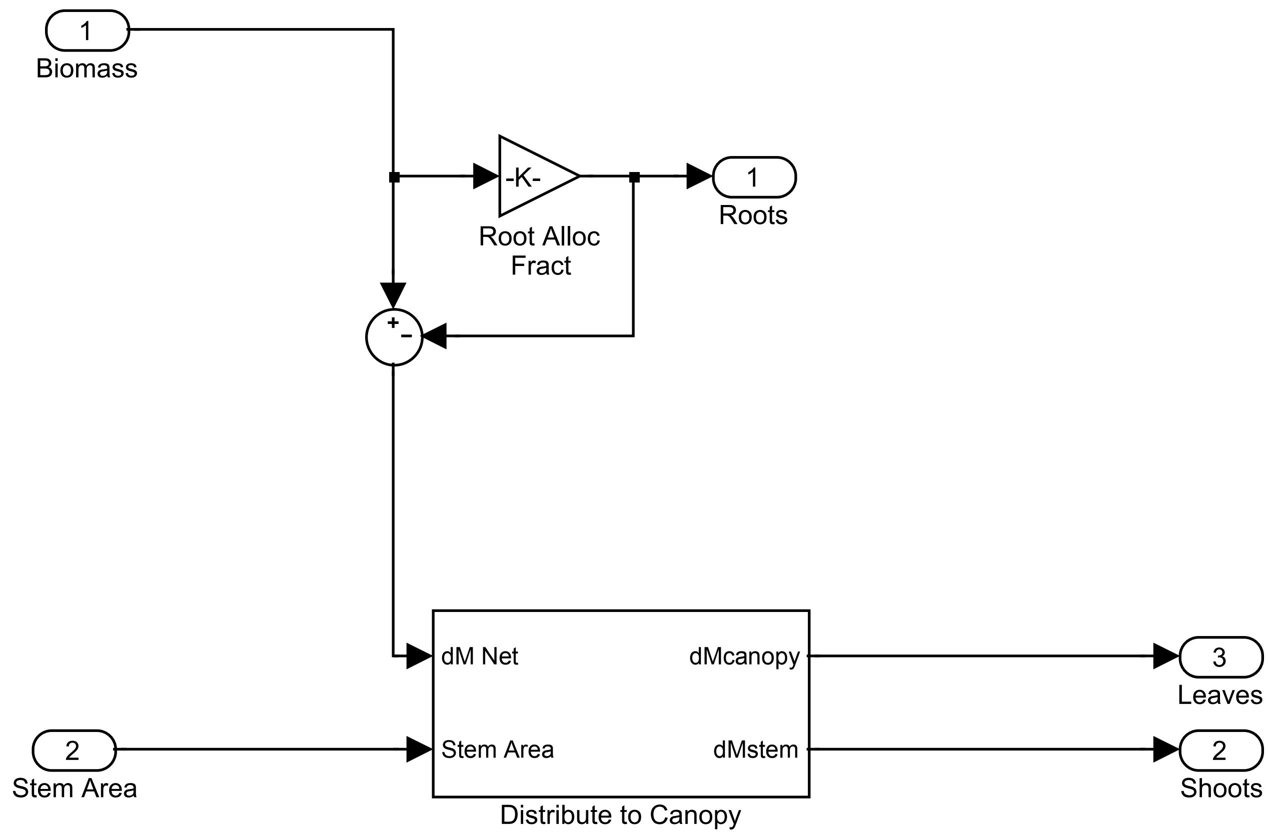
guesssim/Plant Model/Leaf Temp//Soil Temp.



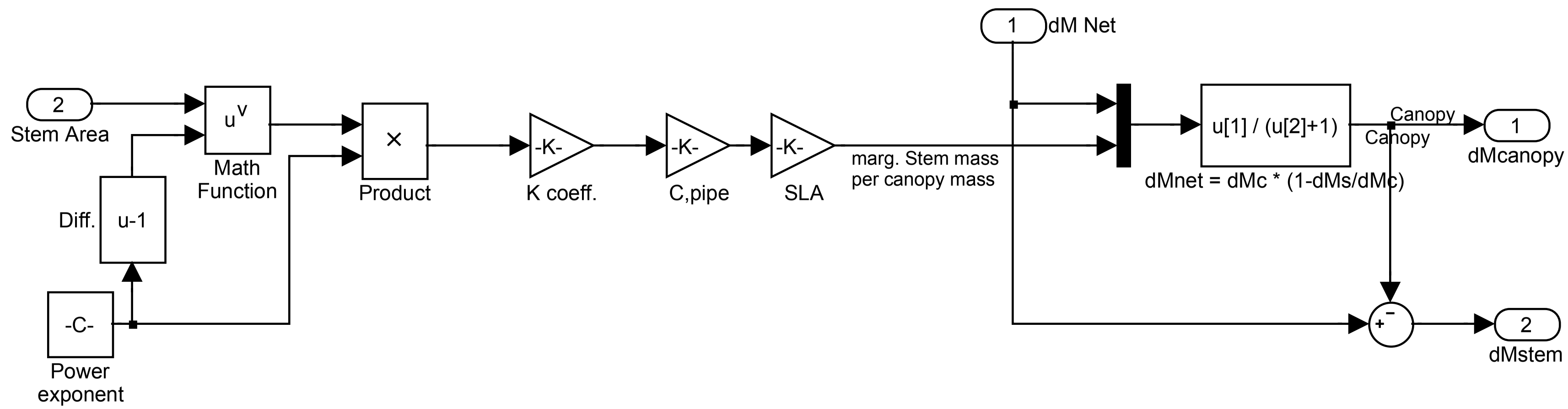
guesssim/Plant Model/Maintenance Respiration



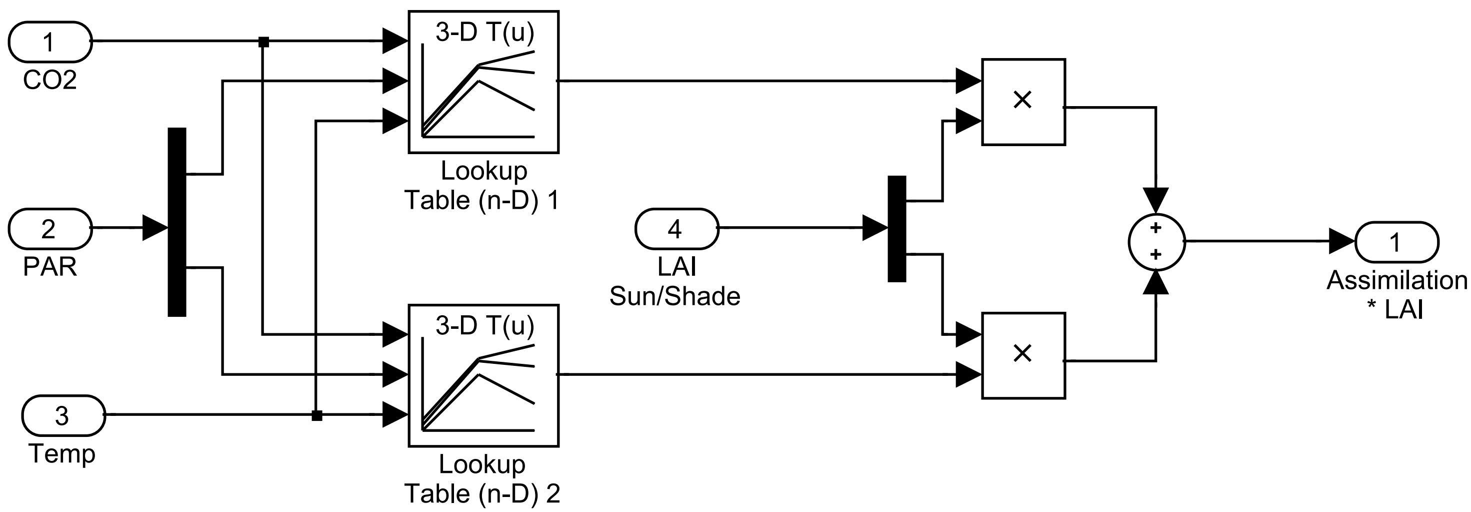
guesssim/Plant Model/Multi Integrator



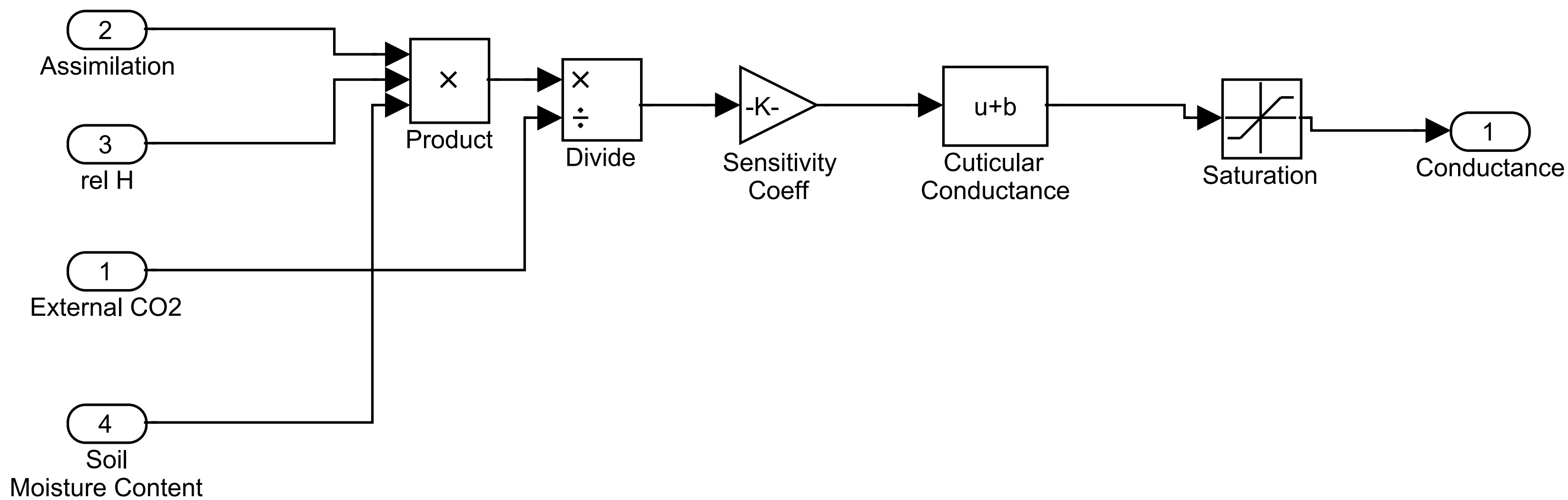
guesssim/Plant Model/Partition



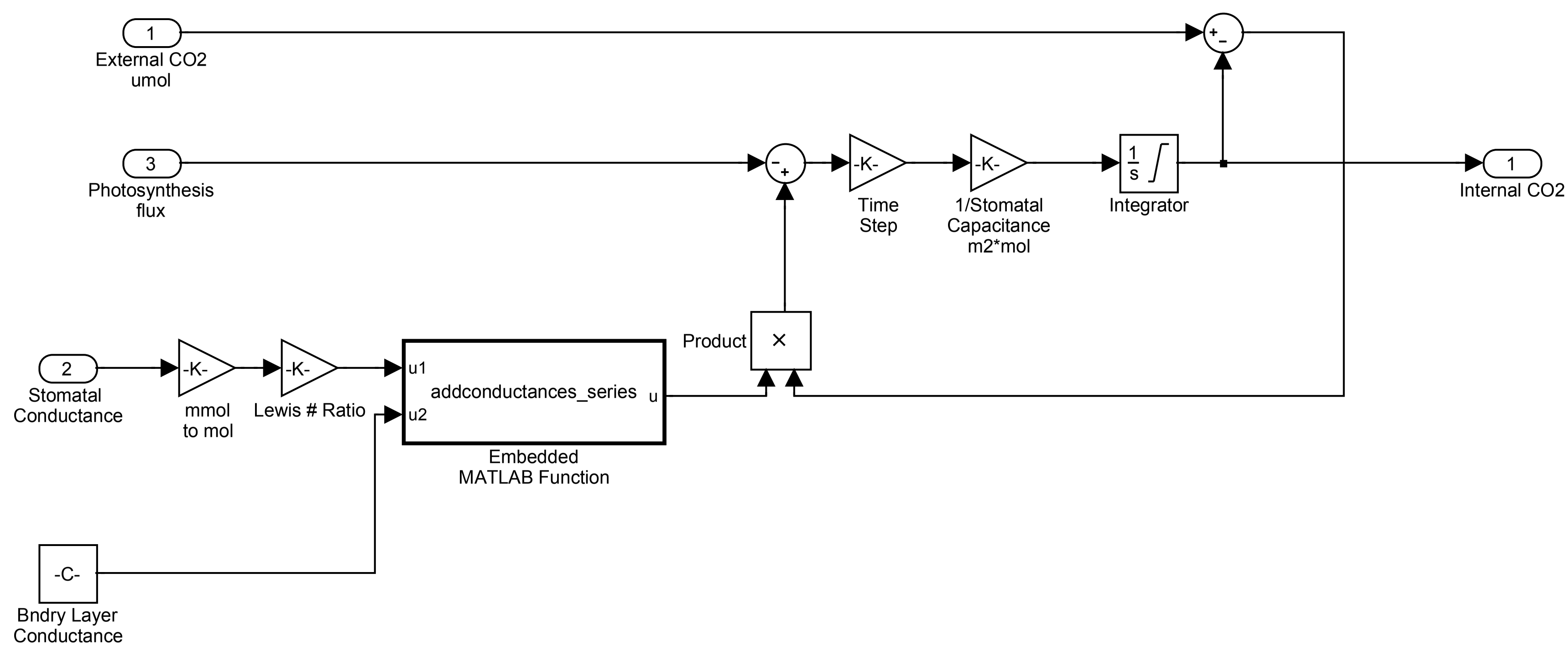
guesssim/Plant Model/Partition/Distribute to Canopy



guesssim/Plant Model/Photosynthesis



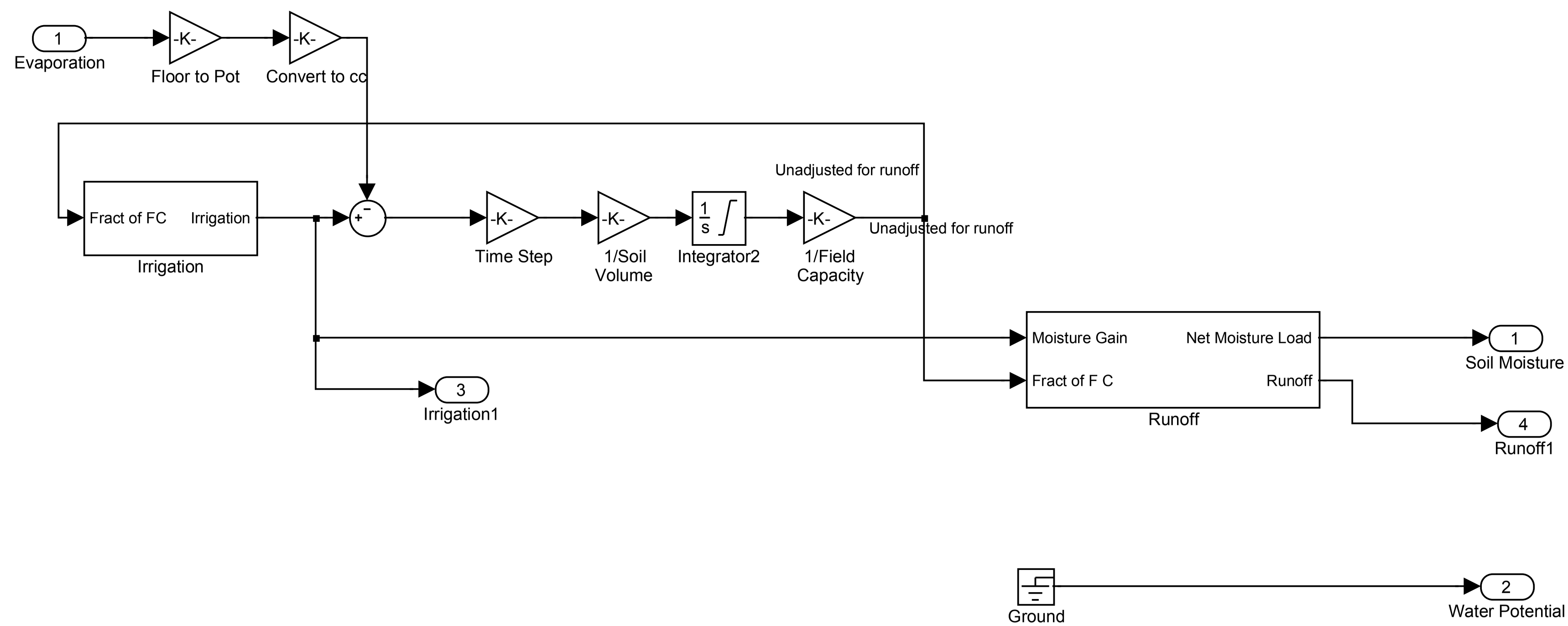
guesssim/Plant Model/Stomatal Conductance



