Interactive Calibration from Flux Tower Data for Earth System Model Calibration: Draft Requirements

K. Arthur Endsley

November 1, 2019

1 Overview and Definitions

1.1 Plain Language Summary

Earth system models are computer simulations of the natural world. The Soil Moisture Active/ Passive (SMAP) Level 4 Carbon (L4C) data product is produced by such a simulation. We simulate how both plants and soil respond to changes in temperature and moisture. The simulation is run for the entire globe each day.

In order for the computer simulation to accurately estimate how plants and soil respond to temperature and moisture, we need information on how specific plants in different parts of the world grow and change. This information comes partly from expert knowledge and published papers—we put this information directly into the model. The rest of the required information has to be learned from data. This is the *calibration process*, in which we use real-world data from observations of how plants grow and change to "teach" the model how it should simulate plant growth and change.

The calibration process has to be supervised by a human. It also has to be run once for each different type of plant (or different collections of plants) we want to simulate. Therefore, we need another computer program to help guide the decision that a human operator has to make when running the calibration. This *calibration software*, in addition to being able to guide the user, needs to include all the same behaviors of the computer simulation so that we can compare the simulated results to the data we collected.

In summary, the calibration software should be able to read in real-world data on plant growth and change; show the user how the model currently predicts plant growth and change; update model behavior based on the data; show the user the results of changing the model to better match the data; output a table of results that are used to run the model after calibration (in a separate computer program); and calculate how well the model predicts part of the real-world data that was not used in calibration.

1.2 Definitions

- **Soil Moisture Active/ Passive (SMAP) satellite, mission**: Variously refers to the satellite itself or to the mission, the latter includes the scientific objectives of launching the satellite

- **Level 4 product:** The National Aeronautics and Space Administration (NASA) defines different levels of data products that are produced for scientific missions. Level 4 products are "model-enhanced" products, i.e., the data that are collected (by a satellite) are made more valuable by combining them with a mathematical model or computer simulation that enhances their accuracy, enhances their resolution or precision, adds additional variables or richness, or any combination of these.

- **Level 4 Carbon (L4C):** The SMAP Level 4 Carbon product contains daily estimates of global ecosystem productivity, including net ecosystem exchange (NEE), gross primary production (GPP), heterotrophic respiration (R_H), and soil organic carbon (SOC), along with quality control metrics. *See "Physics Definitions" for more information.*

- **Biome Properties Lookup Table (BPLUT):** This is a table of parameters specific to a given biome or Plant Functional Type (PFT); biome and PFT are equivalent, in this context. **This table is one of the primary outputs of the calibration procedure.**

1.2.1 Physics Definitions

Read in order, this can be considered a decent "crash course" in the physics of our Earth system model.

- **Gross Primary Production (GPP)** Generally, plants absorb carbon dioxide (CO_2) from the atmosphere; they also emit *some* CO_2 to the atmosphere as part of cellular respiration (i.e., plants also *breathe* out CO_2, just like animals). How much carbon that plants absorb and store in their tissues is their GPP; it usually is measured in "grams of carbon absorbed per unit area per unite time;" i.e., how much carbon a certain area (filled with plants) stored in a certain amount of time.

- **Heterotrophic Respiration ($\mathbf{R}_H$):** *Heterotrophs* are anything that has to eat something else in order to survive. Here, it refers specifically to soil microbes—microscopic bacteria and fungi that decompose plant matter. During decomposition, they release CO_2 to the atmosphere. This release is referred to as *respiration*, just like animals and plants respire CO_2. (Plant respiration is referred to as *autotrophic respiration* because *autotrophs* create their own food.)

- **Autotrophic respiration ($\mathbf{R}_A$):** As just discussed, autotrophic respiration is the CO_2 emitted by plants to the atmosphere as part of their "breathing."

- **Net Primary Production (NPP):** NPP is the amount of carbon assimilated net of the carbon that was lost due to autotrophic (plant) respiration (R_A). That is, we know

that plants don't keep 100% of the carbon they assimilate, because they have to respire some of that carbon to the atmosphere as part of R_A. So NPP can be calculated as: $NPP = GPP - R_A$.

- **Ecosystem respiration (RECO):** We sometimes refer to the sum of heterotrophic and autotrophic respiration as RECO: the total amount of CO_2 that is being emitted by plants and soil microbes.

- **Net Ecosystem Exchange (NEE):** Because plants also emit CO_2, and because soil microbes emit CO_2, we are very interested in the *balance* between CO_2 emitted to the atmosphere vs. CO_2 absorbed by plants and soil. This is what we call NEE: the net amount of CO_2 that is absorbed vs. emitted. You can think about NEE like the balance of your bank account: When you receive money (e.g., in a paycheck), you are storing that money and your bank balance goes up; however, you also have to pay for things, so sometimes money goes out and your bank balance goes down. NEE is like your bank balance, except it uses carbon instead of money. The relationship between these variables can be written as:

$$\begin{aligned} NEE &= (R_A + R_H) - GPP && (1)\\ &= RECO - GPP && (2)\\ &= R_H - NPP && (3) \end{aligned}$$

  When NEE has a negative sign, it means that we are storing carbon in plants (because then GPP must be larger than the sum of R_A and R_H). When NEE has a positive sign, it means that we are, on balance, emitting more carbon to the atmosphere than we are storing.

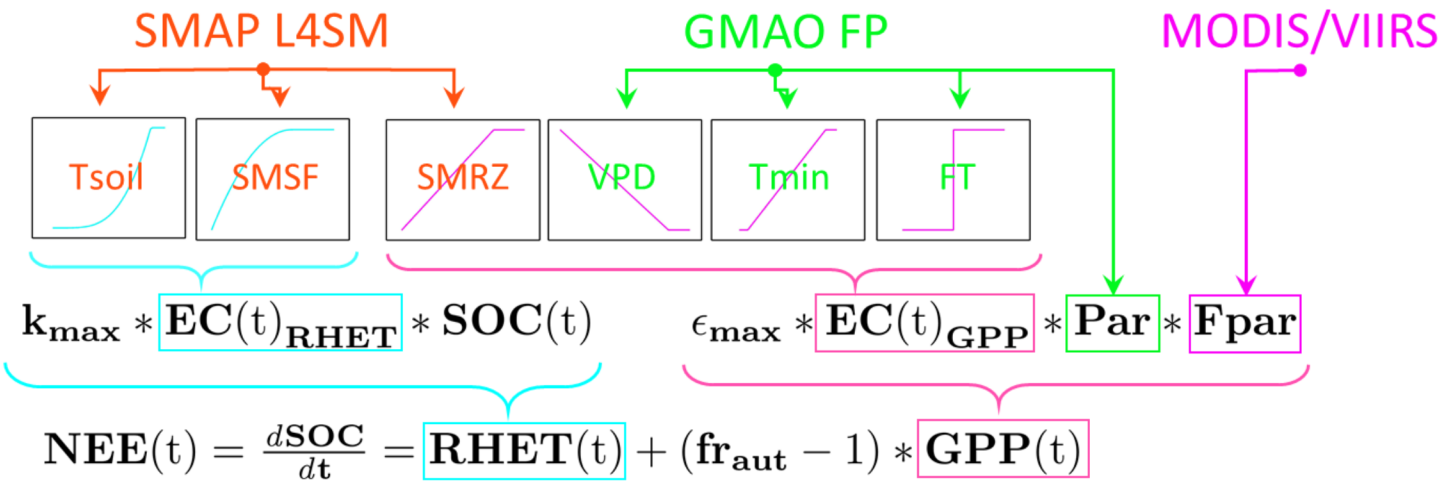

Figure 1: Overview of the SMAP L4C algorithm. The boxes indicate inputs to the algorithm, all of which are calibrated by choosing the minimum and maximum values of the linear ramp function. Soil temperature (T_{soil}) has a special Arrhenius function to define its relationship to heterotrophic respiration (RHET, here), which is why it appears as an exponential curve. Surface soil moisture (SMSF) is one variable that only has a minimum value in the ramp function; there is no upper constraint on RHET with increasing SMSF. Freeze-thaw (FT) status is a binary constraint: if the ground is frozen, there can be no GPP (that's why this constraint appears as a step function).

2 Concepts

2.1 Input Variables to Earth System Model Calibration

2.1.1 Land Cover and Meteorological Variables

	Name	Description
PFT	Plant Functional Type	
PAR	Photosynthetically Active Radiation	Input to L4C; PAR is the energy in those wavelengths of light (from the sun) that activate photosynthesis
FPAR	Fraction of PAR absorbed	Input to L4C; fraction of total PAR absorbed by green vegetation
APAR	Absorbed PAR	Calculated from PAR and FPAR as: $PAR \times FPAR$; this quantity is what drives photosynthesis
SMSF	Surface soil moisture	Amount of moisture in the soil at the surface (approx. 0-5 cm depth)
SMRZ	Root-zone soil moisture	Amount of moisture in the soil at the root zone (i.e., able to be received by plant roots)
TSOIL	Soil temperature	
TSURF	Surface temperature	
TMIN	Minimum air temperature	
FT	Freeze/Thaw state	Basically, a 0 or 1 for whether the ground is frozen (0) or thawed (1)
VPD	Vapor Pressure Deficit	At a given temperature, the difference (in pressure) between the air's current pressure and the pressure if the air were saturated with water
PAW	Plant Available Water	

2.1.2 Flux Tower and L4C Reference Variables

These variables can come either from a flux tower site (i.e., they are our "ground truth" measurements) or as outputs from a prior L4C simulation.