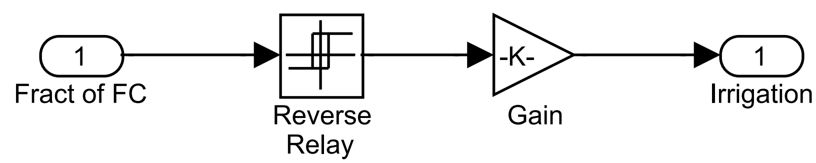
guesssim/Plant Model/Substomatal CO2 Balance guesssim/Plant Model/Substomatal CO2 Balance/Embedded MATLAB Function.eML_blk_kernel

```
function u = addconductances_series(u1, u2)
% This block supports an embeddable subset of the MATLAB language
% See the help menu for details.
% This function adds series conductances
% U = 1/(1/U1 + 1/U2)

u = 1/(1/u1+1/u2);
```

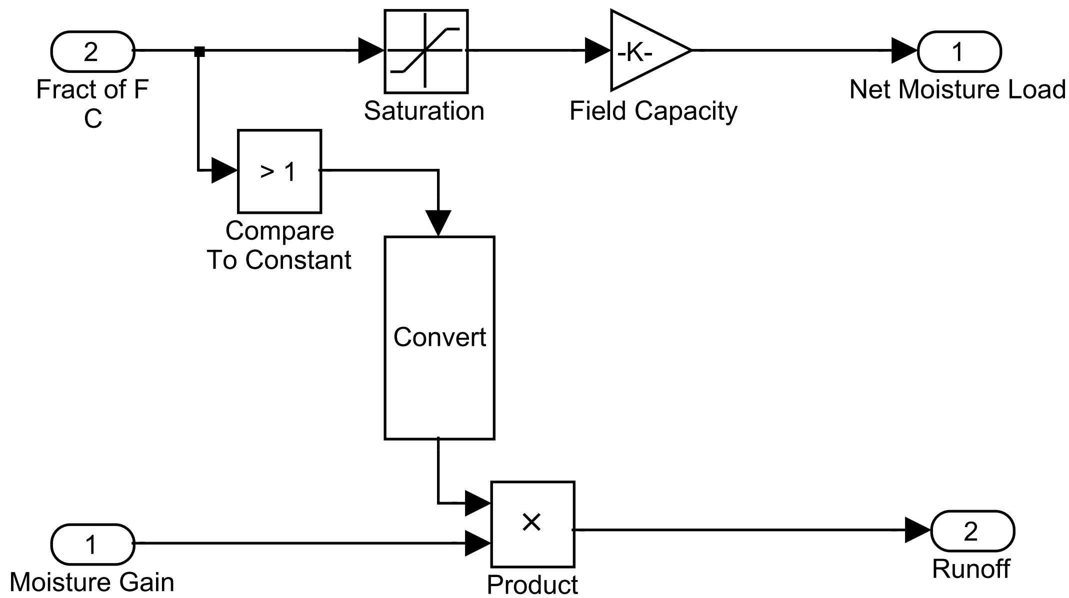
%<fullsystem>



guesssim/Soil Model



guesssim/Soil Model/Irrigation



guesssim/Soil Model/Runoff

GUESS MODEL MATLAB SOURCE CODE

MAIN MODEL

```
% GUESS.m
% Core routine GUESS model
% GUESS (Greenhouse Use of Energy Seedling Simulation)
% GUESS is a dynamic lumped parameter process based model of a Douglas Fir
% seedling production greenhouse.  GUESS models the dynamics of
% photosynthesis, and carbon allocation, climate control, and energy use.

t1 = cputime;
guessinit;              % User Defined Parameters
guessread;              % Load, and process weather data
guessmodel;             % Execute Simulink model
guessoutput;            % Display results

t2 = cputime;
t3 = t2-t1;
disp(sprintf('Model took %4.2f seconds to execute', t3))
```

```
% guessinit.m
% Initializes GUESS Model
% Creates and initializes variables and structures used in the simulation.
% Called before guessGUI to set up program defaults.
% guessinit is the interface for all user configurable parameters
% See GUESS documentation for more information.


%
% ----------- File Data -------------------
% Input File Path

% Simulation Settings

% -------input file parameters -------------
Settings.sim.interpmethod = 'cubic';
Settings.file.filepath = ['C:\Program Files\MATLAB71\work\M.Eng Project\'...
 'Data Sets\Agrimet Hourly Data Corvallis 2004_final_version.csv'];
Settings.file.startrow = 5;
Settings.file.startcol = 2;
Settings.file.delimiter = ',';          % datafile delimeter
Settings.sim.frequency = 'hourly';      % sampling frequency: hourly, 15min, or 1min
Settings.sim.timestep = 2;              % # of minutes for simulation time step
Settings.sim.timelag  = -1;             % # of hours from midnite, when measurements start
Settings.sim.startday = 1;              % 1-365, Julian Day # of first day in data
Settings.sim.tempunit = 1;              % 1 for degF, 0 for degC
Settings.sim.startdate = 1;             % 1 = Jan. 1
Settings.sim.enddate = 365;             % 365 = Dec. 31
Settings.sim.solver = 'ode3';           % solver used
Settings.sim.reltol = 1e-3;             % relative tolerance
Settings.sim.rebuild = false;           % Auto-run code builder not recc'd

% Solution rails, prevent numerical divergence, set them to approx equal
% 3 * largest gain or loss
Settings.rails.humidity = 1;            % kg/step, prevent solution runaway
Settings.rails.temp = 20;               % K/step


% File Names
% Weather Data File
Settings.file.Weather = ['C:\Program Files\MATLAB71\work\M.Eng Project' ...
    '\Data Sets\Agrimet Hourly Data Corvallis 2004_final_version.csv'];
% Parameters: run guessinit then guesssave to create a custom parameters
% matrix.
Settings.file.Parameters = ['C:\Program Files\MATLAB71\work\' ...
    'M.Eng Project\Data Sets\GUESS.mat'];
% Excel report for economic information
Settings.file.Report = ['C:\Program Files\MATLAB71\work\' ...
                       'M.Eng Project\GUESS.xls'];


% Climatic modeling
```

```
Settings.Climate.Wind.exponent = 0.2;        % how fast wind speed increases with respect↙
to
Settings.Climate.Wind.measuredheight = 2;      % meters
Settings.Climate.Radiation.LossesEnabled= true;        % enable longwave x-mission through↙
glazing(turn on for plastic, off for glass)
Settings.Climate.Radiation.Sun2PAR = conv2PAR('sunlight',1,'W/m2');    % umolar/Wm2↙
(typical full sun)
Settings.Climate.ConstantAirDensity = true;
Settings.Climate.modelnonEQET = false;         % model non-equilibrium transpiration
Settings.Climate.enablecondensation = true;              % enables accurate rel H↙
estimates. Turn off for speed.

% Locational Information
location.city = 'Corvallis';
location.lat = 44.63;             % degrees
location.long = 123.19;           % degrees
location.altitude = 78;           % meters or 230'
location.altunit = 'm' ;
location.timezone = 'Pacific';
switch location.timezone          % Select Standard Meridian
    case 'Atlantic'
        location.stdlong = 60;
    case 'Eastern'
        location.stdlong = 75;
    case 'Central'
        location.stdlong = 90;
    case 'Mountain'
        location.stdlong = 105;
    case 'Pacific'
        location.stdlong = 120;
    case 'Alaska'
        location.stdlong = 135;
    case 'Hawaii'
        location.stdlong = 150;
end
if location.altunit == 'm'
    location.pressure = patm (location.altitude);
elseif location.altunit == 'ft'
    location.pressure = patm (convft2m (location.altitude));
else
        error ('Invalid units');
end


 % Fundamental thermodynamic properties of simulation
 Properties.HeatCapacity= 1006;       % Heat capacity of air mass: J/kg-degK @ 25C
 Properties.AirDensity= airD(20,location.pressure);       % kg/m3
 Properties.LatentHeat = convkcal2kJ(540);              % Latent Heat of evaporating one↙
kg of water
 Properties.SpecificHeat.Water = convkcal2kJ(1);
 Properties.SpecificHeat.Steam = convkcal2kJ(0.48);
 Properties.SpecificHeat.IronPipe = .46;  %kJ/kg
```

```
 Properties.RH2O = 0.4619;            % Ideal Gas Constant for Water: kJ/kg-K
 Properties.R = 8.3145;               % Ideal Gas Constant: kJ/kmol-K
 Properties.SolarConstant = 1360;     % W/m^2 Total Extraterrestrial Direct Solar Rad.
 Properties.ratioDcDv     = 2/3;      % Passive diffusion resistance
 Properties.AirMolarMass  = 1000/29;  % mol/kg of air
 Properties.StefanBoltz   = 5.67E-8;  % Stefan-Boltzmann Constant W/m2-T4
 Properties.PsychroConst1 = 6.67E-4;  % Psychrometric Constant/K(raw)
 Properties.PsychroConst = Properties.PsychroConst1 * location.pressure;
 Properties.PriestlyTaylor = 1.26;    % Priestly Taylor constant(only works
                                      % well for well-watered vegation in
                                      % humid environments.
 Properties.kThermal    =  22.2E-6;   % m2/s, air 20C.Source: Monteith, Unsw
 Properties.LeH2O       =  0.89;      % H2O Lewis #, Monteith and Unsworth
 Properties.LeCO2       =  1.48;      % CO2 Lewis #
 Properties.viscosity   =  15.5E-6;   % kinematic, air 20C
 Properties.Pr          = 0.70;       % Prandtl # (kThermal/kin. visc)
 Properties.g           = 9.8;        % gravitational accel.


% Initial Greenhouse Internal Climatic Values
Settings.Climate.ZRef   = 2;           % Height of ref anemometer in m.
Settings.Climate.Initial.Temp = 22;  % deg. C
Settings.Climate.Initial.RelH = 0.6;
Settings.Climate.Initial.Humidity = humratio(location.pressure, ...
Settings.Climate.Initial.RelH * psat(Settings.Climate.Initial.Temp));
Settings.Climate.Initial.Humidity=RH2W(Settings.Climate.Initial.RelH,...
Settings.Climate.Initial.Temp,location.pressure);
Settings.Climate.CO2conc = 370;      %umol/mol air, Mona Loa data (July 2000)
Settings.Climate.O2conc  = 210000;    %umol/mol air
Settings.Climate.internalwindspeed = .03;         % m/s
method = Settings.sim.interpmethod;

% Boundary Layer resistances, for glazing or leaves
% Forced convection boundary layer resistances (roof and leaf)

BndryLayer.Lengthscale   = 0.02;          % needle length, m
BndryLayer.rHLeaf = RFlatPlateForced(Properties.viscosity, ...
            Settings.Climate.internalwindspeed, Properties.Pr, ...
            BndryLayer.Lengthscale, Properties.kThermal);    % resistance
                                                             % to heat, s/m
BndryLayer.rH2OLeaf = BndryLayer.rHLeaf * Properties.LeH2O ^(2/3);
BndryLayer.rCO2Leaf = BndryLayer.rHLeaf * Properties.LeCO2 ^(2/3);
BndryLayer.rRatioForced = (Properties.LeCO2/Properties.LeH2O)^(2/3);
BndryLayer.rRatioNatural = (Properties.LeCO2/Properties.LeH2O)^(3/4);
BndryLayer.Conversion    = .025;          % converts s/m to (m2s)/mol
                                          % Source: Jones, 1992
BndryLayer.gCO2 = 1/(BndryLayer.rCO2Leaf) * BndryLayer.Conversion * 1000;
                                       % Bndry Layer conductance, mmol/(m2s)
BndryLayer.Glazing = 1.64E-2;          % coeff, air @ 295K, Bakker et al.
% Create units structure
units.temp      = 'F';           % Celsius(C) or Farhenheit(F)
units.wind      = 'mph';         % miles/hr (mph) or m/s (mps)
```

```
units.solar     = 'ly';         % langleys-hours(ly) or watts/m^2 (Wm2) or
                                % BTU/ft2hr (Btuf2h)
units.VP        = 'kPa';        % water vapor pressure: kPa or psi
units.relH      = '%';          % as percentage(%) or fract of 1(1)
units.winddir   = 'deg';        % Wind direction as degrees from azimuth



% GREENHOUSE PARAMETERS
% Glazing

Greenhouse.Glazing.material = 'glass (single pane)';
% PAR radiation properties
Greenhouse.Glazing.transSW = 0.82;    % Visible transmissivity(beam normal)
Greenhouse.Glazing.transdiff = 0.78;  % Visible transmissivity (diffuse)
Greenhouse.Glazing.transMin = 0.5;    % visible transmissivity(beam sunset)
Greenhouse.Glazing.transInc = Greenhouse.Glazing.transSW - ...
                          Greenhouse.Glazing.transMin;
Greenhouse.Glazing.diffusivity = 0.1;   % 0.1-0.2(glass) 0.5+(polyethylene)

% Longwave Radiation Properties
Greenhouse.Glazing.transIR = 0.1;       % IR transmissivity
Greenhouse.Glazing.ViewFactor = 1.0;    % View of glazing from ground

% Conductivity
Greenhouse.Glazing.UValue  = convBtuhrft2FtoWm2K(1.1);% hr*F*ft^2/Btu, convert to metric
Greenhouse.Glazing.InsideRVal = 0.68;     % indoor boundary layer
Greenhouse.Glazing.OutsideRVal = 0.25;    % glass+outside boundary layer

% Distribution coefficients -- used to calculate glass temperature
Greenhouse.Glazing.Distribution.Inside= 1-Greenhouse.Glazing.InsideRVal/...
    (Greenhouse.Glazing.OutsideRVal +  Greenhouse.Glazing.InsideRVal);
Greenhouse.Glazing.Distribution.Outside= 1-Greenhouse.Glazing.OutsideRVal/...
    (Greenhouse.Glazing.OutsideRVal +  Greenhouse.Glazing.InsideRVal);

%Shape derived from Langhans et al..
% Default Gabled Greenhouse

Greenhouse.Shape.Style = 'gable roof';
Greenhouse.Shape.Length =  convft2m(100); % feet... convert to metric
Greenhouse.Shape.Width = convft2m(50);   % feet... convert to metric
Greenhouse.Shape.Height2Gable = convft2m(15); % feet... convert to metric
Greenhouse.Shape.GableHeight = convft2m(7.5);
Greenhouse.Azimuth = pi/2;       % angle of orientation E of N(short axis)
% Framing
Greenhouse.Framing.trans = 1.0;  % can vary with altitude angle, we assume
                                 % 1 here, and incorporate it's effects
                                 % into the glazing.

Greenhouse.Framing.material = 'aluminum';

    % Several types of roofs and shapes are available
```

```
% gable roof, gable (A-Frame), arch(hoop or quonset),
% arch (round) roof, gothic, multi-arch, and Venlo (multigable)

%Add Formulas for diff't shape compatability Shape
switch Greenhouse.Shape.Style

    case 'gable roof'
        % Roof Chord
        Greenhouse.Shape.RoofChord = sqrt(Greenhouse.Shape.Height2Gable^2 + ...
                                          Greenhouse.Shape.Width);

        % Glazing Area
        Greenhouse.Glazing.Area = 2* ((Greenhouse.Shape.Width + ...
        Greenhouse.Shape.Length) * Greenhouse.Shape.Height2Gable + ...
        0.5 * Greenhouse.Shape.GableHeight * Greenhouse.Shape.Width + ...
        Greenhouse.Shape.RoofChord * Greenhouse.Shape.Width);

        % Enclosed Air Volume
        % Volume: m3
        Greenhouse.Volume = Greenhouse.Shape.Length * Greenhouse.Shape.Width * ...
        (Greenhouse.Shape.Height2Gable + Greenhouse.Shape.GableHeight/2);
end

Greenhouse.xchngrate = 1.2;         % air exchange rate in vol. changes/hr.

% Floor
Greenhouse.Floor.Area = Greenhouse.Shape.Length * Greenhouse.Shape.Width;
Greenhouse.Floor.Perimeter = 2 * (Greenhouse.Shape.Length + ....
                            Greenhouse.Shape.Width);
Greenhouse.Floor.PerimeterLoss = convUvalCust2SI(0.4)*convft2m(1);  %W/m2-K;
Greenhouse.PerimeterWall.Height = convft2m(2);
Greenhouse.PerimeterWall.Length = Greenhouse.Shape.Length;
Greenhouse.PerimeterWall.Area   = Greenhouse.PerimeterWall.Height*Greenhouse.↙
PerimeterWall.Length;
Greenhouse.PerimeterWall.UValue  = convUvalCust2SI(0.7); %W/m-K
Greenhouse.PerimeterWall.material = 'concrete and 2in. foam';
Greenhouse.Utilization = 0.75;


% ENVIRONMENTAL CONTROL
% ENERGETICS AND FUEL
% Energy Consumption Info
% Source: DOE Energy Information Administration
% http://www.eia.doe.gov/cneaf/electricity/epm/table5_6_a.html
% http://www.eia.doe.gov/emeu/international/ngasprii.html
%
Energy.Prices.Electricity = 11.87;      % Retail cents/kWh (commercial, Mid-Atlantic)
Energy.Fuel.type          = 'natural gas';
Energy.Fuel.LHV           =  888.5;     % Lower Heating Value (kJ/mol)
Energy.Fuel.MolarMass     =  62.5;      % mol/kg
Energy.Fuel.Density       =  0.8;       % kg/m3 **
Energy.Fuel.HeatingValue  =  Energy.Fuel.LHV*Energy.Fuel.MolarMass; % 55530 kJ/kg **
```

```
Energy.Prices.Fuel        = 3.88;       % Dollars/MTBU natural gas (industrial users, US
Avg.)
Energy.Prices.Water       = 0.00;       % $/gal
Energy.Conversion.Fuel    = 9.48e5;     % kJ/MBTU
Energy.Conversion.Electricity = 3600;   % kJ/kWh
Fuel.Stochiometry.nH = 4;               % # of hydrogen atoms/molecule
Fuel.Stochiometry.nC = 1;               % # of carbon atoms/molecule
Energy.Fuel.MW = 12 * Fuel.Stochiometry.nC + Fuel.Stochiometry.nH;
Energy.Fuel.CO2Emission = Fuel.Stochiometry.nC * 44/Energy.Fuel.MW;
Energy.Fuel.Moisture = Fuel.Stochiometry.nH /2 * 18/Energy.Fuel.MW;
Energy.Fuel.Fuel2AirMass = 0.0579;       % Fuel 2 Air Volumetric
Energy.Fuel.Fuel2AirVol = 0.105;        % Fuel 2 Air Mass Basis
% ** source:www.engineeringtoolbox.com

% Ventilation & Cooling

% Fans:
Ventilation.maximumrate = 1.1;        %air changes per minute
%Proportions should add up to 1
Ventilation.Stage(1).setpoint = convFtoC(75);
Ventilation.Stage(1).bandwidth = 2;
Ventilation.Stage(1).proportion = 0.2;      % fraction of full power
Ventilation.Stage(2).setpoint = convFtoC(80);
Ventilation.Stage(2).bandwidth = 1;
Ventilation.Stage(2).proportion = 0.3;
Ventilation.Stage(3).setpoint = convFtoC(85);
Ventilation.Stage(3).bandwidth = 1;
Ventilation.Stage(3).proportion = 0.5;
Ventilation.Fans.MarginalPowerConsumption = 85;  % W/cms or 25 cfm/W

% Below parameters not used in model but useful for calculating Marg. Power
 %Ventilation.Fans.Flowrate = ;          % design. on flat part of curve
 %Ventilation.Fans.Headloss = ;          %approx 2 Pa static pressure, source:
                                         %Lou Albright
 %Ventilation.Fans.Efficiency = ;        choose fan with max. eff. at 2 Pa
 %Ventilation.Fans.Power = Ventilation.Fans.Flowrate* ...
 %Ventilation.Fans.Headloss/Ventilation.Fans.Efficiency;
 %Ventilation.Fans.Number = Greenhouse.Volume/Ventilation.Fans.Flowrate;




% Natural:
Ventilation.Natural.setpoint = convFtoC(75);
Ventilation.Natural.bandwidth = 1;
Ventilation.Natural.enabled = false;
Ventilation.Natural.Area = Greenhouse.Shape.Length * ...
  (Greenhouse.Shape.Height2Gable - Greenhouse.PerimeterWall.Height) * 0.5;
% Area of opening (roof or wall vents, in this case wall vents)
Ventilation.Natural.Height = 0.5 * (Greenhouse.Shape.Height2Gable + ...
    Greenhouse.PerimeterWall.Height);       % Height at eave
```

```
% Time Response of fans
Ventilation.Fans.timeconstant.ascending = 1;
Ventilation.Fans.timeconstant.descending = 3;
Ventilation.Fans.timeconstant.enable = 0;       % 1 or greater for on
Ventilation.Fans.slewrate.ascending = Greenhouse.Volume * Ventilation.maximumrate * 60 /↙
2;
Ventilation.Fans.slewrate.descending = -Ventilation.Fans.slewrate.ascending / 3;
Ventilation.Fans.slewrate.enable = 0;           % 1 or greater for on

Ventilation.Natural.timeconstant.ascending = 1;
Ventilation.Natural.timeconstant.descending = 3;
Ventilation.Natural.timeconstant.enable = 0;       % 1 or greater for on
Ventilation.Natural.slewrate.ascending = 1/2.5;    % full opening in 2.5 time steps
Ventilation.Natural.slewrate.descending = -1/2.5;  % full closing in 2.5 time steps
Ventilation.Natural.slewrate.enable = 1;           % 1 or greater for on
Ventilation.Natural.CDischarge = 0.6;              % default: sharp-crested orifice

% Energy Consumption Information
%

% Heaters
% Combustion
Heat.syseff = 0.95;            % Systemwide efficiency
Heat.combustioneff = 0.94;     % Cumbustion efficiency
Heat.type = 'hot water';
Heat.returnratio = 0.0;        % Amount of combustion gas returned to
                               % greenhouse
Heat.enableCondensation = true;  % enable condensing boiler
Heat.Texhaust = 50;              % deg C depends on unit
Heat.ExhaustEnergy = Properties.HeatCapacity * (Heat.Texhaust-22)* ...
    Energy.Fuel.Fuel2AirMass/1000;
Heat.Condensation = (Energy.Fuel.Moisture/18)*Energy.Fuel.Fuel2AirVol* ...
location.pressure/psat(Heat.Texhaust);
Heat.Heatingrate = 200;            %W/m2, varies w/climate,
                        %determine through design temp diff or experience
Heat.maximumflow = Greenhouse.Floor.Area*Heat.Heatingrate; % W; recc. by Dr. Albright
Heat.Stage(1).setpoint = convFtoC(65);
Heat.Stage(1).proportion = 0.3;     % fraction of maximum activated at this stage
Heat.Stage(1).bandwidth = 2;        % difference between setpoints
Heat.Stage(2).setpoint = convFtoC(60);
Heat.Stage(2).bandwidth = 2;
Heat.Stage(2).proportion = 0.3;
Heat.Stage(3).setpoint = convFtoC(55);
Heat.Stage(3).bandwidth = 2;
Heat.Stage(3).proportion = 0.4;
%Heat.ElectricityUse =  %W electricity/W of heat
% Maximum frequency controller at which can operate
% Full deflection every 20 min minute
Heat.timeconstant.ascending = 5;
Heat.timeconstant.descending = 5;
Heat.timeconstant.enable = 0;       % 1 or greater for on
```

```
Heat.slewrate.ascending = Greenhouse.Floor.Area * Heat.maximumflow / 3;
Heat.slewrate.descending = -Heat.slewrate.ascending / 3;
Heat.slewrate.enable = 0;           % 1 or greater for on


% Evaporative Cooling Pads
Pads.enabled = true;
Pads.setpoint = convFtoC(72);
Pads.efficiency = 0.85;     % fraction of dry bulb, wet bulb difference
Pads.blowdownfract = 1/5;   % fraction of water input wasted to prevent
                            % scaling


% Foggers (used to add humidity) or for evaporative cooling
% General Equation
% Humidity mass transfer conductance(kHa)
% Mass Flux = kHa*[lambda(Wambient-Watwetbulb)]
% Sensible effect = Lambda * Mass Flux.

Foggers.enabled = false;
Foggers.mistingenabled = true;    % operate foggers as evaporative cooling
                                % device in high temp, only useful for
                                % liquid misters

Foggers.efficiency = .97;
Foggers.type = 'mist';      % Type of system (steam or mist)
Foggers.steamtemp = 100;    % deg C, only use for steam foggers
foggingamt = Properties.HeatCapacity / ...   % change in air temp(K) per
(1000*Properties.LatentHeat) * Properties.AirDensity; % m3 of air per kg
                                             % fog added

% General Design Parameters
tempdiff = 6;               % Design temperature difference in C
% Max. amount of fog possible by fogger in kg/hr,
% 1.5 to account for nat. vent.
Foggers.maxRate =  tempdiff * foggingamt * Ventilation.maximumrate *  1.5 * ...
Greenhouse.Volume/60;
Foggers.Stage(3).setpoint = convFtoC(85);
% Tune fogger conductivity(spray diameter) to fully dissipate even at
% maximum setpoint even at 80% rel H. Conductivity measured in s-1.
Foggers.maxkHa = Foggers.efficiency * Foggers.maxRate / ....
   (equivWBhumidity(Foggers.Stage(3).setpoint, 0.8, location.pressure) - ...
   humratio(location.pressure, VPP(0.8,Foggers.Stage(3).setpoint)));

Foggers.maxRate = 3600 * Foggers.maxRate;   % Convert to L/hr, a more familiar unit

% Stage 1
Foggers.Stage(1).setpoint = convFtoC(75);
Foggers.Stage(1).kHa = Foggers.maxkHa * 0.2;
Foggers.Stage(1).Rate = Foggers.maxRate *0.2;

% Stage 2
```

```
Foggers.Stage(2).setpoint = convFtoC(80);
Foggers.Stage(2).Rate = Foggers.maxRate*0.3;
Foggers.Stage(2).kHa =  Foggers.maxkHa * 0.3;

% Stage 3
Foggers.Stage(3).setpoint = convFtoC(85);
Foggers.Stage(3).Rate = Foggers.maxRate*0.5;
Foggers.Stage(3).kHa =  Foggers.maxkHa * 0.5;




switch Foggers.type
    case 'steam'
        Foggers.SensibleEffect = Properties.SpecificHeat.Steam*1000* ...
        (Foggers.steamtemp-20);
        Foggers.FoggingRate = 250;           % g H2O/m^2/hr
        Foggers.Rate = Foggers.FoggingRate/1000 * Greenhouse.Floor.Area; % kg H2O/hr
    case 'mist'
        tempdiff= 5;            % Design temp diff (22C(68F) to 38C(100F))
        Foggers.SensibleEffect = Properties.LatentHeat * 1000 * Foggers.efficiency;
end



%CO2 Enrichment
% Controlled enrichment not implemented


%Lights
% General Parameters
Lights.type = 'HPS';
Lights.WperPAR = 1/conv2PAR(Lights.type,1,'W/m2');  % W per micromolar PAR
Lights.fDiffuse      = 0.7;         % Diffusivity of luminairie
Lights.Height        = Greenhouse.Shape.Height2Gable-convft2m(1.2);
Lights.enabled       = false;
% Control Parameters
Lights.Intensity = 75;            % Light Intensity in umolar/m2
Lights.setpoint = 88;
Lights.bandwidth = 40;
Lights.photoperiod = 12;           % # of hours
% Energy Parameters
Lights.BallastFactor = 1/10;       % W of ballast per W of lamp
Lights.MarginalPowerConsumption = Greenhouse.Floor.Area * ...
(1 + Lights.BallastFactor) * Lights.WperPAR;    % W/umolar * area

% Plant Parameters
% ---------------------- %
% Biomass
Plant.CarbonContent = 0.45;       % 45% carbon on dry-wt basis
Plant.WoodDensity   = 0.53;       % g/cc.
```

```
% Canopy Characteristics
Plant.Canopy.alpha = 0.9;         % shortwave(absorbtivity) at leaf
Plant.Canopy.absSW = sqrt(Plant.Canopy.alpha);  % SW abs at canopy
Plant.Canopy.Structure = 1;       % Leaf Angle Distribution Parameter
                                  % 0 for vertical
                                  % 1 for spherical (most common)
                                  % Inf for fully horizontal
                                  % typically range 0.5(onion) to 5(clover)
Plant.Canopy.kArtificial = kdirect(4*pi/9, Plant.Canopy.Structure);
                                    %Direct interception for overhead
                                    %supplemental lighting

% Photosynthesis

Plant.photoperiod.min = 5;        % umol/m2/sec or microeinstein
Plant.photoperiod.length = 12;    % hours
Plant.Photosynthesis.Tref = 25;   % Reference Temperature

% Amax, Jmax, and Vcmax values were taken from
% Photosynthesis-nitrogen relationships: interpretation of different
% patterns between Pseudotsuga menziesii and Populus × euroamericana
% in a mini-stand experiment, FRANCESCO RIPULLONE et al, 2003.
% Tree Physiology 23, 137-144

% Temperature dependence data for Jmax & Vcmax were taken from:
% The response of photosynthetic model parameters to temperature and
% nitrogen concentration in Pinus radiata D. Don.   A.S Walcroft et al.
% Plant, Cell & Environ. (1997) 20: 1338-1348.
% Values for kCO2, kO2 taken from radiata paper as well.
% Jmax, and VcMax values are expressed on a projected area basis.

% CO2 limited photosynthesis
Plant.Photosynthesis.Vcmax.value = 30.86;  %estimated from max leaf N: 2.5 g/m^2

Plant.Photosynthesis.Vcmax.qinc = 45000/ Properties.R;   % (Ha/R)
Plant.Photosynthesis.Vcmax.qdec1 = 203000/ Properties.R; % (Hd/R)
Plant.Photosynthesis.Vcmax.qdec2 = 650/ Properties.R;    % (Sv/R), may be changed to lower ↙
temperature optima

% Light Reaction limited photosynthesis
% Parameters modified to achieved Vopt @ about 76F
Plant.Photosynthesis.Jmax.value = 85.13;  %estimated from max leaf N: 2.5 g/m^2
Plant.Photosynthesis.Jmax.qinc  = 46000/ Properties.R;
Plant.Photosynthesis.Jmax.qdec1 = 199000/ Properties.R;
Plant.Photosynthesis.Jmax.qdec2 = 650/ Properties.R;

% Optimum temperature can be calculated using
% this formula:  Toptimum = Hd/[Sv - R*ln(Hv/[Hd-Hv])]


% Values for kO2, kCO2 confirmed by ESPM228, Advanced Topics in
% Biometeorology Notes, Dennis Baldocchi, Lecture 29 -- Photosynthesis.
```

```
% AND ESPM 129 Lec 10 Notes
Plant.Photosynthesis.kO2.value = 256000;          % O2 Michaelis constant, mmol/mol
Plant.Photosynthesis.kO2.q = 36000/Properties.R;  % O2 Michaelis constant, temp response
Leuning/Farquhar
Plant.Photosynthesis.kCO2.value = 302;            % CO2 Michaelis constant, umol/mol
Plant.Photosynthesis.kCO2.q = 59430/Properties.R; % CO2 Michaelis constant, temp response
Leuning/Farquhar
Plant.Photosynthesis.tau.value = 2900;            % CO2/O2 Specificity constant, see
Baldocchi

Plant.Photosynthesis.tau.q1 = .0451;              % 1/K, specificity temp response;
Walcroft
Plant.Photosynthesis.tau.q2 = .000347;             % 1/K^2, Walcroft
Plant.Photosynthesis.theta = 0.9;                 % smoothing parameter corrects for non-
Blackman response
Plant.Photosynthesis.qeff = .22;     % Maximum quantum efficiency(inf. CO2)
                                     % Theoretical maximum quantum
                                     % eff. is 0.25 mole electrons / mol
                                     % CO2 fixed & .125 mol CO2 per mole
                                     % light

% Respiration
% Leaf Values taken from Genetics of Dark Respiration and its Relationship
% with drought hardiness in Douglas Fir. Anekonda and Adams,
% Thermochimica Acta: 349,(2000),69-77.


%Plant.Respiration.Leaf.value =  .85;  %Dark Respirtion(LA basis),umol/m2,
%11.11 nmol/g ,mass mean 25C, 3 old yr. (apical meristem)
%Pinus radiata seedling value: 1.1: Walcroft et. al.
Plant.Respiration.Leaf.value = 0.85;  % At. 25C, 298K
Plant.Respiration.Leaf.Tref = 25;    %Dark Respiration, 298K
Plant.Respiration.Leaf.qinc = 9410;

% Q-10 based respiration rates for stems, roots, fruits, cones, etc...
Plant.Respiration.Tref = 25;


%   ROOTS
Plant.Respiration.Root.Q10 = 1.95;
Plant.Respiration.Root.value = 9.55;    % nmol CO2/g biomass(dry)
% Source: Kreuger and Farrell data in "High Soil CO2 inhibits Root ...
% Respiration of Doug. Fir", Qi, Marshall, Mattson, New Phyotologist 128
% Vol 3, Nov 1994.

% STEM(SAPWOOD(xylem, HEARTWOOD(not present in seedlings), PHLOEM)
% Pruyn, Michele L., Gartner, Barbara L. & Harmon, Mark E. (2002)
% "Within-stem variation of respiration in Pseudotsuga menziesii"
% New Phytologist 154 (2), 359-372.

Plant.Respiration.Wood.Q10   = 1.9;
Plant.Respiration.Wood.value = 0.63;    %***nmol CO2/g biomass(dry)
```

```
% we use value of outer sapwood for entire trees, reasoning: closest &
% simplest approx. to young, green wood.


Plant.Respiration.Growth = .25;   % fract. photosynthate devoted to growth
                                  % that is respired: Commonly used value



Plant.Canopy.LAI = 3.5;           % Leaf Area Index (one-sided[projected])
                             % LAI = canopy surface area / planting cell area)
Plant.Canopy.SLA = 76;            % g biomass/m^2, Ripullone et al.




% Allometric Relationships
% The diameter, height and area of the trunk of a tree can be related
% to the total dry stem mass using the following power law relationships:
%  B=Biomass. H=Height. D=Diameter. A = Basal Area
%  B = aD^x*H^y
%  A = pi*D^2/4
%  B = bA^(z)
%  H = cH^(w)
% Based on dimensional analysis:
% x=2, y =1, z=3/2.
% For most species, the exponents are usually slightly less +/-.05 than
% the above values
% For seedlings w can be assumed to be 1, SEE Target Seedling Concepts:
% Height and Diameter.  Mexal, J.G.; Landis T. D.
% Publication: Forest Nursery Proceedings, http://www.RNGR.net

Plant.Allometry.Height.Slope   =  1;  % Slope on log biomass-ht. plot(x)
Plant.Allometry.Diameter.Slope =  2;  % Slope on log biomass-Dia. plot(y)

% Source:
% Wood Density: 530 kg/m^3(Douglas Fir). Source:
            % http://www.simetric.co.uk/si_wood.htm


Plant.Biomass.density =        .530;  % g/ cc (dry wood density)

% Pipe model of stem growth: Astem = C * Acanopy
% Formulated by Shinozaki et al, 1964.
Plant.Allometry.Area.Pipe = 258;  % Pipe factor mm2 stem area/m2 leaf area
%SOURCE: Koskela, J. 2000. A process-based growth model for the grass stage pine ↙
seedlings. Silva
%Fennica 34(1): 3-20.


% ALLOMETRY EQUATION 2:
% Mstem = K * Astem^slope
```

```
% slope approx or exactly 3/2

% Data Parameterized using measurements from:
% Effects of Container Density and Plant Water Status on Growth and Cold
% Hardiness of Douglas-fir Seedlings.  Timmis, Roger; Tanaka, Y.
% Forest Science 22:167-172

Plant.Allometry.Taper = 67.8;       % cm of height per cm of dia. tends to
                                    %
Plant.Allometry.BranchFactor = 5.35; % Ratio: total stem mass to trunk mass
                                     %
Plant.Allometry.TaperFactor = 1;    % 3 = cone, 1=cylinder
% % Intercept of log basal area-log biomass plot g/cm2 of basal area^(3/2)
% OTHERWISE known as K in ALLOMETRY EQUATION 2
Plant.Allometry.Area.Intercept = 2/Plant.Allometry.TaperFactor * ...
        Plant.Allometry.Taper/sqrt(pi)* Plant.Allometry.BranchFactor * ...
        Plant.Biomass.density/1000;

 % Mstem = K[or Plant.Allometry.Area.Intercept]*Astem^(slope)
% Slope of log biomass log basal area plot, should be approx 3/2 +/- 2%
Plant.Allometry.Area.Slope = 1.5;     % Slope of log biomass vs log stem area


% Stomatal Conductance & Evapotranspiration
% Conductance expressed in terms of mmol/m2s(vapor)
 Plant.Stomata.Closedconductance = 17;      % source: Campbell p 91
                                            % source: Campbell p 91
 Plant.Stomata.sensitivitycoeff  = 480;    % Slope of Ball-Berry Model

                                            % source: Campbell

 Plant.Stomata.Openconductance   = 330;    % source: Campbell p 91
                                           % conductance, source: Thornley
 Plant.Stomata.Capacitance     = 10;        % mmol-m2 s-2, estimate



% Initial Plant Information
Plant.Initial.Biomass   = 0.57;         % g dry matter
Plant.Initial.RootFract = 0.2;         % fraction in roots

% Distribute Biomass
[Plant.Initial.StemBiomass Plant.Initial.RootBiomass ...
    Plant.Initial.LeafBiomass]= distributebiomass(Plant.Initial.Biomass,...
        Plant.Initial.RootFract, Plant.Allometry.Area.Intercept, ...
        Plant.Canopy.SLA, Plant.Allometry.Area.Pipe);


% Growth Target(only biomass implemented: height&dia. in future versions)
    Plant.Target.Value = 1.7;        % grams dry weight

% Planting and Container information
```

```
% Example container: 3cm * 12.5cm
% see Container Tree Nursery Manual Vol. 2, Landis: http://www.rngr.net
Planting.Utilization = .75;     % % usable floor area
Planting.Cell.Area = pi*1.5^2;  % cm2
Planting.Cell.Volume = 66;      % cm3
Planting.Cell.Density = 807;    % Cells/m^2

Planting.Cell.Utilization= Planting.Cell.Area * Planting.Cell.Density /(100^2);
Planting.Overlap = .2;          % how far(in cell diams.) canopies overlap
OV = Planting.Overlap + 1;
Planting.Cell.Canopy.Area = OV / Planting.Cell.Density;  %in m^2 canopy area
% with overlap
Planting.Number = Planting.Utilization * Greenhouse.Floor.Area * Planting.Cell.Density;
Planting.Area = Planting.Utilization * Greenhouse.Floor.Area; % total area for planting
Planting.ConversionFactor = 1/Planting.Number;

% Soil Parameters and Capacities in vol H2O/vol soil
Soil.FieldCapacity      = 3.5;          % No drainage under gravity
Soil.Saturation         = 4.5;          % All Pores Filled with H2O
Soil.WiltingPt          = 0.5;          % varies, between 0.5 & 1
Soil.Initial            = 3.0;
Soil.Porosity           = 0.75;         % volume voids/
                                        % total volume(voids + dry soil)
Soil.dryVolume          = (1-Soil.Porosity) * Planting.Cell.Volume;
                                    % Above is dry volume of soil per pot

% Irrigation Controls (volumetric moisture levels * field capacity)
Irrigation.setpointon   = 0.8;
Irrigation.setpointoff  = 1.4;
```

```
% guessread.m
% Jamison Hill
% 3-4-2005 -- 5-24-2006
% Opens, reads, processes, and interpolates (per minute basis) hourly
% weather data from a text file and converts it into weather vectors for
% use by the Simulink model.  Also calculates vapor pressures, and
% Weather data must be in tab, space, or comma delimited text format
% And MUST include in the following columns in this order
% date column
% time column
% Running time in days from first data point
% Outdoor temperature F or C
% Outdoor Relative Humidity % of VP Saturation
% Dewpoint temperature
% Wind Direction in degrees/radians E of N(azimuth)
% Wind Speed in mph or m/s
% Solar radiation in langleys/hr or W/m2
%
% startcol = column with running time in hrs, min or seconds
% startrow = column at end of header
% NOTE:---------
% This M-File cannot be executed standalone, and must be called
% from within the main GUESS module.  To run GUESS, type GUESS in the
% command window prompt, and press <enter>.
% ----------------------------------------------------------------
% last modified 4-30-06

%-----------
% Read data
%-----------
%try                     % Look for errors
tic                      % Start timer
warning off
fprintf('\nWeather Data File Processing\n')
fprintf('Reading File......');
W = dlmread(Settings.file.filepath, Settings.file.delimiter, ...
    Settings.file.startrow, Settings.file.startcol);
switch Settings.sim.frequency
    case 'hourly'
        Hours = W(:,1);               % Convert to hourly data set
    case '15min'
        Quarters = W(:,1);
    %case '5min'
    %case '2min'
    case '1min'
        Minutes = W(:,1);
    otherwise
        error('\n Invalid time step \n');
    return
end
DateRaw     = W(:,2);            % Days elapsed since start of growing season
TempRaw     = W(:,3);            % in deg C or F
```

```
RelHRaw     = W(:,4);            % rel. Humidity %