	Name	Description
GPP	Gross Primary Productivity	Amount of carbon absorbed by plants
NEE	Net Ecosystem Exchange	Balance of carbon absorbed by plants versus carbon released by plants and soil; if positive, represents net carbon flux to atmosphere
RECO	Ecosystem respiration	Total respiration from plants and soil; calculated as: $NEE + GPP$ or as $R_H + R_A$ (equivalent)

2.1.3 BPLUT Variables to Optimize

These variables are probably updated (optimized) during the calibration procedure, for each PFT, but the current (or "old," as of the time of calibration) values are needed as inputs for the calibration procedure so we can quantify improvement in the model. Calculating the lower and upper limits to the **ramp functions** are one of the primary goals of calibration. See also Table 5 for the bounds on these optimized values.

Optimization		Name	Description
GPP	**LUE**	(Maximum) Light-Use Efficiency	Sometimes denoted LUE_{max}; the upper limit on light-use efficiency for a given PFT. Limiting environmental conditions reduce LUE from this theoretical upper limit.
GPP	$\mathbf{VPD}_{min}$	Lower bound on VPD ramp function	
GPP	$\mathbf{VPD}_{max}$	Upper bound on VPD ramp function	
GPP	$\mathbf{SMRZ}_{min}$	Lower bound on SMRZ ramp function	
GPP	$\mathbf{SMRZ}_{max}$	Upper bound on SMRZ ramp function	
GPP	$\mathbf{TMIN}_{min}$	Lower bound on TMIN ramp function	
GPP	$\mathbf{TMIN}_{max}$	Upper bound on TMIN ramp function	
GPP	$\mathbf{FT}_{mult}$	Multiplier for frozen conditions	
RECO	$\mathbf{f_{aut}}$	Autotrophic respiration fraction	The fraction of GPP used in autotrophic (plant) respiration
RECO	$\beta_{\mathbf{TSOIL}}$	Arrhenius soil temperature parameter	Parameter controlling the response of soil microbes to temperature
RECO	$\mathbf{SMSF}_{min}$	Lower bound on SMSF ramp function	
RECO	$\mathbf{SMSF}_{max}$	Upper bound on SMSF ramp function	
SOC	$\mathbf{R_{opt}}$	Optimal metabolic SOC respiration rate	The metabolic SOC pool is also referred to as the "fast" pool; it is the rate at which litterfall (leaves, stems) decay.

SOC	$\mathbf{k_{str}}$	Structural SOC respiration ratio	The decay rate for the structural SOC (or "medium") pool is expressed as a ratio to R_{opt}, i.e., the decay rate is: k_{str}/R_{opt}.
SOC	$\mathbf{k_{rec}}$	Recalcitrant SOC respiration ratio	The decay rate for the recalcitrant SOC (or "slow") pool is expressed as a ratio to R_{opt}, i.e., the decay rate is: k_{rec}/R_{opt}.
SOC	$\mathbf{f_{met}}$	Fraction of litterfall allocated to metabolic SOC	This is the fraction of litterfall that enters the metabolic ("fast") SOC pool.
SOC	$\mathbf{f_{str}}$	Fraction of metabolic SOC allocated to structural SOC	This is the fraction of SOC in the metabolic ("fast") pool that enters the structural ("medium") SOC pool.

2.2 Global Flux Tower Network

We use a network of global eddy covariance (EC) flux towers as the source of data to calibrate SMAP L4C. A full description of the EC method is outside the scope of this document. Suffice to say, by measuring the trace gases moving up and down at different heights (see Figure 2), we can calculate the net gas exchange between a parcel of ground (usually an area 100 to 1,000 meters in radius) and the atmosphere. Averaged over time, we can calculate the **vertical flux** of a gas over that period of time. The flux is the *net* amount of gas (by weight or mass) that has been moved either up (to the atmosphere) or down (to the terrestrial biosphere). In the latter case, we assume that such downward fluxes represent the gas being absorbed by the terrestrial biosphere.

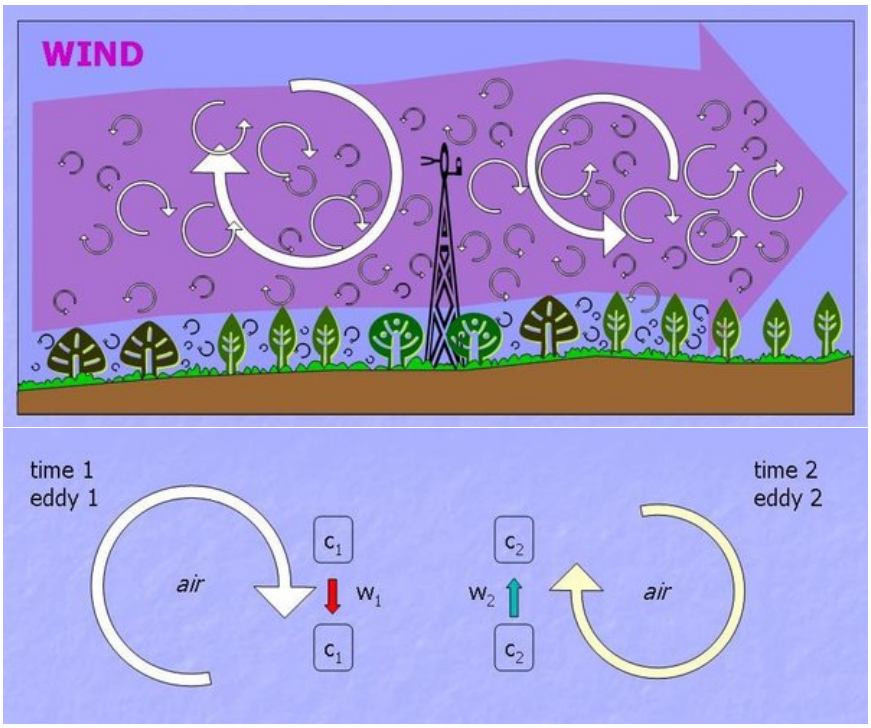

Figure 2: Cartoon diagram of an eddy covariance flux tower station. By measuring the concentrations of different gases, the overall pressure, temperature, humidity, and wind speed, it is possible to calculate the vertical flux of a gas at a given time. Figures taken from Burba, G., 2013. Eddy Covariance Method for Scientific, Industrial, Agricultural and Regulatory Applications: a Field Book on Measuring Ecosystem Gas Exchange and Areal Emission Rates. LI-COR Biosciences, Lincoln, USA.

We use measurements of the CO_2 gas exchange from a global network of flux tower stations to calibrate SMAP L4C. These measurements represent the "ground-truth" of gas exchange between the terrestrial biosphere and the atmosphere. Therefore, we use some of these data records (others are reserved for validation) to train SMAP L4C to accurately simulate such gas exchanges: how much is ultimately assimilated by plants and how much is emitted by the decay of soil organic carbon (SOC).

2.3 Plant Functional Types (PFTs)

Plant Functional Types (PFTs) are types of plants (plant communities, actually) for which the the SMAP L4C model is individually calibrated. There are 8 PFTs modeled in SMAP L4C (see Table 4). Not included are land-cover types/ PFTs that sometimes appear in certain datasets (based on the same land-cover product), such as Water (numeric code 0).

A key feature of the SMAP L4C Calibration process is the SMAP L4C algorithm is calibrated separately for each PFT. Because different types of plants respond differently, we have to calculate the parameters that govern the simulation of GPP, RECO, etc. *separately* for each PFT. That's why the calibration procedure has to be run once for each PFT (i.e., 8 times).

The *dominant* PFT within a 9-km model cell is the PFT with the most 1-km subgrid cells (there are 81 1-km cells in a 9-by-9 km grid cell). The location of flux towers is used to determine which PFT a flux tower's data represent; the PFT that is dominant in that 9-km cell is the flux tower's representative PFT (see Figure 3).

Table 4: Table of Plant Functional Types (PFTs) used in the SMAP L4C Model.

Numeric Code	Plant Functional Type (PFT)
1	Evergreen Needleleaf
2	Evergreen Broadleaf
3	Deciduous Needleaf
4	Deciduous Broadleaf
5	Shrub
6	Grass
7	Cereal Crop
8	Broadleaf Crop

2.4 Model Data Spatial Structure: EASE-Grid 2.0

Data inputs and outputs are registered on Equal-Area Scalable Earth (EASE) Grids. These are grids designed to nest perfectly within one another at a set of specified scales (see Figures 3 and 4): The 1-km and 9-km (referring to the length of one side of a square cell) grids are those used in SMAP L4C data inputs and data products:

- The land-cover or Plant Functional Type (PFT) data are described, at their smallest resolution, on a 1-km EASE-Grid. The *dominant* PFT at 9-km scale is the PFT with the majority of 1-km subgrid cells (there are 81 1-km cells within a 9-km cell). The dominant PFT of a 9-km cell determines which flux tower sites are selected for calibrating a given PFT (see Figure 3).

- The SMAP L4C model runs at 1-km scale. However, because posting the data products at 1-km scale would require 81 times as much storage space, per variable per (simulation) day, as posting data at 9-km scale, the data are spatially aggregated to 9-km grid cells.