%DewPRaw     = W(:,5);           % Dewpoint Temp C or F; eliminate dewpoint
WindDirRaw1 = W(:,6);            % Wind Direction in degrees from South, azimuth
WindRaw     = W(:,7);            % Wind Speed
SolarRaw    = W(:,8);            % Solar insolation in langleys or watts/m^2
fprintf('DONE\n')

% ----------------
% Unit Conversion
% ----------------
% ---------------
% Convert date into metric for ease of calculations
% ---------------


if units.temp == 'F'
    Temp_C  = convFtoC(TempRaw);
%    DewP_C  = convFtoC(DewPRaw);
elseif units.temp == 'C'
    Temp_C = TempRaw;
%    DewP_C = convFtoC(DewPRaw);
else
    error ('Invalid temperature units');
end

switch units.wind
    case 'mph'
        WindRaw = convmph2mps (WindRaw);
    case 'mps'
        % do nothing
    otherwise
        error ('Invalid wind speed units');
end
    %WindRaw = PowerLawWindConversion
switch units.solar
    case 'Wm2'
        % do nothing
    case 'ly'
        SolarOld = SolarRaw;
        SolarRaw = convLyhr2Wm2 (SolarRaw);
    case 'Btuf2h'
        SolarRaw = convBtuhrft2toWm2(SolarRaw);
    otherwise
        error('Invalid solar units');
end

switch units.relH
    case '%'
        RelHRaw = RelHRaw / 100;
    case '1'% do nothing
    otherwise
        error('Invalid humidity unit');
```

```
end

switch units.winddir
    case 'deg'
        WindDirRaw = WindDirRaw1 / 180*pi();
    case 'rad' %do nothing
    otherwise
        error('Invalid wind direction unit');
end

%-------------------------
% Check sampling rate
%-------------------------
switch Settings.sim.frequency
    case 'hourly'             % 1hr. sampling rate
         TimeRaw = Hours .*(60/Settings.sim.timestep);
    case '15min'              % 15 min. sampling rate
         TimeRaw = Quarters .*(15/Settings.sim.timestep);
    case '1min'               % 1 min. sampling rate
         TimeRaw = Min;
end
EOTime = TimeRaw(length(TimeRaw));



% PAR Conversion
LightRaw     = conv2PAR('sunlight',SolarRaw,'W/m2');
fprintf('Vapor Pressure, Wetbulb Calculations ......');
% Calculate humidity measurements: vapor pressure or humidity ratios
% Calculate Saturation Vapor Pressures
SatVPRaw    = psat(Temp_C);                  % Saturation Vapor Pressures
% modify SatVP to use Teten's formula (will run faster)
% modify all to run with arrays; add *. and ./
VPRaw       = RelHRaw .* SatVPRaw;      % Vapor Pressures
HumRaw  = humratio(location.pressure, VPRaw); % humidity ratio (mass H2O/mass air)
WetBulbRaw   = (wetbulb (Temp_C, RelHRaw, location.pressure))';
WetBulbVP    = psat(WetBulbRaw);
WetBulbHumRaw = humratio(location.pressure, WetBulbVP);
fprintf('DONE\n');

fprintf('Wind Pressure Calculations ......');
% Calculate Wind Incidence Angle and Natural Ventilation Pressure Coeff.
IncidenceAngleRaw1 = WindDirRaw - Greenhouse.Azimuth;   % Inlet 1
IncidenceAngleRaw2 = WindDirRaw + Greenhouse.Azimuth;   % Inlet 2

% Coefficient of Pressures

CpRaw2 = ones(length(IncidenceAngleRaw1),1);
CpRaw1 = ones(length(IncidenceAngleRaw1),1);
for X = 1:length(IncidenceAngleRaw1)
    CpRaw1(X) = calcWindPressCoeff(IncidenceAngleRaw1(X));
    CpRaw2(X) = calcWindPressCoeff(IncidenceAngleRaw2(X));
```

```
end
% Ventilation Rate
WindFactorRaw = abs(CpRaw1 - CpRaw2)./(sqrt(abs(CpRaw1 - CpRaw2)));
UNatVRaw = Ventilation.Natural.CDischarge .* WindFactorRaw .* WindRaw;       % Ventilation
Rate

% ---------------------------------
% Calculate Wind Pressures in Pascals
% Find Windspeed at eave height
WindRaw = WSConvert(WindRaw, Settings.Climate.Wind.measuredheight, ...
            Ventilation.Natural.Height,Settings.Climate.Wind.exponent);
% Wind Pressure Inlet 1
WindPressure1 = 0.5*AirD(Temp_C, location.pressure)*WindRaw.^2* CpRaw1;
% Wind Pressure Inlet 2
WindPressure2 =  0.5*AirD(Temp_C, location.pressure)*WindRaw.^2* CpRaw2;
WindPressureRaw = WindPressure1 - WindPressure2;  %Flow driven by pressure diff.
fprintf('DONE\n');
%
fprintf('Solar Radiation Calculations ......');

% --- Solar Altitude & Clearness Index ---
% Solar Time
Hrs = TimeRaw/(60/Settings.sim.timestep) + Settings.sim.timelag;
Clocktime = mod(Hrs, 24);
DayRaw = 1+ Hrs ./ 24;
DayRaw2 = floor(Clocktime ./ 24);
HrAngle = HourAngleCorrect(Clocktime, DayRaw,...
    location.long - location.stdlong);
Declination = declination (DayRaw);

% Solar Altitude
Altitude = solaraltitude (Declination, location.lat, HrAngle);
Altitude(Altitude < 0) = 0;    % can use -6 deg for civil twilight
Altitude = (pi/180)* Altitude; % convert to radians

% Clearness Index
ETSolar = sin(Altitude) * Properties.SolarConstant;  %Calculate ET Radiation
KIndex(1:length(SolarRaw),1)= 0.8;
%KIndex = SolarRaw./ETSolar;    % clearness index
for i = 1:length(TimeRaw)       % Correct for div by/zero
    if SolarRaw(i) > ETSolar(i) % Can't have clearness index > 1
        KIndex(i,1) = 1;
    elseif ETSolar(i) == 0 && i > 2      % Maintain previous value
        KIndex(i,1) = KIndex(i-1,1);     % throughout the night
    else
        KIndex(i,1) = SolarRaw(i)./ETSolar(i);    %clearness index
    end
end


% Diffuse vs Direct
[DiffuseRaw DirectRaw] = splitbeam(SolarRaw, KIndex);
```

```
fractDiffRaw = fDiffuse(KIndex);
% fDirect  = 1 - fDiffuse;
fprintf('DONE\n');

%---------------------
% Longwave Sky Balance
%---------------------
fprintf('Longwave Calculations ......');
e_sky = eSkyB(TempRaw, VPRaw, KIndex);
TSkyRaw =  e_sky.^(1/4) .* TempRaw;
fprintf('DONE\n');
fprintf('Interpolating ......');
%------------------
% Interpolate to 1 minute time step for simulation
%------------------
Settings.sim.Maxtime = EOTime;
%TimeFill = ((1:60:EOTime).');
Time     = ((1:EOTime).');
%Time2   = (1:EOTime/2).';
Date     = interp1(TimeRaw, DateRaw, Time,'linear');
Temp     = interp1(TimeRaw, Temp_C, Time, method);
RelH     = interp1(TimeRaw, RelHRaw, Time, method);
% DewP     = interp1(TimeRaw, DewP_C, Time, method);
Wind     = interp1(TimeRaw, WindRaw, Time, method);
Solar    = interp1(TimeRaw, SolarRaw, Time, method);
UNatV    = interp1(TimeRaw, UNatVRaw, Time, method);
WindFact = interp1(TimeRaw, WindFactorRaw, Time, method);
Diffuse  = interp1(TimeRaw, DiffuseRaw, Time, method);

fractDiffuse = Diffuse ./Solar;
fractDiffuse(~isfinite(fractDiffuse))= 0;
%Direct   = interp1(TimeRaw, DirectRaw, Time, method);
SatVP    = interp1(TimeRaw, SatVPRaw, Time, method);
Humidity = interp1(TimeRaw, HumRaw, Time, method);
Wetbulbs    = interp1(TimeRaw, WetBulbRaw, Time, method);
WBHumidity  = interp1(TimeRaw, WetBulbHumRaw, Time, method);
VP          = interp1(TimeRaw, VPRaw, Time, method);
Light       = interp1(TimeRaw, LightRaw, Time, method);   % Light in PAR
Angle       = interp1(TimeRaw, Altitude, Time, method);
TSky        = interp1(TimeRaw, TSkyRaw, Time, method);
%--- Wind and Natural Ventilation  ---
WindDir  = interp1(TimeRaw, WindDirRaw, Time, method);
%IncidenceAngle = interp1(TimeRaw, IncidenceAngleRaw, Time, method);
%Cp = interp1(TimeRaw, CpRaw, Time, method);
WindPressure = interp1(TimeRaw, WindPressureRaw, Time, method);
fprintf('DONE\n');

fprintf('Generating Lookup Tables ......\n');
fprintf('Saturation Humidity ......');
%--- Saturation Vapor Look up Table ---
PSatLookUpTable.T = 0:0.5:55;
```

```
PSatLookUpTable.P = PSat(PSatLookUpTable.T);
%--- dPSat Look Up Table ---
dPSatLookUpTable.T = 0:0.5:55;
dPSatLookUpTable.P = dPSat(dPSatLookUpTable.T);
%--- Humidity at WetBulb Table ---
equivWBLookUpTable.T = linspace(0,50,100);
equivWBLookUpTable.H = linspace(0,1,100);
[T2 H] = meshgrid(equivWBLookUpTable.T, equivWBLookUpTable.H);
equivWBLookUpTable.Values = equivWBhumidity(T2, H/100, location.pressure);
%--- Plant Stuff ---
InitPlant;
fprintf('DONE\n');

fprintf('Packing and Cleanup ......');

%---------------
% Create Weather structure in format wanted by Simulink
% and organize into structure for ease of packaging
% Simulink Format
% Signal = [timestep data];  use column vectors for both.
% ----------------

Weather.Temp         = [Time Temp];
Weather.RelH         = [Time RelH];
Weather.Solar        = [Time Solar];
Weather.Wind         = [Time Wind];
Weather.WindFact     = [Time WindFact];
Weather.UNatV        = [Time UNatV];
Weather.SatVP        = [Time SatVP];
Weather.Humidity     = [Time Humidity];
Weather.WetBulb      = [Time Wetbulbs];
Weather.VP           = [Time VP];
%Weather.Light        = [Time Light];
Weather.WindDir      = [Time WindDir];
Weather.WBHumidity   = [Time WBHumidity];
Weather.WindP        = [Time WindPressure];
Weather.Angle        = [Time Angle];
Weather.fractDiffuse = [Time fractDiffuse];
Weather.TSky         = [Time TSky];
%Weather.Direct      = [Time Direct];
%----------------
% Create timer object to override Simulink built-in clock for output
% graphing and scoping
% ------------------


%--------------
% Cleanup
% Clear uneeded data
%--------------

clear W DateRaw RelHRaw  DewPRaw  T_C CpRaw VP Light SolarOld TempRaw
```

```
clear Hours Quarters Minutes  Light RelH SatVP Humidity Wetbulbs Wind
clear Wind Humidity DewP DewP_C IncidenceAngle  HumRaw Cp WetbulbVP
clear WindDir WindDirRaw1  VPRaw SatVPRaw VP SolarOld WindPressure SolarRaw
clear IncidenceAngleRaw WetBulbRaw Temp  WetBulbHumRaw WBHumidty Angle
clear DiffuseRaw DirectRaw  Direct CpRaw1 CpRaw2 DayRaw Altitude
clear DayRaw2 WBHumidity TimeRaw Temp_C Time TimeFill  KIndex
pack;
fprintf('DONE\n\n');
disp('Ready for simulation!');
toc
warning on
%catch
%    disp('Corrupt/Invalid Weather data file OR');
%    disp('guessread is not a standalone m-file, run guessinit first');
%end
```

```
% guessmodel.m
% Guess model Simulink--MATLAB interface
disp('');
fprintf('Loading Model.........');
warning off
options = simset('RelTol', Settings.sim.reltol, ...
                 'Solver',Settings.sim.solver);
T_Start = (Settings.sim.startdate - Settings.sim.startday) * ...
          (24 * 60 / Settings.sim.timestep);
T_Final = (Settings.sim.enddate - Settings.sim.startday) * ...
          (24 * 60 / Settings.sim.timestep);
guesssim;
fprintf('DONE\n');
disp('');
% Run Accelerator if Necessary
if Settings.sim.rebuild
    disp('Building Model');
    accelbuild('guessim');
    disp('DONE');
end
% Start timer
tic
fprintf('Running Simulation.........');
sim('guesssim', [T_Start T_Final]);
fprintf('DONE\n');
disp('Simulation Completed!');
toc
```

```
% guessoutput.m
% Created May 9-2006
% Output File for GUESS model
% Draws graphs, generates spreadsheets of results from the guess model

Outputfile = Settings.file.Report;
% Extract Temperature
GuessOutput.date = Guess_Date.signals.values;
GuessOutput.temp = GuessOutput_Temp.signals.values;

switch Settings.sim.tempunit
    case 1
        indoor = 'Indoor Temp.(F)';
        outdoor = 'Outdoor Temp.(F)';
        axis = 'Temp. (F)';
    case 2
        indoor = 'Indoor Temp.(C)';
        outdoor = 'Outdoor Temp.(C)';
        axis = 'Temp. (C)';
end
% Extract Humidity
GuessOutput.relH = GuessOutput_Humidity.signals.values;

% Extract Energy Costs
GuessOutput.Costs.total = GuessOutput_Costs.signals.values(:,1);
GuessOutput.Costs.gas = GuessOutput_Costs.signals.values(:,2);
GuessOutput.Costs.electricity = GuessOutput_Costs.signals.values(:,3);
GuessOutput.Costs.water = GuessOutput_Costs.signals.values(:,4);

% Extract Energy Quantites
GuessOutput.Quants.gas =GuessOutput_Quantities.signals.values(:,4)/...
                        Energy.Fuel.Density;
GuessOutput.Quants.electricity = GuessOutput_Quantities.signals.values(:,2);
GuessOutput.Quants.water = GuessOutput_Quantities.signals.values(:,3);
FuelConverter = (1/(convft2m(1))^3)/Energy.Fuel.Density;  % from kg to ft^3

%Extract Plant Data
GuessOutput.Plant.Diam      = Guess_PlantGrowth.signals.values(:,3);
GuessOutput.Plant.Height    = Guess_PlantGrowth.signals.values(:,4);
GuessOutput.Plant.Biomass   = Guess_PlantGrowth.signals.values(:,1);
GuessOutput.Plant.Crops     = Guess_PlantGrowth.signals.values(:,2);

% Extract Lighting Data
GuessOutput.Lights.integral = GuessOutput_light.signals.values(:,2);
GuessOutput.Lights.photoperiod = GuessOutput_light.signals.values(:,1);

% Extract Operating Conditions
GuessOutput.Environment.CO2 = Guess_Environment.signals.values(:,1);
GuessOutput.Environment.PAR = Guess_Environment.signals.values(:,2);
% Create Guess XLS Workbook
%Define worksheet 1
date = GuessOutput.date (1:((60*24)/Settings.sim.timestep):end);
```

```
biomass = GuessOutput.Plant.Biomass (1:(60*24)/Settings.sim.timestep:end);
diam = GuessOutput.Plant.Diam (1:(60*24)/Settings.sim.timestep:end);
height = GuessOutput.Plant.Height (1:(60*24)/Settings.sim.timestep:end);
integral = GuessOutput.Lights.integral(1:(60*24)/...
            Settings.sim.timestep:end);
photoperiod = GuessOutput.Lights.photoperiod(1:(60*24)/...
            Settings.sim.timestep:end);
WS1 = {'Time(day)', 'Biomass(g)', 'Height(cm)', 'Diam.(mm)', ...
       'Photoperiod(hr)', 'Light Integral(moles)'; ...
       date, biomass, diam, height, photoperiod, integral};
%Write worksheet 1
% There is a bug in xlswrite when writing cell arrays to a data file
success = xlswrite(Outputfile, WS1(1,:), 'Growth', 'A1');
success = xlswrite(Outputfile, cell2mat(WS1(2,:)), 'Growth', 'A2');
if ~success
    error('Worksheet failed');
end
%Define worksheet 2
total = GuessOutput.Costs.total (1:(60*24)/ Settings.sim.timestep:end);
gas = GuessOutput.Costs.gas (1:(60*24)/ Settings.sim.timestep:end);
electric = GuessOutput.Costs.electricity (1:(60*24)/ ...
            Settings.sim.timestep:end);
water = GuessOutput.Costs.water (1:(60*24)/ Settings.sim.timestep:end);
WS2 = {'Time(day)', 'Total Cost($)', 'Fuel ($)', 'Electric($)', ...
       'Water($)'; date, total, gas, electric, water};

%Write worksheet 2
success = xlswrite(Outputfile, WS2(1,:), 'Costs','A1');
success = xlswrite(Outputfile, cell2mat(WS2(2,:)), 'Costs','A2');
if ~success
    error('Worksheet failed');
end


% Draw Graph 1: Weather
% Temp. subgraph
figure(1)
subplot(2,1,1)
plot(GuessOutput.date, GuessOutput.temp)
title('Temperatures');
xlabel('Day');
ylabel(axis);


% Humidity subgraph
subplot(2,1,2)
plot(GuessOutput.date, GuessOutput.relH)
title('Relative Humidity % of 100');
xlabel('Day');
ylabel(axis);
legend('outdoor','indoor', 'Location', 'SouthEast', 'Orientation',...
        'horizontal');
```

```
h = toptitle('Indoor vs. Outdoor Climatic Conditions');
set(h, 'FontSize', 14)



% Costs Diagram
figure(2)
plot(GuessOutput.date, [GuessOutput.Costs.total, GuessOutput.Costs.gas,...
    GuessOutput.Costs.electricity, GuessOutput.Costs.water])
legend('Total', 'Natural Gas', 'Electricity', 'Water', 'Orientation', ...
       'Horizontal', 'Location','Best')
h = title('Energy Costs');
xlabel('Day')
ylabel('Cost ($)')
set(h, 'FontSize', 14)

% Quantities Diagram
figure(4)
plot(GuessOutput.date, [GuessOutput.Quants.gas*FuelConverter,...
    GuessOutput.Quants.electricity, GuessOutput.Quants.water])
legend('Natural Gas(ft^3)', 'Electricity(kWh)', 'Water(gal)', 'Orientation', ...
       'Horizontal', 'Location','Best')
   xlabel('Day')
   ylabel('Energy Quantity')
   h = title('Energy Quantities');
   set(h, 'FontSize', 14)

% Plant Growth Diagram
figure(5)
% Height Graph, subplot 1
subplot(2,2,1)
plot(GuessOutput.date, GuessOutput.Plant.Height);
xlabel('Day');
ylabel('Height (cm)');

subplot(2,2,2)
plot(GuessOutput.date, GuessOutput.Plant.Diam);
xlabel('Day');
ylabel('Stem Diameter (mm)');

subplot(2,2,3)
plot(GuessOutput.date, GuessOutput.Plant.Biomass);
xlabel('Day');
ylabel('Total Dry Biomass (g)');

subplot(2,2,4)

plot(GuessOutput.date, GuessOutput.Plant.Crops);
xlabel('Day');
ylabel('Crops Harvested(#)');
```

```
h = toptitle('Plant Growth Characteristics');
set(h, 'FontSize', 14);

% Indoor Temperature Distribution
figure
plot(GuessOutput.date, GuessOutput.temp)
title('Temperatures');
xlabel('Day');
ylabel(axis);
legend('outdoor','indoor');
figure
hold on
hist(GuessOutput.temp(:,2));
title('Indoor Temperature Distribution');
xlabel(axis);
ylabel('freq.');
meantemp = mean(GuessOutput.temp(:,2));
disp(sprintf('Mean Indoor Temperature: %3.2f', meantemp));
stdev = std(GuessOutput.temp(:,2));
disp(sprintf('Standard Deviation Indoor Temperature: %3.2f', stdev));
```

```
function h = toptitle(string)
% TOPTITLE
%
% Places a title over a set of subplots.
% Best results are obtained when all subplots are
% created and then toptitle is executed.
%
% Usage:
%            h = toptitle('title string')
%

% Patrick Marchand (prmarchand@nvidia.com)
% Thomas Holland (tholland@infinityassociates.com)

titlepos = [.5 1]; % normalized units.

ax = gca;
set(ax,'units','normalized');
axpos = get(ax,'position');

offset = (titlepos - axpos(1:2))./axpos(3:4);

h2 = text(offset(1),offset(2),string,'units','normalized',...
     'horizontalalignment','center','verticalalignment','middle');

% Make the figure big enough so that when printed the
% toptitle is not cut off nor overlaps a subplot title.
h = findobj(gcf,'type','axes');
set(h,'units','points');
set(gcf,'units','points')
figpos = get(gcf,'position');
set(gcf,'position',figpos + [0 0 0 15])
set(gcf,'units','pixels');
set(h,'units','normalized');

% Return title object
h = h2;
```

PSYCHROMETRIC AND WEATHER TOOLBOX

```
% AIRD.M
% Psychrometric Toolbox
% Air Density Calculator
% Determines the air density for a given atmospheric pressure, vapor
% partial pressure, and dry-bulb pressure
% T    -- Dry-bulb temperature in C
% P    -- Pressure of dry air in kPa
% PV   -- Partial pressure of H20 vapor
% rho  -- Density of air kg/m^3
% rho = AIRD (T, P, (PV optional))

function [rho]= AirD (T, P, PV)
T1 = T + 273.15;
if nargin == 3
    rho = P*1000 /(287.05 * T1) + PV*1000 /(461.495 * T1);
end
if nargin == 2
    rho = P*1000 /(287.05 * T1);
end
return
```

```
% calcWindPressCoeff.m
% Calculates the Pressure Coefficent (Cp) for wind-induced natural
% ventilation. The pressure coefficient corrects for the effect of
% non-perpendicular wind flow.  And it used along with the orifice equation
% to detemine the mass flow rate of air leaving a naturally ventilated
% building.
% Uses formula from Burns and Deru 2003.
% The background paper(NREL/CP-550-33698) can be found
% on the NREL website using this code:
% Add google search3434NREL/CP-550-33698
% Usage:
% Cp = calcWindPressCoeff(phi)  where phi is angle of incidence between
% building and the wind.
function Cp = calcWindPressCoeff(phi1)
phi = 0;
for x = 1:length(phi1)
    phi = phi1(x);
    if phi1(x)>2*pi
        phi = mod(phi1(x),2*pi);
    end
    if phi1(x)>pi
        phi = pi-(phi1(x) - pi);
    end

    U =(1.248-.703.*(sin(phi/2))-1.175 .*(sin(phi)).^2+ ...
    0.131*(sin(2*phi)).^3+ 0.769 .* (cos(phi/2)) + 0.07*(sin(phi/2)).^2+ ....
    0.717 .* (cos(phi/2)).^2);
    Cp(x) = 0.6*log(U);
end
return
```

```
% DEWPOINT.m
% Calculates the dew-point temperature (C).                             }
% Based on formulae in the ASHRAE Handbook of Fundamentals.             }
% Applies for temperatures between -60 and +70 C.                       }
% Inputs are dry-bulb temperature (C) and water vapor pressure (kPa).  }
% DP        -- Dewpoint (C)
% T         -- Dry-bulb temperature C
% RH        -- relative humidity
% DP = DEWPOINT (T,RH)

function [DP]= dewpoint (T, RH)
    t = T;
    P = log(1000.0 * VPP(RH, t));
    if (t < -60) || (t > 70)
        warning ('PYSCH07: Temperature out of range -60 : 70 C');
    end
    if (RH < 0) || (RH > 1)
        error ('PSYCH02: FATAL Humidity out of range 0 : 1.0');
    end
    if (t <= 0)
        DP= -60.45 + 7.0322*P + 0.37*P^2;
    else DP = -35.957 - 1.8726*P + 1.1689*P^2;
    end
return
```

```
% DPSAT.m
% Psychrometric Toolbox
% Computes the first derivative (slope) of the sat. vapor-pressure curve
% Used in the Penman-Monteith Equation
% Formula valid from 0 to 120 deg C
% T     -- Dry-bulb temperature in C
% dP    -- Slope of saturation vapor pressure curve in kPa/L
% dP  =  dVPSat (T)  returns slope in kPa/K
%
function [DVapSat] = dPSat(T)
if ~isempty(T((T < 0))) || ~isempty(T((T > 120)))
    warning ('PSYCH03: Temperature out of range 0 : 120C')
end
T1 = T + 273.15;
    DVapSat = 1/1000 * (5800.2206./T1.^2 - 0.048640239 + .83529536e-4.*T1 -...
    .43356279e-7.*T1.^2 + 6.5459673./T1).* exp(-5800.2206./T1 + 1.3914993 -...
    0.04860239.*T1 + .41764768e-4.*T1.^2 - .14452093e-7.*T1.^3 + ...
    6.5459673.*log(T1));
return
```

```
% ENTHALPY.m
% Psychrometric Toolbox
% Calculates the enthalpy of air/water vapor mixture (kJ/kg).
% Based on formula in the ASHRAE Handbook of Fundamentals.
% Inputs are dry-bulb temperature (C) and humidity ratio (unitless).
% H     -- enthalpy in kJ/kg
% T     -- Dry-bulb temp. C
% W     -- Humidity Ratio
% H = ENTHALPY(T, W)
function [H] = enthalpy(T, W)
H = 1.006*T + W.*(2501.0 + 1.805*T);
return
```

```
% equivWBhumidity.m
% Calculates the humidity ratio (kg H2O/ kg air)
% at wetbulb for a given temperature and rel. H at drybulb.
% WBB = equivWBhumidity (temp, relH, Pair)
% temp expressed as fraction. relH expressed as fraction.
% Pair expressed in kPa.
function WBH = equivWBhumidity (temp, relH, Pair)
WBH = humratio(Pair, psatmat(wetbulb (temp, relH, Pair)));
```

```
% eSkyB.m
% Calculates sky emissivity for use in longwave radiation calculations
% A clearness index correction was added to take in account cloud effects.
% SkyTemp = TSky (T_ambient, vp, Ki)
% Temperatures are expressed in degrees Celsius
% vp Vapor Pressure in kPa
% ki clearness index
% Clear sky emissivities calculated using Brutsaert's method (1982)
% As detailed in the paper, "Cloud Effects in Estimation of Instaneous
% Downward Longwave Radiation", Water Resources Research, vol 29, 599-605.
function e_sky = eSkyB(T_ambient, vp, Ki)
if nargin == 2
    e_sky = 1.24 .* (10 .* vp./(T_ambient+273)) .^ (1/7);
elseif nargin == 3
    %if Ki =>
    e_sky = min(1.27 .* (10 .* vp./(T_ambient+273)) .^ (1/7)...
    .* Ki.^-.0227, 0.99);
end
```

```
% HUMRATIO.m
% Calculates humidity ratio W (kg vapor/kg air)
% PV -- Partial pressure of water vapor
% P --  Atmospheric air pressure in kPa (P)
%
% W = HUMRATIO (P, PV)
function [W] = humratio (P, PV)
% if (PV > P) | ((PV < 0) | (P < 0))
%    P
%    PV
%    error ('PSYCH03: FATAL Illegal or negative pressure values')
% end
W = 0.62198 * PV ./(P - PV);
return
```

```
% PATM.m
% Psychrometric Toolbox
% Calculates atmsopheric pressure in kPa
% Equation based on formulae in the ASHRAE Handbook of Fundamentals
% h --  elevation in meters
% P --  atmospheric pressure in kPa
% P = PATM (h)
function [P] = PAtm (h)
P = 101.325 * exp(-0.00011943*h) - 6.799e-06*h - 6.976e-08*h.^2;
return
```

```
%  PSAT.M
%  Psychrometrics Tool Box
%  Calculates Saturation Vapor Pressure
%  Based on formulae provided in the ASHRAE Handbook of Fundamentals
%  Valid for range -40 C to 120 C
%  T --    Dry bulb temperature in C
%  P --    Saturation Vapor Pressure of Water in kPa
%  P = PSAT (T)
function [VPSat2] = psat(T)
siz = size(T);
rowmax = siz(1);
colmax = siz(2);
NMax  = rowmax * colmax;
VPSat1 = ones(rowmax, colmax);
for k = 1:NMax;
    Tn=T(k);
    if (Tn < -40) || (Tn > 120)
        Tn
        warning ('PHYSC01: Temperature out of range -40 : 120 C');
    end
    T1 = Tn + 273.15;
    if T1 >= 273.15
        VPSat1(k) = exp(-5800.2206/T1 + 1.3914993 - 0.048640239*T1 + ...
            0.41764768e-4*T1.^2 - 0.14452093e-7*T1.^3 + 6.5459673 * ...
            log(T1))/1000;
    else
        VPSat1(k) = exp(-5674.5359/T1 + 6.392547 - 0.9677843e-2*T1 + ...
            0.6221570e-6*T1^.2 + 0.20747825e-8*T1.^3 + ...
            0.9484024E-12*T1.^4 + 4.1635019* log(T1))/1000;
    end
end
VPSat2 = VPSat1;
```

```
%  PSATMat.M
%  Psychrometrics Tool Box
%  Calculates Saturation Vapor Pressure
%  Based on formulae provided in the ASHRAE Handbook of Fundamentals
%  Valid for range -40 C to 120 C
%  Used for temperature matrices
%  Tmatrix --    Dry bulb temperature in C
%  Pmat --    Saturation Vapor Pressure of Water in kPa
%  Pmat = PSATMat (T)
function [VPSat2] = psatmat(T)
siz = size(T);
rowmax = siz(1);
colmax = siz(2);
NMax  = rowmax * colmax;
VPSat1 = ones(rowmax, colmax);
for k = 1:NMax;
    Tn=T(k);
    if (Tn < -40) || (Tn > 120)
        Tn
        warning ('PHYSC01: Temperature out of range -40 : 120 C');
    end
    T1 = Tn + 273.15;
    if T1 >= 273.15
        VPSat1(k) = exp(-5800.2206/T1 + 1.3914993 - 0.048640239*T1 + ...
            0.41764768e-4*T1.^2 - 0.14452093e-7*T1.^3 + 6.5459673 * ...
            log(T1))/1000;
    else
        VPSat1(k) = exp(-5674.5359/T1 + 6.392547 - 0.9677843e-2*T1 + ...
            0.6221570e-6*T1^.2 + 0.20747825e-8*T1.^3 + ...
            0.9484024E-12*T1.^4 + 4.1635019* log(T1))/1000;
    end
end
VPSat2 = VPSat1;
```

```
% RFlatPlateForced.m
% Calculates the resistance of a flate plate or leaf to the convective
% transfer of heat.By providing an optional Lewis #, resistances for mass
% or momentum transfer can be calculated as well.
% r = RFlatPlateForced(visc, windspeed, Prandtl, lengthscale , k, Lewis)
% k is thermal diffusivity m2/s
% Lewis # is ratio(k/Di), where D is the diffusivity of species I
% Pr = Prandtl #, typically 0.72 in air
% Source: Principle of Environmental Physics, Monteith & Unsworth, 1990.
% Resistances are outputed as s/m.
function r = RFlatPlateForced(visc, windspeed, Pr, lengthscale , k, Lewis)
Re = windspeed .* lengthscale ./ visc;      % Reynold's #
if Re <= 10^4              % laminar, transition happens sooner(roughness)
    Nu = 0.68 * Re .^0.5 * Pr .^(1/3);      % Nusselt #
else                       % turbulent
    Nu = .036 * Re .^0.8 * Pr .^(1/3);
end

if nargin == 6
    Nu = Nu * Lewis ^ (1/3);         % Convert Nusselt # to a Sherwood #
end
r = lengthscale./(k * Nu);
```

```
% RFlatPlateNatural.m
% Calculates the resistance of a flate plate or leaf to the convective
% transfer of heat.By providing an optional Lewis #, resistances for mass
% or momentum transfer can be calculated as well.
% r = RFlatPlateForced(visc, deltaT, T, Pr, g, lengthscale , k, Lewis)
% k is thermal diffusivity m2/s
% Lewis # is ratio(k/Di), where D is the diffusivity of species I
% Pr = Prandtl #, typically 0.72 in air
% Source: Principle of Environmental Physics, Monteith & Unsworth, 1990.

function RFlatPlateNatural(visc, deltaT, T, Pr, g, lengthscale , k, Lewis)
Gr = g * deltaT . *lengthscale./(T.* visc^2);      % Grashof #
Gr
if Gr <= 10^8              % laminar, transition happens sooner(roughness)
    Nu = 0.55 * (Gr * Pr) .^0.25;   % Nusselt #, natural convection
else                       % turbulent
    Nu = 0.13 * (Gr * Pr) .^(1/3);
end

if nargin == 6
    Nu = Nu * Lewis ^ (1/4);         % Convert Nusselt # to a Sherwood #
end
RFlatePlateNatural = lengthscale./(k * Nu);
```

```
% TSky.m
% Calculates sky temperature for use in longwave radiation calculations
% SkyTemp = TSky (T_ambient, T _ dewpoint)
% Temperatures are expressed in degrees Celsius
% Clear sky emissivities calculated using Walton's method
function SkyTemp = TSky(T_ambient, T_dewpoint)
e_sky = .787 + .764 * log((T_dewpoint + 273)/273);
T_clearsky = (T_ambient + 273) * e_sky ^ (1/4);
SkyTemp = T_clearsky - 273;
```

```
% VPP.m
% Psychromterics Toolbox
% Calculates H20 vapor partial pressure of H20 in kPa
% T     -- dry-bulb temperature in C
% RH    -- relative humidity from 0 to 1.0
% PP    -- partial pressure of vapor in kPa
% PP = VPP(RH, T)
% Temperatures must be expressed in row vectors.
function [PP] = VPP (RH, T)
if (RH<0) || (RH>1.0)
    RH
    warning('PSYCH02: FATAL Rel. Humidity value not in range 0.0 and 1.0');
end
PS = psat (T);
PP = RH .* PS;
return
```

```
% W2RH.m
% Psychrometric Toolbox
% Converts humidity ratio to relative humidity
% RH -- Ratio between partial pressure of H20 vapor in the air, and
%       p.p at saturation.
% W --  Ratio between mass ratio kg/kg of H20, and mass ratio at
%       saturation
% P --  Atmospheric air pressure in kPa (P)
% T --  Air temperature in C
% RH = W2RH (W, T, P)

function [RH] = W2RH (W, T, P)
    DG = DEGSAT (W, T, P);
    RH = DS2RH (DG, T, P);
return
```

```
% W2VP.m
% Psychromterics Toolbox
% Calculates H20 vapor partial pressure of H20 in kPa
% T     -- dry-bulb temperature in C
% W     -- humidity ratio (kg/kg)
% rho   -- specific density of air
% PP = W2VP(T, W, rho)
% Temperatures must be expressed in row vectors.
function [PP] = W2VP (T, W, rho)
% if (RH<0) || (RH>1.0)
%     RH
%     error('PSYCH02: FATAL Rel. Humidity value not in range 0.0 and 1.0');
% end
R = 8.314;   %J/mol/K
M = 1000/18.00;  %per kg H2O
Rbar = 0.4619;    %J/kg/K
T2 = T + 273.15;
PP = rho*W/M*R*T2;
return
```

```
% wetbulbs.m
% Finds the wet bulb temperature for a given dry bulb temperature in C,
% relative humidity in and air pressure in kPa.
% Valid for range from -50C to 120C
% Arguments must be supplied as column vectors or matrices OF EQUAL SIZE
% for evaluation of multiple pts.
% TWetBulb =  wetbulb(Tdrybulb, RelH, AirPressure, gamma)
% Tdrybulb in C, Rel H as a fraction. Air Pressure in kPa. Gamma is the
% psychrometer constant.
% Created by Jamison Hill 5-25-2005: Updated 2-25-06
function [Twetbulb] = wetbulb (Tdrybulb, RelHs, AirPressure, Psy)
global Td VPAir AirP gamma
if nargin == 4
    gamma = Psy;
else
    gamma = 6.66E-4;     %Psychrometric constant per degree K(20C)
end
AirP = AirPressure;
% Determine Size of Input Matrices
siz = size(Tdrybulb);
rowmax = siz(1);
colmax = siz(2);
Twetbulb = Tdrybulb .* ones(rowmax, colmax);
NMax = rowmax * colmax;
for J = 1:NMax
    Td = Tdrybulb(J);
    RH = RelHs(J);
    VPAir = VPP(RH, Td);
    if RH > 0.25
        Tguess = (Td + dewpoint(Td, RH))/2;
    else
        Tguess = Td;
    end
    Twetbulb(J) = fzero(@PsychrometricEq, Td, Tguess);
end

% Psychrometric Equation
function delta = PsychrometricEq(Tw)
global Td VPAir AirP
gamma = 6.66E-4;             %Psychrometric constant per degree K
delta = VPAir - psat(Tw) + gamma .* AirP .* (Td-Tw);
```

```
% WSconvert.m
% 11-24-05
% Wind Speed Converter
% Uses empirical power law relationship to convert from one wind speed to
% another
% (WSnew/WSref)=(Znew/Zref)^a
% WSnew --- New wind speed at point of interest
% Znew  --- Height of point of interest
% WSref --- Ref station wind speed
% Zref  --- Ref station height (typically 2m)
% a     --- exponent, depends on surface roughness.
%           = 1/7 for smooth flat plate, or ice
%           = 1/5 hilly terrain
%           = 1/4 rough urban terrain
%
% WSnew = WSConvert(WSref, Href, Hnew, a)

function WSnew = WSConvert (WSref, Zref, Znew, a);
WSnew = WSref * (Znew/Zref).^a;
```