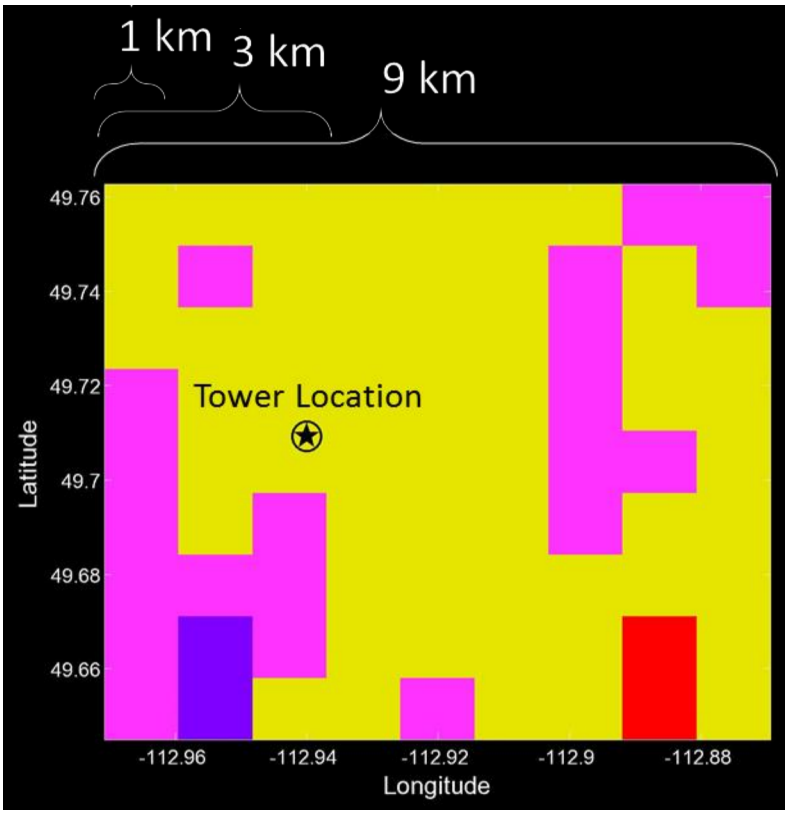

Figure 3: Example of a flux tower site's location on a 1-km land-cover (PFT) grid and within a larger 9-km grid cell. The 9-km grid cell is made up of multiple different PFTs but the "yellow" PFT is the dominant PFT.

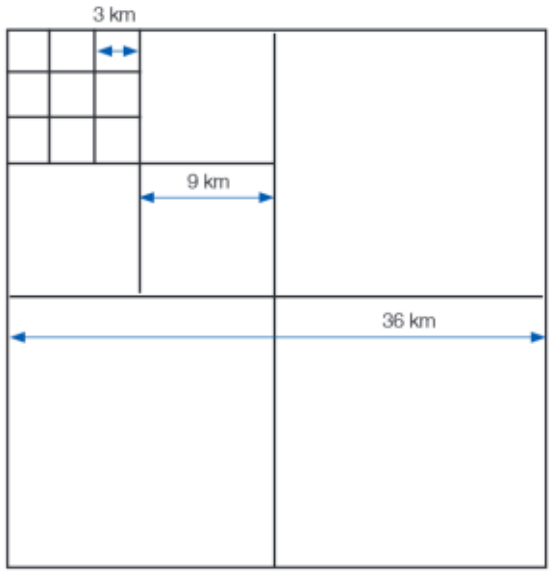

Figure 4: Example of nested EASE-Grid 2.0 grids from 3 to 36 km.

2.5 L4C Model Logic

L4C is based on a *light-use efficiency (LUE) model.* LUE models have a core assumption that under ideal (non-limiting) conditions (i.e., plants have the perfect amount of light, water, and nutrients), there is a fixed, upper-limit to the rate of photosynthesis. The photosynthetic rate is essentially a rate of conversion of light and carbon dioxide gas into stored (assimilated) carbon material. This theoretical upper-limit to LUE is a fixed value defined for each unique Plant Functional Type (PFT).

2.5.1 Model Calculation of GPP

The formula below describes how GPP is calculated in the L4C model. Here, ε_{max} corresponds to the theoretical upper limit on LUE.

$$GPP = APAR \times \varepsilon_{max} \times E_{mult} \tag{4}$$

$$= (FPAR \times PAR) \times \varepsilon_{max} \times E_{mult} \tag{5}$$

E_{mult} is a multiplier, between 0 and 1, that represents the real-world environmental conditions that affect plant photosynthesis. When $E_{mult} = 0$, photosynthesis cannot occur; when $E_{mult} = 1$, photosynthesis proceeds at the maximum rate (maximum LUE or ε_{max}). Thus, E_{mult} is basically the fraction of the maximum LUE rate that we expect under the given environmental conditions. **Those environmental conditions determine the value of E_{mult} according to the formula below.**

$$E_{mult} = f(VPD)\, f(TMIN)\, f(SMRZ)\, FT_{mult} \tag{6}$$

Each $f(*)$ function above is a simple linear ramp function. That is, if you imagine an X-Y plot and define lower and upper bounds on the vertical axis as $[0, \varepsilon_{max}]$, the left and right bounds on the horizontal axis will describe a unique linear ramp from 0 to ε_{max}. An example of what such a plot might look like is given in Figure 5. (You can also see some cartoon versions of the ramp functions in Figure 1.)

The ramp functions are defined by:

$$f(x) = \begin{cases} 1 & \text{if } x \geq x_{max}, \\ 0 & \text{if } x \leq x_{min}, \\ (x - x_{min})(x_{max} - x_{min})^{-1} & \text{otherwise} \end{cases} \tag{7}$$

FT_{mult} is the multiplier to use when ground conditions are frozen; it takes on the value of 1.0 when ground conditions are thawed (i.e., no constraint on GPP).

2.5.2 Model Calculation of RECO

Recall that RECO can be calculated simply from GPP and NEE: $RECO = NEE + GPP$. Therefore, once we have calculated (separately) GPP, we can calculate RECO. RECO can also be calculated as the sum of autotrophic (plant) respiration, R_A, and heterotrophic (soil) respiration, R_H:

$$RECO = NEE + GPP$$

$$= R_A + R_H$$

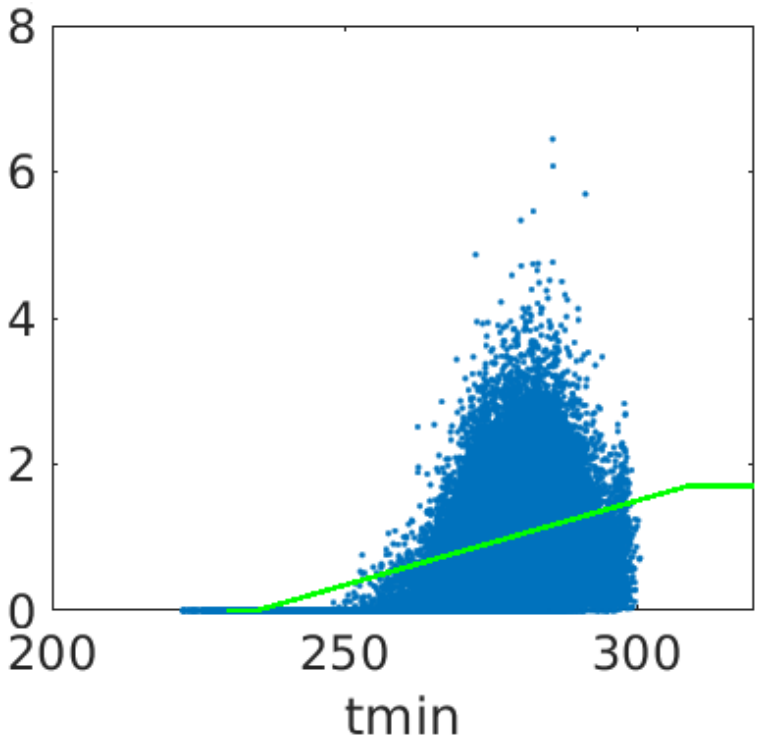

Figure 5: Example arbitrary ramp function, where the lower and upper bounds on the minimum air temperature (TMIN) describe a unique ramp on the Y-axis value from 0 (no photosynthesis) to the maximum photosynthetic or LUE rate (about 1.7 in this example).

We calculate R_A as some constant fraction of GPP (e.g., 50% of GPP is used for R_A):

$$R_A = \mathbf{f_{aut}} \times GPP$$

Then, we just need to calculate R_H; this is more complicated. There are three (3) different soil organic carbon (SOC) pools, so we have to calculate R_H in the ith pool at time t:

$$R_H(i,t) = K_{mult}\, k_i\, C_i(t) \tag{8}$$

Where:

$$K_{mult} = f(TSOIL)\, f(SMSF) \tag{9}$$

Just like with GPP, these $f(*)$ functions are dimensionless multipliers. $f(SMSF)$ is a linear ramp function between SMSF_{min} and SMSF_{max}. The $f(TSOIL)$ function is different; it is an Arrhenius equation that describes the response of soil microbes to soil temperature:

$$f(TSOIL) = \exp\left[\beta_{TSOIL}\left(\frac{1}{66.02} - \frac{1}{TSOIL - 227.13}\right)\right] \tag{10}$$

Where exp[$*$] refers to the natural exponential function (i.e., e^*).

We now have to simplify the calculation of R_H (Equation 8 because we don't know the size of the SOC pools at time $t = 0$ (which means we can't figure out R_H at time $t = 0$ or any later times). Let's call this initial carbon pool size $\bar{C}$. **A good guess for what $\bar{C}$ should be is some high value of R_H/K_{mult}.** Therefore, we might take, e.g., the 95th percentile of R_H/K_{mult}.