SOLAR RADIATION TOOLBOX

```
% declination.m
% Jamison Hill
% 12-20-05
% Solar Declination Calculator
% Calculates angle of solar declination for given calender day
% dec = declination(day)
% day is between 1 & 365: where 1 is Jan 1 and 365 is Dec 31.
%
function dec = declination (day)
A = sin (356.6+ .9856.*day);
dec = asind(0.39785.* sind(278.97 + .9856 .*day + 1.9165 .*A));
```

```
% fDiffuse.m
% Jamison Hill
% Created: 1-10-06
% Calculates the fraction of solar radiation in the difuse form for a given
% clearness index.
function fractdif = fDiffuse (kI)
fractdif1 = 0.9511 - 0.1604 .* kI + 4.388 .* kI .^2 - ...
    16.638 .* kI .^3 + 12.336 .* kI .^4;    % for 0.22 < k > 0.8
% Correct for clear skies, non standard values
for i = 1:length(kI)
    if kI(i) >= 0.8
        fractdif1(i) = 0.1653;
    end
end
A = (kI < 0) | (kI > 1);
fractdif = fractdif1;
% if ~isempty(A)
%          warning('Invalid clearness index');
% end
% The following correction is insignificant and hence was removed
%       if kI(i) <= 0.22
%         fractdiff(i) = 1 - .09* kI(i);
%     end
% size Global
% size fractdiff
% size kI
%DiffuseR = fractdiff .* Global;
%DirectR  = (1-fractdiff) .* Global;
```

```
% HourAngleCorrect.m
% Jamison Hill
% 12-20-2005
% Converts clock time to solar hour angles:
% Hour angles are defined as 15 degrees * # hrs from noon.
% Positive angles correspond to the afternoon. Negative angles correspond
% to the morning.This function takes in account the effects of time zones,
% and the equation of Time (analema).
%
% HrSolar = NoonAngleCorrect(HrClock, day, deltaLong);
%
% HrSolar    -- solar hour angle(degrees)
% HrClock    -- clock time in hours (1-24)
% day        -- calendar day (Jan 1 = 1, Dec 31 = 365)
% deltaLong  -- distance in deg. longitude W of the standard
%            meridian, see table below.
%
% Location of Standard Meridians
% Time Zone        Degrees west of Greenwich
% Atlantic                   60
% Eastern                    75
% Central                    90
% Mountain                  105
% Pacific                   120
% Alaska                    135
% Hawaii                    150


function HrSolar = HourAngleCorrect (HrClock, day, deltaLong);
%Equation of Time
f = 279.575 + .9856 .* day;
A = -104.7 .* sind(f)+596.2 .*sind(2 .*f)+4.3 .*sind(3 .*f)-12.7 .*sind(4 .*f);
B = 429.3 .* cosd(f)+2.0 .*cosd(2 .*f)-19.3 .*cosd(3 .*f);
ET = (A-B) ./ 3600;      % # of hours diff between clock noon(12)
                         % and solar noon.
HrSolar = (HrClock - (12 - ET - deltaLong./15)) .* 15;
```

```
%Solaraltitude.m
% Calculate solar altitude angle given declination, latitude, and hour angle.
% altitude = solaraltitude (declination, latitude, hrangle)
% all angles expressed in degrees.
function altitude = solaraltitude (declination, latitude, hrangle)
altitude= asind (sind(latitude).*sind(declination) + ...
    cosd(latitude).*cosd(declination).*cosd(hrangle));
```

```
% SolarET.m
% Calculate extraterrestrial solar radiation on a horiz surface
% ET = SolarET (altitude)
% altitude angle in degrees
% ET =  Solar Constant[1360 W/m^2] * sin alt

function ET = SolarET (altitude)
SolarConstant = 1360;   % W/m^2
ET = SolarConstant * sin (altitude);
```

```
% SplitBeam.m
% Jamison Hill
% Created: 1-10-06
% Divides the solar beam into diffuse and direct components given
% global radiation and hourly clearness index.
% Source of diffuse model
% D.G. Erbs, S.A. Klein and J.A. Duffie, "Estimation of the diffuse
% radiation fraction for hourly,daily and monthly average global radiation"
% Solar Energy, 28(4), 293-304, 1982.


% [Diffuse Direct] = splitbeam(GlobalRad, kI)
function [DiffuseR DirectR] = splitbeam (Global, kI)
fractdiff = 0.9511 - 0.1604 .* kI + 4.388 .* kI .^2 - ...
    16.638 .* kI .^3 + 12.336 .* kI .^4;    % for 0.22 < k > 0.8
% Correct for clear skies, non standard values
for i = 1:length(kI)
    if kI(i) >= 0.8
        fractdiff(i) = 0.1653;
    end
end
A = (kI < 0) | (kI > 1);
% if ~isempty(A)
%         warning('Invalid clearness index');
% end
% The following correction is insignificant and hence was removed
%       if kI(i) <= 0.22
%         fractdiff(i) = 1 - .09* kI(i);
%     end
% size Global
% size fractdiff
% size kI
DiffuseR = fractdiff .* Global;
DirectR  = (1-fractdiff) .* Global;
```