Putting it all together, we get:

$$RECO = ((\mathbf{f_{aut}} \times GPP) + (K_{mult} \times \bar{C})) \tag{11}$$

2.5.3 Model Calculation of SOC

So, $\bar{C}$ is needed to kick-start the calibration of RECO, and it essentially stands in for the unknown soil organic carbon (SOC) pool size. However, we want an accurate estimate of the initial SOC pool size for each flux tower so we can propagate this forward in time. We estimate initial SOC pool sizes by:

1. First, we solve analytical equations (see Equations 12, 13, and 14) that represent the steady-state of each SOC pool.

2. Second, using these initial starting estimates, we run the L4C model forward in time for a few years using the annual climatology in order to "nudge" these starting estimates towards even more accurate starting values. **This is described in Section 4.8: L4C Soil Model Forward Run.**

So, for Step (1): After some fancy math (setting the derivative of each of the differential equations that describe the time-ordered change in a pool to zero, then solving analytically), we determined that the initial state of each SOC pool is:

$$\text{Metabolic or “Fast” pool:} \quad C_{met} = \frac{f_{met} \sum NPP}{R_{opt} \sum K_{mult}} \tag{12}$$

$$\text{Structural of “Medium” pool:} \quad C_{str} = \frac{(1 - f_{met}) \sum NPP}{R_{opt}\, k_{str} \sum K_{mult}} \tag{13}$$

$$\text{Recalcitrant or “Slow” pool:} \quad C_{rec} = \frac{f_{str}\, k_{str}\, C_{str}}{k_{rec}} \tag{14}$$

Note: Equation 14 may look strange, but if you plug Equation 13 into it, then it makes sense. The sums above refer to the (annual) sum of NPP or K_{mult} over a 365-day climatology.

We also calculate a new parameter, $\bar{C}_0$ as the sum of all three starting pools:

$$\bar{C}_0 = (R_{opt} \times C_{met}) + (k_{str}\, R_{opt} \times C_{str}) + (k_{rec}\, R_{opt} \times C_{rec}) \tag{15}$$

3 Product Description

3.1 Platform and Implementation

The calibration routines should be implemented in the Python 3 programming language. Although a human user is required to guide the calibration process, ideally, major steps of the calibration procedure are implemented such that they can be invoked in a pure Python 3 environment programmatically, e.g., for developing unit tests. Suggested publicly available and well-established Python 3 packages that may be useful for implementation include: `numpy`, `scipy`, `matplotlib`, `pandas`, `h5py`, and `GDAL`.

The calibration procedure should, at a minimum, work on UN*X systems (e.g., GNU/Linux or Apple Mac OSX). Microsoft Windows support is not required. The platform can be expected to have Python 3 and a web browser (Mozilla Firefox or Google Chrome) installed. The use of Python virtual environments, through `virtualenv` or `conda`, is recommended but not required.

At a minimum, a command line interface (CLI) should be developed to guide the user through the calibration procedure. This CLI would need to periodically prompt the user for inputs and also launch plot windows (e.g., with Tkinter or Qt, in a Python 3 environment on a UN*X system).

As a stretch goal ("nice to have"), a graphical user interface (GUI) would refine the end-user experience. It is suggested that a browser-based front-end would be the easiest for development and require no additional platform dependencies. However, there are no specific language or framework requirements for the GUI. Obviously, the chosen framework will have to interface with the calibration codebase implemented in Python 3.

3.1.1 Other Technical Considerations

- The input datasets will all be formatted as Hierarchical Data Files (HDF), version 5 (HDF5).

3.2 Overview of Procedure

At the highest level, the L4C simulation has two (2) primary components:

1. **Simulation of plant productivity (GPP) and calculation of constraints (K_{mult}) on soil respiration (R_H);**
2. **Simulation of soil organic carbon (SOC) dynamics and the soil respiration (R_H) that decays SOC.**

The first part is required in order to simulate the second part.

Calibration of the L4C simulation, then, begins with two highly similar procedures:

1. **Optimization of GPP parameters;**
2. **Optimization of RECO parameters;**

Optimization of these parameters is based entirely on a constrained non-linear minimization of the sum-of-squared errors between the observed (flux tower) GPP/RECO and the model-simulated GPP/RECO.

Then, based on these optimized parameters, a series of repeated model spin-ups (vectorized calculations) and forward simulations (permutation of state variables) proceeds:

1. "Preliminary" calculations of GPP, K_{mult}, and NPP at 1-km subgrid scale are performed for a 365-day climatological year, then used to plot and quantify the linear correspondence between observed and predicted SOC stock size.

2. We *again* do a "preliminary model spin-up" over a 365-day climatological year: Calculate GPP, K_{mult}, NPP, and litterfall (daily average of NPP).

3. We perform the "analytical spin-up" whereby we calculate the initial SOC stock sizes based on a 365-day climatological year.

4. We then do a "numerical spin-up", simulating a 365-day climatological year for multiple iterations. This nudges the initial SOC parameters closer to the true values.

5. We run the "preliminary" spin-up calculations of GPP, K_{mult}, and NPP/litterfall again, this time over the full operational record (approximately 2000-2019).

6. We then do the numerical spin-up yet again, this time over the full operational record (approximately 2000-2019).

This repetition is necessary to initialize model parameters and nudge them toward their final values.

3.3 Procedure Description

We need an interactive software tool that guides a user through the process of calibrating the Earth system model for each Plant Functional Type (PFT). The result of the calibration is three (3) output files:

- An updated Biome Properties Lookup Table (BPLUT);
- Soil organic carbon (SOC), initial stock size maps: Four (4) global, 1-km maps of SOC in the "fast," "medium," and "slow," pools, plus "litterfall" pool;
- Root zone soil moisture (SMRZ) minimum and maximum for the Nature Run period (approximately 2000-2018);

In addition, the calibration procedure should produce a detailed log. The procedure is described mostly for a single PFT; the steps would need to be repeated by the user for each PFT. The procedure is outlined below. A more thorough description is provided for certain elements that are **bold and underlined** in **Section 4: Business Logic Details.**

1) **Read in a Configuration File.**

2) **Read in various datasets for the flux tower locations and compile the observational record at each tower.**

   a) Read in the complete list of flux tower sites, their location within the global 9-km grid, and the dominant land-cover.

   b) Read in the list of sites to exclude.

   c) **Low Priority:** Calculate the dominant PFT at each flux tower site, based on its position within the global 1-km PFT grid (see Figure 3).

   d) **Low Priority:** Calculate the percentage of a flux tower site's 9-km cell that is occupied by 1-km sub-cells for each PFT (i.e., the percentage of area occupied by each PFT within the 9-km cell, see Figure 3).

   e) Read in the **IGBP FAO soil organic carbon (SOC) stock data**.

   f) Read in the last-used **L4C Reference dataset**.

   g) Read in the **L4C Meteorological Input dataset**.

   h) Read in the **Flux Tower Fluxes dataset**.

   i) Compile into a single table or data structure the Nature Run data for each flux tower site for each day for each PFT (NEE, GPP, RECO) along with the error metrics (RMSE) for NEE.

   j) Compile into a single table or data structure the L4C meteorological inputs (SMRZ, SMSF, TMIN, TSOIL, TSURF, VPD, PAR, and FPAR).

   k) Compile into a single table or data structure the flux tower flux estimates (NEE, GPP, RECO).

   l) Subset all time series variables to the (user-configurable) period January 1, 2000 to December 31, 2014, inclusive.

3) **Allows the user to select a Plant Functional Type (PFT) to begin calibration (NOTE: From this step forward, the user will have to repeat the steps for each PFT.)**

4) **Subsets the flux tower data (and other datasets) to those sites that have the selected PFT as their dominant land-cover type.**

   a) For PFT 3 (Deciduous Needleleaf Forest), it is necessary to make an exception: Allow tower sites that have *any* amount of PFT 3 among the 1-km subgrid cells to be used for calibrating PFT 3. This is because PFT 3 is not dominant at 9-km scale anywhere; but this should be configurable for any PFT (i.e., if the user wants to calibrate on tower sites that contain any amount of any PFT, not just for PFT 3). **This should be a general rule, i.e., calibration cannot proceed if there are zero flux tower sites where a chosen PFT is dominant.** In such cases, the user should be prompted with an indication that calibration will proceed using those flux tower sites that have *any* amount of the chosen PFT.

b) Subset the **Flux Tower Fluxes dataset** to those stations that have the selected PFT as their dominant land-cover type. **Prompt the user with a warning if the number of tower sites for calibrating a given PFT is less than 30.** This threshold is somewhat arbitrary so it should be exposed somewhere obvious in the code base so it can be changed later.

c) Subset the **L4C Meterological Input dataset** to those sites that have the chosen PFT as the dominant land cover type.

d) Subset the last-used **L4C Reference dataset** to those sites that have the chosen PFT as the dominant land cover type.

e) Compile ancillary information on each tower site for the calibration of this PFT: Site weights, Dominant PFT (at 9-km scale), Proportion of area occupied by each PFT on the 1-km subgrid, and **the soil organic carbon (SOC) stock size for this tower site** (from IGBP FAO record).

f) Compile all data required for calibration of this PFT (i.e., for the subset of flux tower sites related to this PFT): FPAR, PAR, TSURF, TSOIL, SMSF, VPD, TMIN, SMRZ, minimum SMRZ, maximum SMRZ, and PAW. *Note that FPAR is defined separately for each PFT, i.e., there is an FPAR for PFT 1, for PFT 2, and so on.*

5) **Computes average annual climatology at subset flux tower sites: the average meteorological conditions across the PFT-subset of flux tower sites on each of 365 days for a "typical year" in the period January 1, 2000 through December 31, 2014 (prior to the year of SMAP mission launch).** (However, these start and end dates should be user-configurable for flexibility.) Steps below refer to "flux tower" variables: those variables in the **Flux Tower Fluxes dataset**.

a) For that subset of flux tower sites related to the current PFT, for each ordinal day (January 1 through December 31), average the PAR, VPD, TMIN, TSURF, TSOIL, SMRZ, SMSF, and FPAR (ignore February 29 whenever it occurs). *Note that there are 81 values of FPAR for each flux tower site for each day; this is because FPAR is defined on the 1-km subgrid. These 81 values should be averaged separately such that FPAR climatology is* 365×81 *values.*

b) Automatically remove non-physical (negative) values in flux tower GPP and flux tower RECO.

c) Guide the user through removing outliers (spikes) in the flux tower GPP: A user-configurable smoother/ low-pass filter should be used here. Currently, we use "robust spline smoothing" but are open to any comparable smoother (e.g., Savitzky-Golay filter or SciPy's `filtfilt()` implementation of the phase-invariant forward-backward filter). Perhaps the choice of smoother could be left up to the user and the results of a smoothing trial made visible in a plot. At a minimum the *parameters* of the smoother will have to be left up to the user (e.g., for the forward-backward filter, the size of the moving window, in days, will have to be a user choice) and the results of smoothing, for a small number of randomly chosen stations, should be displayed.

d) *Guide the user through the same outlier-removal for flux tower RECO.*

e) Harmonize missing data across flux tower GPP, flux tower RECO, and flux tower NEE, i.e., remove from all three variables the values at the same time index that are missing from any one variable.

6) Pre-optimization for GPP: Prepare subset flux tower data for optimizing model simulation of GPP and display current parameters to the user.

a) Read in, from the current BPLUT table, the PFT's LUE value and those values to be optimized in calibrating modeled GPP, VPD_{min}, VPD_{max}, SMRZ_{min}, SMRZ_{max}, TMIN_{min}, TMIN_{max}, and FT_{mult}; **use these as the initial values in the optimization.**

b) We're going to need to calculate and use a new quantity, APAR, for these next few steps. It is a simple calculation based on the (subset flux tower) FPAR and PAR: $APAR = FPAR \times PAR$ (see Equation 5). Allow the user to specify both lower and upper limits for APAR (cases with APAR greater than this value will be discarded and not used in the linear ramp function calculation), defaulting to the minimum and maximum calculated APAR values.

c) Calculate linear ramp function given *current* BPLUT values. This is the slope of the line on a plot of GPP/APAR (GPP divided by APAR) against VPD, SMRZ, or TMIN (separately) that is constrained to increase from 0 to LUE (or from 0 to ε_{max}, see Equations 4, 5 and 6) on the Y-axis and between the lower and upper bounds of VPD, SMRZ, or TMIN on the X-axis. **Note that the direction of this ramp function varies along with the variable:** increasing VPD should decrease GPP/APAR; increasing SMRZ or TMIN should increase GPP/APAR.

d) Display to the user the model's *current* ramp functions for VPD, SMRZ, and TMIN (an example of such a plot is shown in Figure 5). This is a plot of GPP/APAR on the vertical axis and VPD, SMRZ or TMIN on the horizontal axis, for each flux tower site for each day (see Equations 4 and 5). Allow the user to save these plot(s) as file(s).

e) Display: Allow user to plot GPP against E_{mult}, separately, as an option.

f) NOTE: It's conceivable that if this is the first time a calibration is performed (this won't be the case for SMAP L4C), there are no existing lower/upper bounds with which to plot an initial ramp function. In this case, use the theoretical lower/upper bounds prescribed in Table 5.

7) Use constrained non-linear optimization to minimize the sum of squared errors between the observed (ground truth) flux tower GPP and the model simulated GPP.

a) Allow user to choose which parameters to use in optimizing GPP: Any combination of LUE_{max} (1 parameter), VPD (2 parameters: lower and upper bound of ramp function), SMRZ (2 parameters: lower and upper bound of ramp function), TMIN (2 parameters: lower and upper bound of ramp function), and FT (1 parameter, multiplier for frozen conditions). Default to all of these.

b) Numerical optimization is supported in many software libraries. Our calibration software will simply have to prepare a vector of initial parameters (those listed under "Optimization: GPP" in Table 5) and the **GPP Optimization Objective Function**.

c) **After optimization,** report the differences between the old and new parameter values. Also report which parameters (if any) hit the lower or upper bounds. A plot would be helpful (see Figure 6).

d) **After optimization, allow user to run Step 6 (above) again, i.e., allow user to plot the *new* (optimized) ramp functions.**

8) **Pre-optimization for RECO: Prepare subset flux tower data for optimizing model simulation of RECO and display current parameters to user.**

a) Read in, from the current BPLUT table, the current RECO parameter values: $\mathbf{f_{aut}}$, $\beta_{\mathbf{TSOIL}}$, SMSF_{min}, and SMSF_{max}; **use these as the initial values in the optimization.**

b) Allow the user to specify two parameters: $\mathbf{P_{RH}}$ (Percentile of R_H/K_{mult} to use in initializing SOC pools) and $\mathbf{P_K}$ (Percentile to use as minimum threshold for acceptable K_{mult} values). Both P_{RH} and P_K must be between 0 and 1, inclusive.

c) Calculate linear ramp function for SMSF and Arrhenius response curve (see Equation 10) for TSOIL given *current* BPLUT values.

d) **Calculate $\bar{C}$ for each flux tower site:** This is the P_{RH} percentile of R_H/K_{mult}, *after* filtering out K_{mult} values below the P_K percentile. So, for example, if $P_K = 0.5$ (50th percentile or median) and $P_{RH} = 0.9$ (90th percentile), we first filter out the bottom 50% of K_{mult} values, calculate R_H/K_{mult}, then pick the 90th percentile of this ratio. $\bar{C}$ is then assigned this value at every time point for the respective flux tower station.

e) Display: A plot of $R_H/\bar{C}$ against TSOIL with the Arrhenius curve on top; A plot of $R_H/\bar{C}$ against SMSF with the SMSF ramp function on top. Both plots are similar to the example shown in Figure 5.

f) Display: Allow user to plot $R_H/\bar{C}$ against K_{mult}, separately, as an option.

g) NOTE: It's conceivable that if this is the first time a calibration is performed (this won't be the case for SMAP L4C), there are no existing lower/upper bounds with which to plot an initial ramp function. In this case, use the theoretical lower/upper bounds prescribed in Table 5.

9) **Use constrained non-linear optimization to minimize the sum of squared errors between the observed (ground truth) flux tower RECO and the model simulated RECO.**

a) Allow user to choose which parameters to use in optimizing RECO: Any combination of f_{aut}, β_{TSOIL}, SMSF_{min}, and SMSF_{max}. Default to all four (4).

b) Numerical optimization... Our calibration software will simply have to prepare a vector of initial parameters (those listed under "Optimization: RECO" in Table 5) and the **RECO Optimization Objective Function**.

c) **After optimization,** $\bar{C}$ will have to be calculated once more for each flux tower site using the optimized RECO parameters (see Step 8d, above). This provides the final, "best" estimate of $\bar{C}$.

d) **After optimization,** report the differences between the old and new parameter values. Also report which parameters (if any) hit the lower or upper bounds. A plot would be helpful (see Figure 6).

e) **After optimization, allow user to run Step 8 (above) again, i.e., allow user to plot the *new* (optimized) ramp functions.**

10) Fit soil organic carbon (SOC) storage and decay parameters.

a) Read in from the last-used (current) BPLUT the SOC parameter values: $\mathbf{f_{met}}$, $\mathbf{f_{str}}$, $\mathbf{R_{opt}}$, $\mathbf{k_{str}}$, $\mathbf{k_{rec}}$; **use these as the initial values in the optimization.**

b) **Calculate GPP*, $\mathbf{K^*_{mult}}$, and NPP* at 1-km scale based on a 365-day average climatology according to the Analytical Model Spin-Up.** *Note that not all of the Spin-Up steps are needed at this time; litterfall and stead-state SOC pool sizes do not need to be calculated.* *Denotes that these variables are calculated based on a 365-day average annual climatology and are not the full, inter-annual time-series record.

c) For each flux tower site, calculate:

$$\sigma = \frac{1}{M} \frac{\sum_{t=1}^{365} NPP^*(t)}{\sum_{t=1}^{365} K^*_{mult}(t)}$$

Where NPP^* and K^*_{mult} are the $[365 \times N_s \times 81]$ matrices of NPP and K_{mult}, *filtered* to the M 1-km pixels (out of 81) that match the chosen PFT. M is then the number of "valid" 1-km pixels that remain after filtering out those that don't match the chosen PFT.

d) Calculate β_{SOC}:

$$\beta_{SOC} = s \times \left(f_{met} + \frac{1 - f_{met}}{k_{str}} + \frac{f_{str}(1 - f_{met})}{k_{rec}} \right) R_{opt}^{-1}$$

Where s is a scaling parameter. The default should be $s = 0.001$, which will result in SOC values in kg C m^{-2}.

e) Plot $(\sigma \times \beta_{SOC})$ against the ground-truth flux tower SOC sizes; show the 1:1 line on this plot and also the value of Pearson's correlation coefficient. See Figure 7 for an example.

11) Collect all optimized model parameters. Run Analytical and Numerical Model Spin-Up procedures for a climatological year.

a) Allow the user to specify a number of Numerical Spin-Up iterations to run.

b) **Run the Analytical Model Spin-Up** (Section 4.6). The analytical model spin-up calls the **L4C Forward Model Run** (Section 4.8) to generate an initial L4C climatology.

c) **Numerical Model Spin-Up** (Section 4.7). The numerical model spin-up is used to further nudge the SOC stock sizes to their most accurate starting values. It also calls the **L4C Forward Model Run** (Section 4.8) but runs it for several iterations, as specified by the user in step (a), above.