PLANT GROWTH TOOLBOX

```
% distributebiomass.m
% Allocates biomass to stem, root, and leaf pools.
% Usage: [stem root leaf] = distributebiomass(total_biomass, rootfract, ...
%         K, SLA, Pipe)
% SLA   = specific leaf area
% K     = allometric coefficient
% Pipe  = pipe model coefficient
% stemBiomass = K * stemArea ^ X
% X assumed to be 3/2 in this case.

function [stem root leaf] = distributebiomass (total_biomass, rootfract,...
        K, SLA, Pipe)
    root = total_biomass * rootfract;
    leftover = total_biomass - root;
    leaf = fzero(@(L) total(L, Pipe, SLA, K) - leftover, [0 leftover]);
    stem = leftover - leaf;
end

% Stem Distribution function
function stem = calcstem(leaf, Pipe, SLA, K)
    Area = leaf * Pipe/SLA;
    stem = Area^ (3/2) * K;
end

% Total biomass distribution function
function B = total(leaf, Pipe, SLA, K)
    B = leaf + calcstem(leaf, Pipe, SLA, K);
end
```

```
% Farquhar.m
% Created: 1-3-2006
% by Jamison Hill
%
% DESCRIPTION
% The Farquhar, von Caemner,and Berry mode or Farquhar for short is
% a mechanistic representation of C3 photosynthesis.
% It was first described by Farquhar et al. in their paper:
% "Towards a biochemical model of photosynthesis". 1980. Planta 149: 78-90.
% Although originally conceived for the chloroplast level, it has been used
% succesfully in numerous ecological models to describe the effects of
% temperature, CO2 concentration, and irradiance on whole leaves
% and even canopy photosynthesis rates.  The model is robust and well
% understood. It accounts for the effects photorespiration and the Blackman
% response at light or CO2 saturation. And it can be modified to account
% sucrose synthesis and other sink based limitations.
% Photosynthesis is described as two potentially rate limiting steps
% carboxylation(dark reaction, Jc) and electron transport
% (light reactions,Je). A third step, Js was added by Sharkey(1985)to model
% limitation by sucrose use and synthesis. The Photosynthesis rate is to
% taken to be the minimum of Je, Jc, Js, with empirical smoothing factors
% (thetas) to account for the transitional states between different
% limiting regimes.
%
% USAGE
% A = Farquhar(Ipar, CO2i, O2i, T, thetaJe2Jc, Plantdata, VcJs, theta2Js)
%
% DESCRIPTION OF PARAMETERS
% REQUIRED
% A             -- Assimilation Rate umol C fixed/m^2 leaf area /s
% Ipar          -- Incident Radiation as umol PAR/m^2/s
% CO2i          -- CO2 concentration inside the substomatal cavity
% O2            -- O2 concentration inside the substomatal cavity
% T             -- Leaf/Canopy Temperature
% Plantdata     -- data structure describing model parameters (see below)
% thetaJe2Jc    -- sharpness of transition between light & CO2 limited
%                  photosynthesis.
%
% OPTIONAL
% Js        -- Sink/ Biosynthetic limited rate of photosynthesis,
%              when enabled, sets Amax equal the rate of growth + respiration
% theta2Js  -- sharpness of transition between sink & non sink limited
% photosynthesis

% Plant Data Structure:
% Describes the kinetic parameters of photosynthesis:

%   GENERAL FIELDS
%   Tref: Reference temperature for rate constants and kinetics
%   theta :   smoothing parameter controls switching behavior between CO2
%   and light(e-transport limited photosynthesis),Value between 0 & 1.
%   1 is the Blackman response, 0 is the Michaelis-Menten type saturation
```

```
%   tau.value: CO2/O2 Specificity constant, used to calculate CO2
%   compensation point
%   tau.q:  1/K, specificity temp response


% PROCESS SPECIFIC
% CO2 limited photosynthesis
% Vcmax.value: max rate of Rubisco(C02) limited photosynthesis
% Vcmax.qinc: Increasing temp response= Eactivation / R
% Vcmax.qdec1: Declining temp response 1 = Edeactivation/ R
% Vcmax.qdec2: Declining temp response 2 = Sdeactivation/R
%
% Light Reaction limited photosynthesis
% qeff:       initial quantum efficiency
% Jmax.value: saturation rate of electron transport
% Jmax.qdec1: Declining temp response 1 = Edeactivation/ R
% Jmax.qdec2: Declining temp response 2 = Sdeactivation/R

% Additional Rubisco Kinetic Parameters
% kO2.value:   Photorespiration, O2 Michaelis constant, mmol/mol
% kO2.q:       O2 Michaelis constant temp response Leuning/Farquhar
% kCO2.value:  CO2 Michaelis constant, umol/mol
% kCO2.q:      CO2 Michaelis constant temp response

% All parameters are measured in umol fixed/s at are determined at T= 25C
% R is the ideal gas constant



function A = Farquhar (Ipar, CO2i, O2i, T, thetaJe2Jc, Plantdata, Vc2Js, theta2Js)
if nargin < 6
   error('Not enough input arguments');
end
Tref = Plantdata.Tref;      % reference temp. in C

% CO2 compensation point
X = (T-Tref) .* Plantdata.tau.q1 +(T-Tref).^2 * Plantdata.tau.q2;
Y = X + 1;
tau = Plantdata.tau.value .*Y;
lambda = 0.5 .* O2i / tau;

% Rubisco (CO2) limited reactions
Vcmax = Plantdata.Vcmax.value .* kT2(T, Tref,Plantdata.Vcmax.qinc, ...
Plantdata.Vcmax.qdec1, Plantdata.Vcmax.qdec2);
tau = Plantdata.tau.value;

kO2  = Plantdata.kO2.value  .* kT1(T, Tref, Plantdata.kO2.q);
kCO2 = Plantdata.kCO2.value .* kT1(T, Tref, Plantdata.kCO2.q);
JCO2 = VCO2 (CO2i, Vcmax, lambda, O2i, kO2, kCO2);

% Electron transport (Light) limited reaction
Jmax = Plantdata.Jmax.value .* kT2(T, Tref, Plantdata.Jmax.qinc, ...
```

```
Plantdata.Jmax.qdec1, Plantdata.Jmax.qdec2);
Je = VLight(JLight(Ipar, Plantdata.qeff, Jmax), CO2i, lambda);



% Co-limitations according to Collatz et al 1991
Abar = Quadhyp (thetaJe2Jc, JCO2, Je);


if nargin == 6
    A = Abar;
elseif nargin == 8
    % Sink Limited: Sucrose synthesis / RuBP regen. limited
    JS =  Vcmax / Vc2Js;
    A = Quadhyp (theta2Js, JS, Abar);
end
```

```
% InitPlant.m
% Created 1-12-2005
% Jamison Hill
% Creates lookup tables for photosynthesis, respiration, and maybe stomatal
% conductance that are used to optimize the plant process.  Computational
% time spent creating lookup tables will save time later when running model

% Setup Ranges
fprintf('\n');
fprintf('Generating Lookup Tables......\n');
fprintf('Photosynthesis......');

CO2range = linspace(0,2000,50);         % ppm of CO2
Lightrange = linspace(0,2000,50);    % micromolar
temprange = linspace(5,45,50);          % degree C
[CO2, Light, T] = ndgrid(CO2range, Lightrange, temprange);
% Elucidate Initial Parameters
Plantdata = Plant.Photosynthesis;
Respiration = Plant.Respiration.Leaf;
theta = Plantdata.theta;
O2 = Settings.Climate.O2conc;

% Dark Respiration
Rd = Respiration.value * kT1(T, Respiration.Tref, Respiration.qinc);
theta1 = Plantdata.theta;
% If enableTrioseLimitation
    %theta2 = PlantData.thetaTPU;
    % Pnet = Farquhar(Light, CO2, O2, T, theta1, Plantdata, theta2);
netPhotosynthesis = Farquhar(Light, CO2, O2, T, theta1, Plantdata) - Rd;
PhotosynthesisLookupTable.temp   = temprange;
PhotosynthesisLookupTable.CO2    = CO2range;
PhotosynthesisLookupTable.Light  = Lightrange;
PhotosynthesisLookupTable.Values = netPhotosynthesis;
fprintf('DONE\n');

fprintf('Respiration......');
StemrespirationTable.temp = temprange;
StemrespirationTable.Values = Q10Metabolism(Plant.Respiration.Wood.value,...
              Plant.Respiration.Wood.Q10,temprange,Plant.Respiration.Tref);
RootrespirationTable.temp = temprange;
RootrespirationTable.Values = Q10Metabolism(Plant.Respiration.Root.value,...
              Plant.Respiration.Root.Q10,temprange,Plant.Respiration.Tref);
fprintf('DONE\n');

fprintf('Light Interception....');
DirectTable.angle = linspace(0,pi*(87/180),87);      %singularity at 90 deg.
DirectTable.Values = kdirect(DirectTable.angle, Plant.Canopy.Structure);
DiffuseTable.LAI = linspace(0.1,10.1,40);
DiffuseTable.Values = kdiffuse(DiffuseTable.LAI, Plant.Canopy.Structure);
fprintf('DONE\n');
clear Rd CO2 Light T netPhotosynthesis temprange Plantdata
```

```
% JLight.m
% Equation for light(electron-transport) limited photosynthesis rate.
% This equation is based upon the Farquhar-von Caemner model of
% photosynthesis. Equation derived from an implementation by Harley and
% Tenhenunen 1991.
% Source: "Modeling Crop Photosynthesis-- from Biochemistry to Canopy."
% CSSA Special Publication no. 19.
% I     = irradiance(microeinstein)
% abs   = canopy absorptivity
% qeff  = theoretical quantum efficiency (mol CO2 fixed/mol photon),
%       usually .06
% Jmax  = maximum rate of electron transport (umol/m2/s)
% J = JLight (abs*I, qeff, Jmax)
function J = JLight (I, qeff, Jmax)
J = (I.* qeff) ./ sqrt(1 + qeff^2 .* I.^2 ./ Jmax.^2);
```

```
% JLight2.m
% Equation for light(electron-transport) limited photosynthesis rate.
% This equation is based upon the Farquhar-von Caemner model of
% photosynthesis. Equation derived from an implementation by Leunig et al,
% 1992
% add citation
% I     = irradiance(microeinstein)
% abs   = canopy absorptivity
% qeff  = theoretical quantum efficiency (mol CO2 fixed/mol photon),
%       usually .06
% Jmax  = maximum rate of electron transport (umol/m2/s)
% theta = curvature (0 = Michaelis-Menten, 1 = Blackman), typically 0.9
% J = JLight (abs*I, qeff, Jmax, theta)
function J = JLight2 (I, qeff, Jmax, theta)
J = Quadhyp (theta, I*qeff, Jmax);
```

```
% kDiffuse.m
% Calculates Diffuse transmissivity coeff, by numerically integrating
% kdirect from angle 0 to pi/2(nadir)
% kd = kdiffuse(LAI,x)
% LAI = leaf Area Index vector
% x =  canopy structure parameter, see kdirect
function Kd= kdiffuse(LAI, x)
global LI                               % Leaf Area index
global X
X = x;
Kd = ones(length(LAI),1);
for i = 1:length(LAI)
    LI = LAI(i);
    taudiff = 2.* quad(@trans,0,pi/2);
    Kd(i) = -log(taudiff)/LI;
end
end


function tau= trans(angle, LAI)         % transmissivity
global LI
global X
tau = exp(-kdirect(angle,X).*LI) .* sin(angle) .* cos(angle);
end
```

```
% kdirect.m
% direct extinction coefficient
% kd = kdirect(angle,x )
% angle = solar altitude in radians
% x is leaf angle distribution parameter:
% 0 for vertical
% 1 for spherical distributions(default)
% Inf for horizontal
% if x is ommitted, spherical distribution assumed.
% Formula for ellipsoidal (x ~= 1) leaf angle distributions taken from
% Campbell
function K = kdirect(angle,x)
if nargin < 1
    error ('Not enough input arguments.');
end
if nargin == 1 || x == 1
    K = 1./(2*cos(angle));
else
    if isinf(x)
        K = 1;
    else
        K = sqrt(x.^2 + (tan(angle)).^2) / ...
            (x + 1.7774 .*(x + 1.182).^-0.7333);
    end
end
```

```
% kT1.m
% Temperature Response function for photosynthesis and respiration models.
% kT2 features a high-temperature cutoff feature, kT1 does not. For
% reactions sensitive to high temperature, use kT2
% Usage
% V_T = Vref * kT1(T, Tref, q);
% V_T  = reaction rate at current temperature
% Vref = reaction rate at ref. temperature
% q    = temperature response factor
% Tref = reference temperature (any unit: K, F, C), usually 25C
% T    = current temperature
function k = kT1(T, Tref, q)
TK = 273 + T;
TrefK = Tref + 273;
k = exp (q*(1./TrefK - 1./TK));
```

```
% kT2.m
% Temperature Response function for photosynthesis and respiration models.
% kT2 features a high-temperature cutoff feature, kT1 does not. For
% reactions sensitive to high temperature, use kT2
% Usage
% V_T   = Vref * kT2(T, Tref, qinc, qdec1, qdec2);
% V_T   = reaction rate at current temperature
% Vref  = reaction rate at ref. temperature
% qinc  = increasing temperature response factor Ha(enthalpy of
% activation / R[ideal gas constant])
% qdec1  = decaying temperature response factor (Hd/R)
% Hd = energy of deactivation
% Sv = entropy of deactivation
% qdec2  = decaying temp response factor 2, Sv/R
% Tref  = reference temperature (any unit: K, F, C), usually 25C
% T     = current temperature

function k = kT2(T, Tref, qinc, qdec1, qdec2)
TK = T + 273;
TrefK = Tref + 273;
k = exp (qinc*(1./TrefK - 1./TK)) ./ (1 + exp (qdec2 - qdec1./TK));
```

```
% Q10Metabolism.m
% Created 2-13-06
% This function used to calculate respiration rates for tissues that follow
% the Q10 model.
% respiration = Q10Metabolism(Rsref,Q10,T,Tref)
function respiration = Q10Metabolism(Rsref,Q10,T,Tref)
respiration = Rsref.*Q10.^((T-Tref)/10);
```

```
% Quadhyp.m
% In many biological processes such as photosynthesis, the rate of reaction
% is co-limited by one or more substrates. Mathematically, this can be
% represented by the following quadratic hyperbolic function:
% theta*J^2 - (J1+J2)*J + (J1*J2) = 0
% J1 & J2 refer to maximum reaction rates when the system is limited by
% single substrate or enzyme concentration.
% theta takes on a value between 0 & 1 and is called the sharpness
% parameter. It is a measurement of how sharply the reaction transitions
% from one limiting state to another. Theta = 1 corresponds to Blackman
% system, or the law of minimums. A Blackman system transitions
% discontinuously from J1 to J2 at the point where J2 = J1. Theta = 0
% corresponds to the Michaelis-Menten system. Depending on substrate
% concentration, the Michaelis-Menten system asymptotically approaches
% either the substrate limited rate (J1), or the enzyme limited rate (J2).
% Real-world photosynthetic processes in response to light and CO2 tend to
% approach the Blackman system, with theta values of 0.7 or greater.
% USAGE
% Jm = Quadhyp (theta, J1, J2)
function Jm = Quadhyp (theta, J1, J2)

    switch theta
        case 0
            Jm = (J1 .* J2)/(J1 + J2);
        case 1
            Jm = min ([J1 J2]);
        otherwise
           if (theta>0) && (theta<1)
                a = theta;
                b = J1+J2;
                c = J1.*J2;
                Jm = (b - sqrt((b.^2-4.*a.*c)))/(2.*a);
           else
                error('theta out of range (0...1)');
           end
    end
```

```
% VCO2.m
% Equation for CO2(Rubisco) limited photosynthesis rate.
% This equation is based upon the Farquhar-von Caemner-Berry (1980) model of
% photosynthesis.
% This particular implementation is derived from Collatz et al (1991)
% CO2    = CO2 concentration(stomatal)
% Vcmax  = maximum rate of carbon fixation
% lambda = CO2 compensation point
% O2     = oxygen concentration
% kO2    = Michaelis-Menten constant for oxygen
% kCO2   = Michaelis-Menten constant for CO2
% add citations for Farquhar, Collatz.
function J = VCO2(CO2, Vcmax, lambda, O2, kO2, kCO2)
J = (Vcmax .* (CO2-lambda)) ./ (CO2 + (kCO2 .* (1 + O2 ./ kO2)));
```

```
% VLight.m
% 1-3-2006
% Equation for light(electron-transport) limited photosynthesis rate.
% This equation is based upon the Farquhar-von Caemner model of
% photosynthesis. Equation derived from an implementation by Harley and
% Tenhunen, 1991
% Source: "Modeling Crop Photosynthesis-- from Biochemistry to Canopy."
% CSSA Special Publication no. 19.
%
% J      = rate of CO2 saturated electron transport limited photosynthesis
% lambda = CO2 compensation point(ppm(umol/mol air))
% Jmax   = maximum rate of electron transport
% V = VLight (J, CO2, lambda)
function V = VLight (J, CO2,lambda)
V = J .* (CO2 - lambda) ./ (4 .* CO2 + 8 .* lambda);
```