12) **Spin-up and run forward over the full time series** (approx. 2000-2019) for the subset flux tower sites.

a) Run the **Preliminary Spin-Up: Arbitrary Period** (Section 4.5.1) over the full operational record (January 1, 2000 to "present"); obtain GPP, K_{mult}, and average daily litterfall.

b) Run the **L4C Soil Model Forward Run** (Section 4.8) over the full operational record (where T is approximately equal to 365×19 years).

13) **Compute comprehensive validation and fit statistics for each site.**

(a) *In calculating each statistic, below, make sure that a harmonized missing data screen is used.* Remove any case where data is missing in at least one of the datasets: site-observed values, the **L4C Reference dataset**, the modeled values based on optimized parameters, or the **L4C Forward Model Run.**

(b) For each flux tower site, Calculate RMSE (see Section 4.9), the unbiased RMSE (see Section 4.10), and Pearson's r (correlation coefficient) for between site-observed GPP ("ground truth") and: a) GPP in the **L4C Reference dataset**; b) GPP that was fit in the GPP optimization; and c) GPP from the **L4C Forward Model Run.** This results in 3 sets of statistics for GPP.

(c) For each flux tower site, Calculate RMSE (see Section 4.9), the unbiased RMSE (see Section 4.10), and Pearson's r (correlation coefficient) for between site-observed RECO and: a) RECO in the **L4C Reference dataset**; b) RECO that was fit in the RECO optimization; and c) RECO from the **L4C Forward Model Run.** This results in 3 sets of statistics for RECO.

(d) Compute NEE as $NEE = RECO - GPP$, based on RECO and GPP calculated using the optimized RECO and GPP parameters from steps **7**) and **9**), respectively.

(e) For each flux tower site, Calculate RMSE (see Section 4.9), the unbiased RMSE (see Section 4.10), and Pearson's r (correlation coefficient) for between site-observed NEE and: a) NEE in the **L4C Reference dataset**; b) NEE that was fit in the NEE optimization; and c) NEE from the **L4C Forward Model Run.** This results in 3 sets of statistics for NEE.

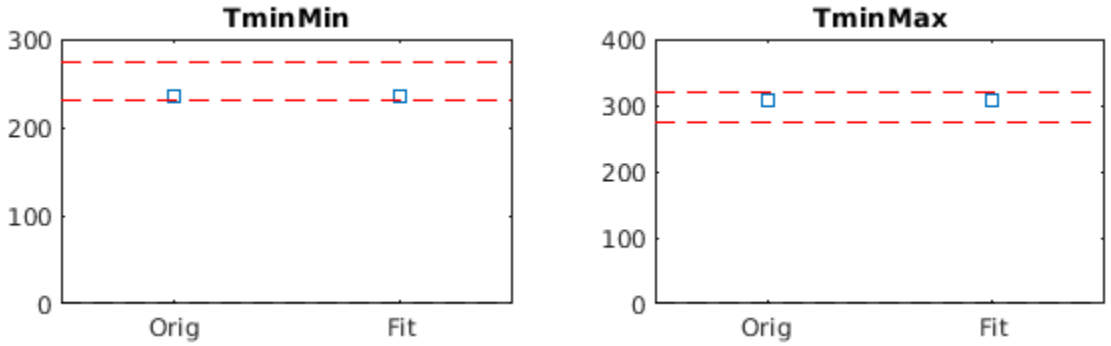

Figure 6: Example of plot comparing two of the GPP calculation's parameters (shown as blue squares); "Orig" refers to the original (old) parameter value and "Fit" refers to the optimized value (better names would be "Original" and "Updated Fit"). In this example, the old and new values are the same because nothing changed in the calibration data. Lower and upper bounds are shown as dashed red lines.

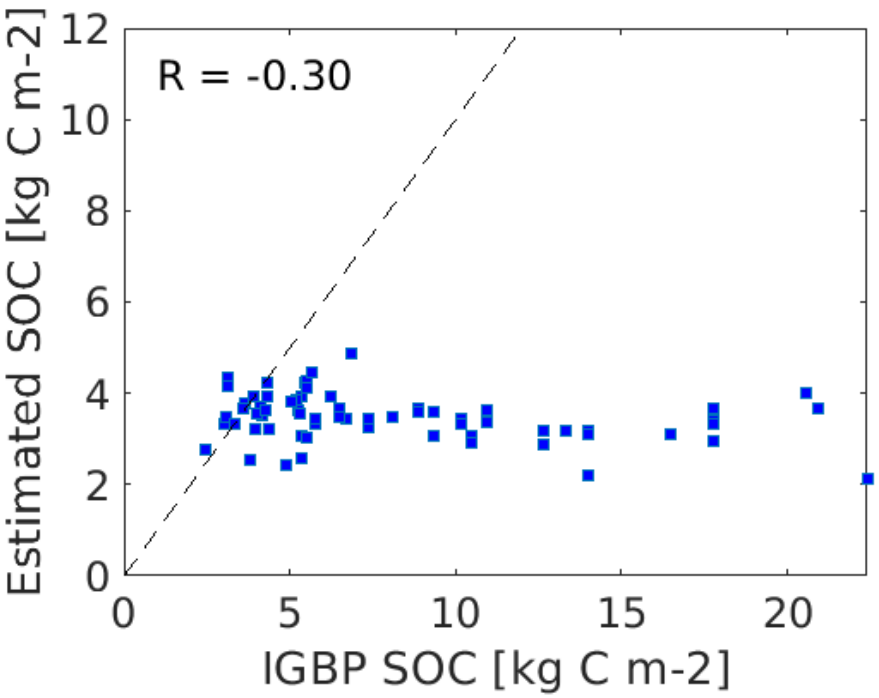

Figure 7: Example of plot comparing the flux tower SOC ("IGBP SOC [kg C m-2]") against the model-estimated SOC (on the vertical axis). The 1:1 line is shown and Pearson's correlation coefficient is also displayed. Each blue square is the inventory/ ground-truth SOC stock size at a single flux tower site (for the chosen PFT).

4 Business Logic Details

4.1 Input Datasets for Calibration

There are four (4) datasets that need to be read in at the beginning of the calibration procedure; these three datasets have daily observations for each flux tower location:

- **L4C Reference dataset**; it is acquired from NASA and needs to be spatially subset to the flux tower locations.
- **L4C Meteorological Input dataset**
- **Flux Tower Fluxes dataset**
- **IGBP FAO Soil Organic Carbon (SOC) Inventory dataset**

All four datasets will be provided by NTSG in HDF5 format.

The **L4C Reference Dataset** is typically the last version of the L4C Nature Run. The **L4C Meteorological Input dataset** will be formatted such that FPAR is re-scaled from a percentage to a proportion; bad FPAR values are masked out, then filled-in with an ancillary FPAR climatology; and the 8-day average FPAR values are interpolated to daily FPAR.

The **Flux Tower Fluxes dataset** will include a list of the flux tower sites, their latitude-longitude coordinates, and **the station weights**.

4.2 Configuration File

The **Configuration File** should define:

- The current or **Reference BPLUT Table**, which was produced from the last calibration run;
- A list of flux tower sites;
- A list of flux tower sites *to exclude from calibration*;
- The file path to the last-used **Nature Run dataset**;
- All file paths for input HDF5 files;
- All file paths for the output files;

4.3 GPP Optimization Objective Function

The **GPP Optimization Objective Function** has two main components:

1. Given the input parameters ("Optimization: GPP" in Table 5), calculates GPP according to Equations 4, 5, and 6, reproduced below.

2. Calculate the sum of squared errors (SSE) between the calculated/ simulated GPP and the flux tower observed GPP. The SSE is the value given to the optimizer.

These steps are repeated multiple times, driven by the optimizer, until there is only a very small difference in SSE between iterations. In Step (1), above. we calculate GPP according to the formula (reproduced from Equations 4, 5, and 6):

$$GPP = (FPAR \times PAR) \times \varepsilon_{max} \times E_{mult}$$
$$E_{mult} = f(VPD)\, f(TMIN)\, f(SMRZ)\, FT_{mult}$$

The $f(*)$ are the linear ramp functions between the minimum (x_{min}) and maximum (x_{max}) values of each environmental constraint (VPD, TMIN, or SMRZ), such that $f(*)$ takes on a value between 0 and 1 (see Equation 7). FT_{mult} is not a ramp function because freeze-thaw conditions are binary; FT_{mult} is simply the multiplier value used when conditions are frozen (the multiplier is 1.0 when conditions are thawed).

The optimizer is iteratively changing the parameter values for those parameters in Table 5. At each iteration, SSE is calculate; the optimizer compares the new value of SSE to the last value to determine how to change the parameter values. In this way, the optimizer gets "closer" to the best-fit parameter values. See Section 4.9, **Sum-of-Squared Errors Calculation Used in Objective Functions**, below.

4.4 RECO Optimization Objective Function

The **RECO Optimization Objective Function** has two main components:

1. Given the input parameters ("Optimization: RECO" in Table 5), calculates RECO according to Equations 9 and 11, reproduced below.

2. Calculate the sum of squared errors (SSE) between the calculated/ simulated RECO and the flux tower observed RECO. The SSE is the value given to the optimizer.

These steps are repeated multiple times, driven by the optimizer, until there is only a very small difference in SSE between iterations. In Step (1), above, we are calculating RECO according to the formula below.

$$\begin{aligned} RECO &= R_A + R_H \\ &= (\mathbf{f_{aut}} \times GPP) + R_H \\ &= (\mathbf{f_{aut}} \times GPP) + (K_{mult}\, \bar{C}) \\ &= (\mathbf{f_{aut}} \times GPP) + \left[f(TSOIL)\, f(SMSF)\, \bar{C} \right] \end{aligned}$$

Note that $\bar{C}$ must also be calculated as part of optimizing RECO in each iteration; this is calculated the same way as it was calculated initially in Step 8d (see Section 3.3).