UNIT CONVERSION TOOLBOX

```
% convBtuhrft2FtoWm2K.m
% Converts heat flux from Btu/ft^2-hr to W/m^2.
% Wm2 = convBtuhrft2FtoWm2K (Btuhrft2)
% Created by Jamison Hill 5-23-2005
% ------------------------------------------------
function Wm2 = convBtuhrft2FtoWm2K (Btuhrft2)
Wm2 = .293 * Btuhrft2/.093/(5/9);

% convUvalCust2SI.m
% Converts U-Values from customary units(Btu/hr-degF-ft^2) to SI Units
% (W/m^2-K).
% U_SI = convUvalCust2SI (U_cust)
function U_SI =  convUvalCust2SI (U_cust)
U_SI = 5.8182*U_cust;

% convft2m.m
% created by Jamison Hill on 5-17-05
% Converts lengths from feet to meters
% Length_m = convft2m (Length_ft)
function Length_m = convft2m (Length_ft)
Length_m = 0.3408 * Length_ft;

% convFtoC.m
% Converts temperature from Fahrenheit to Celsius.
% Usage:
% T_degF = convFtoC(T_degC)
function Temp_C = convCtoF(Temp_F)
Temp_C = (Temp_F-32)*5/9;

% convCtoF.m
% Converts temperature from Fahrenheit to Celsius.
% Usage:
% T_degF = convCtoF(T_degC)
function Temp_F = convCtoF(Temp_C)
Temp_C = (Temp_F-32)*5/9;

% convkcal2kJ.m
% Coverts kilocalories to kilojoules
% kJ = convkcal2KJ (kcal)
% 7-20-05
function kJ = convkcal2kJ (kcal)
kJ = 4.184*kcal;

% convLyhr2Wm2.m
% Converts solar radiation from langleys per hr (calories/cm^2)
% to SI units (W/m^2)
% Flux_Wm2 = convLyhr2Wm2 (Flux_Lyhr)
% Created by Jamison Hill 5-23-2005
% ------------------------------------------------
function Wm2 = convLyhr2Wm2 (Lyhr)
Wm2 = Lyhr * 11.522;
```

```
% convmph2mps.m
% Converts wind speed from mph to mps.
% Usage:
% Speed_mps = convCtoF(Speed_mph)
function Speed_mps = convmph2mps(Speed_mph)
Speed_mps = Speed_mph* .44704;
```