Recall that K_{mult} can also be calculated as follows:

$$K_{mult} = f(TSOIL)\, f(SMSF)$$

$$f(TSOIL) = \exp\left[\beta_{TSOIL}\left(\frac{1}{66.02} - \frac{1}{TSOIL - 227.13}\right)\right]$$

These are reproduced from Equations 9 and 10. Recall that $f(SMSF)$ is a simple linear ramp function as in Equation 7.

4.5 Preliminary Model Spin-Up Calculations

These calculations are used to obtain the plant productivity estimates (GPP, NPP, and the R_H constraint multiplier K_{mult}) which ultimately determine soil organic carbon (SOC) storage and decay (soil respiration). **Basically, these calculations have to be performed prior to any simulation of SOC or heterotrophic respiration.** There are two versions of these calculations: One over a time period of arbitrary length and one over a 365-day climatological year. Otherwise, the procedure is the same (simulate GPP, calculate K_{mult}, calculate NPP, then average daily litterfall).

4.5.1 Preliminary Spin-Up: Arbitrary Period

This is the more general spin-up of model parameters: **Unlike the Climatology Spin-Up, this simulates GPP, K_{mult}, and NPP for any arbitrarily long time series (not just a 365-day climatological year.** Model spin-up (over a climatological year) is invoked as at the beginning of the Analytical Model Spin-Up, the Numerical Model Spin-Up, and the L4C Soil Model Forward Run; model spin-up over the full time-series record is invoked at the beginning of the Full L4C Forward Simulation.

The simulation of GPP, K_{mult}, and NPP are either performed at or propagated to (i.e., values repeated for) the 1-km scale. The length of the time series simulated, T, may vary; for the Full L4C Forward Simulation, it is usually from January 1, 2000 to the latest date available in the operational record (i.e., within about 10-14 days of the start of calibration). GPP and K_{mult} are each a matrix $[T \times N_s \times 81]$ in size where T is the number of days in the full time-series record and N_s is the number of flux tower sites for the given PFT.

1. **Calculate GPP using optimized GPP parameters for each 1-km grid cell at each subset flux tower site.** We optimized the GPP parameters in Step 7; now calculate GPP using the updated parameters (see equation in Section 4.3).

2. **Calculate $\mathbf{K_{mult}}$ using optimized RECO parameters.** We optimized RECO in Step 9; now calculate K_{mult} using the updated parameters (see equations in Section 4.4). **Propagate the K^*_{mult} value for a given site 81 times to each of its 1-km subgrid cells.**

3. **Calculate NPP from the 1-km GPP just calculated.** Recall that NPP is $GPP - R_A$ (see Equation 3). Use the value of f_{aut} that was optimized as part of Step 9 to calculate NPP as: $\mathbf{NPP} = \mathbf{GPP} - (f_{aut} \times \mathbf{GPP})$.

4. **Calculate average daily litterfall.** This is the total NPP over the time series divided by T, to obtain the average daily NPP (which is assumed to be seasonal litterfall).

4.5.2 Preliminary Spin-Up: Climatology Spin-Up

In contrast with the Preliminary Spin-Up: Arbitrary Period, this version spins up over the 365-day climatological year. That is, the only differences between this section and the preceding one is that, here, $T = 365$ and the input time series data for calculating GPP, NPP, and K_{mult} are restricted to a 365-day climatological average year.

This procedure calculates $\mathbf{GPP^*}$, $\mathbf{K^*_{mult}}$, $\mathbf{NPP^*}$, and litterfall over a 365-day climatological year at 1-km. Note that, below, the asterisk (*) simply denotes quantities that are calculated over the 365-day climatological year.

1. **Calculate $\mathbf{GPP^*}$ using optimized GPP parameters and *the annual, 365-day climatology* for each 1-km grid cell at each subset flux tower site.** We optimized the GPP parameters in Step 7; now calculate GPP^* using the updated parameters (see equation in Section 4.3) and the 365-day climatology that was calculated in Step 5 of Section 3.3. The result is a matrix $[365 \times N_s \times 81]$ in size where N_s is the number of flux tower sites for the given PFT.

2. **Calculate $\mathbf{K^*_{mult}}$ using optimized RECO parameters and *the annual, 365-day climatology.*** We optimized RECO in Step 9; now calculate K^*_{mult} using the updated parameters (see equations in Section 4.4) and the 365-day climatology that was calculated in Step 5 of Section 3.3. **Propagate the K^*_{mult} value for a given site 81 times to each of its 1-km subgrid cells.** The result is a matrix $[365 \times N_s \times 81]$ in size where N_s is the number of flux tower sites for the given PFT.

3. **Calculate $\mathbf{NPP^*}$ from the 1-km, 365-day GPP^* just calculated.** Recall that NPP is $GPP - R_A$ (see Equation 3). Use the value of f_{aut} that was optimized as part of Step 9 to calculate $\mathbf{NPP^*}$ as: $\mathbf{NPP^*} = \mathbf{GPP^*} - (f_{aut} \times \mathbf{GPP^*})$. The result is a matrix $[365 \times N_s \times 81]$ in size where N_s is the number of flux tower sites for the given PFT.

4. **Calculate average daily litterfall.** This is the total annual NPP divided by 365 (days). Total annual NPP is the sum across NPP for the 365-day climatological year; dividing the sum by 365 produces the daily average.

4.6 Analytical Model Spin-Up

This procedure is used to initialize soil organic carbon (SOC) stock sizes. SOC, in the real world, takes decades to centuries to build up through the gradual accumulation of leaf litter, stems, dead roots, and other organic matter. We use fancy math to make a good estimate

of what these SOC stock sizes should be so as to avoid having to simulate this accumulation over many years (i.e., many model runs). Note that, below, the asterisk (*) simply denotes quantities that are calculated over the 365-day climatological year.

1. **Run the Preliminary Climatology Spin-Up** (Section 4.5.2). At this point, we have calculated all the quantities needed for fitting the SOC storage and decay parameters: $\mathbf{GPP}^*$, $\mathbf{K}^*_{\mathbf{mult}}$, and $\mathbf{NPP}^*$. We proceed, below, only as part of Step 11, when we want to calculate the initial stock sizes prior to the running the forward model.

2. **Calculate $\mathbf{C_{met}}$, $\mathbf{C_{str}}$, and $\mathbf{C_{rec}}$ according to Equations 12, 13, and 14 (respectively), then calculate $\bar{\mathbf{C}}_\mathbf{0}$ according to Equation 15.**

3. **Calculate the sums of NPP^* and K^*_{mult} over the climatological year:** $\sum_{t=1}^{365} NPP^*(t)$ and $\sum_{t=1}^{365} K^*_{mult}(t)$.

The results of this Spin-Up procedure (and the antecedent Preliminary Climatology Spin-Up) are:

- GPP^* at 1-km for a 365-day climatological year (for each subset flux tower site);
- K^*_{mult} at 1-km for a 365-day climatologial year (for each subset flux tower site);
- Average daily Litterfall at 1-km (one value per 1-km subgrid cell per flux tower site, or an $[N_s \times 81]$ matrix);
- The C_{met}, C_{str}, and C_{rec} pools (one value per 1-km subgrid cell per flux tower site, or an $[N_s \times 81]$ matrix for each pool);
- The quantity $\bar{C}_0$ (one value per 1-km subgrid cell per flux tower site, or an $[N_s \times 81]$ matrix);
- Sum of NPP^* over the climatological year: $\sum_{t=1}^{365} NPP^*(t)$;
- Sum of K^*_{mult} over the climatological year: $\sum_{t=1}^{365} K^*_{mult}(t)$;

4.7 Numerical Model Spin-Up

This requires simply running the **L4C Soil Model Forward Run** over and over again (as many times as specified by the user). Below, in pseudocode, `ForwardRun()` refers to the routine described in Section 4.8. Below, `analysis` refers to the `track` of variables from a single `ForwardRun()` iteration.

```
get number_of_iterations from user
set total_iterations as [1, 2, ... number_of_iterations]

for each iteration in total_iterations
  # Run the Forward Run for a climatological year
  get analysis, c1, c2, c3, cbar0 from ForwardRun()
```

```
  set spinup_track = {
    # NOTE: Concatenating latest iteration's cbar0 with all prior
    set cbar = [analysis{cbar0}, spinup_track{cbar0}]
    set c1 = [analysis{c1}, spinup_track{c1}]
    set c2 = [analysis{c2}, spinup_track{c2}]
    set c3 = [analysis{c3}, spinup_track{c3}]
    set tolerance = analysis{tolerance}
  }
endfor
```

4.8 L4C Soil Model Forward Run

This "forward run" simulates soil organic carbon (SOC) build-up and decay (soil respiration). It depends on the following parameters; names in pseudocode is given first:

- `ropt, kstr, krec`: Decay rates/ ratios for each SOC pool: $\mathbf{R_{opt}}, \mathbf{k_{str}}, \mathbf{k_{rec}}$;
- `fmet, fstr`: Fractional inputs of each SOC pool: $\mathbf{f_{met}}, \mathbf{f_{str}}$;
- `faut`: Fraction of productivity used in respiration: $\mathbf{f_{aut}}$;

And requires the following input variables as a time series, for T days (e.g., if needed for a climatological year, $T = 365$); name in pseudocode is given first:

- `gpp1km[]`: **GPP** or **GPP***, which is GPP at 1-km resolution for each flux tower site for each of T days;
- `kmult1km[]`: $\mathbf{K_{mult}}$ or $\mathbf{K^*_{mult}}$ (Which is given: $K_{mult} = f(TSOIL)\, f(SMSF)$), or K_{mult} at 1-km resolution for each flux tower site for each of T days;

And the following additional variables that are static or initialized at time $t = 0$; name in pseudocode is given first:

- `litterfall[]`: **Average daily litterfall** at 1-km resolution for each flux tower site;
- `pft[]`: **PFT of each 1-km subgrid cell** for each flux tower site;
- `cmet0[], cstr0[], crec0[]`: **Initial SOC pool sizes for each flux tower site:** $\mathbf{C_{met}}, \mathbf{C_{str}}, \mathbf{C_{rec}}$;

And the algorithm must calculate the following state variables at each time step from day $t = 1$ to day $t = T$; name in pseudocode is given first:

- `rh1[], rh2[], rh3[]`: Daily R_H from each SOC pool (3 pools total);
- `cbar0[]`: Daily $\bar{C}_0$ (see Equation 15), which will change as the SOC pools change size;
- `dc1[], dc2[], dc3[]`: Daily change in each SOC pool (3 pools total);

- `c1[], c2[], c3[]`: Total size of each SOC pool (3 pools total) on the given day;

The procedure is as follows (enumerated here and then in pseudocode below):

1. For each flux tower site, calculate the number of valid 1-km subgrid cells, matching the chosen PFT, within the flux tower site's 9-km cell *(not shown in pseudocode)*.
2. Set the following state variables to zero for each flux tower site for each 1-km cell (i.e., each is a $[N_s \times 81]$ matrix, initially zero, where N_s is the number of flux tower sites for the given PFT): R_H, $\bar{C}_0$, and the instantaneous (daily) change in each C pool (below, given as: `dc1, dc2, dc3`.
3. Set the C pool state variables, each an $[N_s \times 81]$ matrix, to their respective initial pool sizes, i.e., `c1, c2, c3` are set to, respectively, C_{met}, C_{str}, and C_{rec}.
4. For each C pool, calculate RH according to Equation 8 using the 1-km, 365-day K^*_{mult}.
5. Calculate $\bar{C}_0$ according to Equation 15 (this is a sum across all three pools).
6. Calculate change in each pool, based on average daily litterfall, according to...

```
# NOTES
# * denotes element-wise multiplication, not matrix multiplication

# Initialize state matrices that are [Ns x 81] in size with zeros
set rh1[], rh2[], rh3[] = 0 # Instantaneous RH, by pool
set dc1[], dc2[], dc3[] = 0 # Instantaneous change in C, by pool

# Set the initial pool sizes; these are [Ns x 81] matrices
set c1[] = cmet0[]
set c2[] = cstr0[]
set c3[] = crec0[]

for each day in T days
  set cbar0 = 0 # Reset cbar0

  # kmult1km is a [T x Ns x 81] matrix
  set rh1 = (ropt * kmult1km[day] * c1)
  set rh2 = (ropt * kstr * kmult1km[day] * c2)
  set rh3 = (ropt * krec * kmult1km[day] * c3)

  # Accumulate cbar0
  set cbar0 = cbar0 + (ropt * c1) + (ropt * kstr * c2) + (ropt * krec * c3)

  # Calculate change in each pool (litterfall is [Ns x 81] matrix)
  set dc1 = (litterfall * fmet) - rh1
  set dc2 = (litterfall * (1 - fmet)) - rh2
```

```
  set dc3 = (fstr * rh2) - rh3

  # Slight adjustment for "humification"
  set rh2 = rh2 * (1 - fstr)

  # Accumulate C pools
  set c1 = c1 + dc1
  set c2 = c2 + dc2
  set c3 = c3 + dc3

  # Calculate total change across all 3 pools, explicitly by site, 1-km cell
  for each site
    for each 1km cell
      # So dc_total is a [Ns x 81] matrix
      set dc_total[site, cell] = dc1[site, cell]
        + dc2[site, cell]
        + dc3[site, cell]
    endfor
  endfor

  # And we keep track of this fine-scale change as model moves forward
  set tolerance = tolerance + dc_total

  # IMPORTANT: Keep track of state on each day
  set track = {
    set rh_total[day] = rh1 + rh2 + rh3
    set c_total[day] = c1 + c2 + c3
    set ra[day] = faut * gpp1km[day] # Autotroph. resp. on this day
    set reco[day] = ra[day] + rh_total[day] # RECO or sum of RA and RH
    set nee[day] = reco[day] - gpp1km[day] # NEE
    set tolerance[day] = tolerance
  }
endfor
```

Outputs from the L4C Soil Model Forward Run are:

- All tracked variables for each day in T days (i.e., the `track` associative array in pseudocode, above);
- The final C stock size in each pool, which is `c1, c2, c3`;
- The final state of $\bar{C}_0$, which is `cbar0`;

By "final state," it is meant the state of the variable on day T (e.g., for a climatological year, this would be day 365).

4.9 Sum-of-Squared Errors Calculation Used in Objective Functions

The sum-of-squared errors (SSE) is an error metric; larger SSE values mean a worse fit/ correspondence between observed or "ground-truth" values and modeled values. The equations below provide a general formula for calculating SSE, where Obs refers to the observed values and Pred refers to the predicted values.

$$RMSE(s) = \frac{1}{N_s - 1}\sqrt{\sum_{i=1}^{N_s} \left(\text{Obs}(s,i) - \text{Pred}(s,i)\right)^2} \tag{16}$$

$$SSE = 100 \sum_{s=1}^{S} (N_s)^{-1} \sum_{s=1}^{S} N_s \, w_s \; RMSE(s) \tag{17}$$

N_s is the number of non-missing observations for flux tower station s and w_s is the weight assigned that flux tower station (this refers to the flux tower weights defined by the user and read-in from the **Flux Tower Fluxes dataset**). The sole reason to scale SSE by 100, above, is to avoid numerical issues in the optimization. Matlab code for the SEE calculation is provided below, as an example; in this example, flux-tower GPP (ground truth) is compared against modeled GPP.

```
function [rmse, ns] = rmsexy(x,y)
[n,m] = size(y);

% Too-fancy way of calculating which cases are missing in both x and y
msng = isnan(x.*sum(y,2));

% Calculate how many non-missing cases there are
ns = sum(~msng);

% Pre-allocate output vector
rmse = NaN.*ones(1,1);

rmse(j) = sqrt(sum((x(~msng)-y(~msng)).^2)./(ns-1));
end

% For each station, calculate RMSE
for s = 1:Ns
[sses(s),n(s)] = rmsexy(fluxTower.gpp(:,s), modelPrediction.gpp(:,s));
end

% Calculate SSE
sse = 100.*sum(sses.*siteWeights.*n/sum(n),'omitnan');
```

4.10 Unbiased RMSE Calculation

The "unbiased" RMSE calculate is similar to the RMSE calculation (see Section 4.9), but includes detrending the residual differences.

$$uRMSE(s) = \frac{1}{N_s - 1}\sqrt{\sum_{i=1}^{N_s} \left[D\left(\text{Obs}(s,i) - \text{Pred}(s,i)\right)\right]^2} \tag{18}$$

Above, $D(Y) = D(\text{Obs}(s,i) - \text{Pred}(s,i))$ is the detrending function. $D(Y)$ is equal to ε in the linear regression:

$$Y = X\beta + \varepsilon \quad \text{where} \quad X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & \vdots \\ 1 & N \end{bmatrix}$$

Where N is the length of X (as well as the length of Y).

Plainly, the unbiased RMSE calculation is identical to RMSE except that the residual differences between observations and predictions are first detrended: the linear trend in the residual differences is removed through simple linear regression. An R script, below, demonstrates the effect on synthetic data. The increasing trend (from left to right) in the original signal is removed.

```
sig <- rep(c(0, 1, -2, 1, 0, 1, -2, 1, 0), 2)
trend <- ((1:18)/18)^3

# Signal corrupted by a (cubic) linear trend
y <- sig + trend

x <- 1:18
y.detrended <- residuals(lm(y ~ x))

# Plot original signal in black dashed, detrended signal in red
plot(y, type = 'l', bty = 'n', lwd = 2, lty = 'dashed')
lines(y.detrended, col = 'red', lwd = 2)
```

Table 5: Lower and upper bounds on the optimization of parameters in the BPLUT table. The specific L4C output variable that is determined by these input parameters (e.g., GPP) is listed in the first column.

Optimization	**BPLUT Variable**	**Lower Bound**	**Upper Bound**
GPP	**LUE** or ε_{max}	0.5	4.0
GPP	$\mathbf{VPD}_{min}$	0	1500
GPP	$\mathbf{VPD}_{max}$	1500	7000
GPP	$\mathbf{SMRZ}_{min}$	-30	30
GPP	$\mathbf{SMRZ}_{max}$	31	100
GPP	$\mathbf{TMIN}_{min}$	230	274
GPP	$\mathbf{TMIN}_{max}$	275	320
GPP	$\mathbf{FT}_{mult}$	0.0	1.0
RECO	$\mathbf{f_{aut}}$	0.0	0.7
RECO	$\beta_{\mathbf{TSOIL}}$	1	800
RECO	$\mathbf{SMSF}_{min}$	-50	10
RECO	$\mathbf{SMSF}_{max}$	10	100

5 Supporting Documentation

5.1 Additional Details

5.1.1 Bounds on Optimized Values

5.2 External Documents

SMAP L4C Global Daily Carbon Data Products - User Guide at NSIDC

- `https://nsidc.org/data/SPL4CMDL/versions/4`
- This is a guide for end-users of the L4C data products.