Geosci. Model Dev. Discuss.,
doi:10.5194/gmd-2016-49-RC2, 2016

Geoscientific Model Development Discussions
Open Access
EGU

Interactive comment on "Simple Process-Led Algorithms for Simulating Habitats (SPLASH v.1.0): Robust Indices of Radiation, Evapotranspiration and Plant-Available Moisture" *by* Tyler W. Davis et al.

Anonymous Referee #2

Received and published: 25 May 2016

This manuscript presents the new SPLASH model, which builds on the widely-used STASH model for calculating bioclimatic variables. The authors describe improvements to the model algorithms that resolve some known issues with STASH. In addition, the authors have also made available the model code in C++, FORTRAN, Python, and R versions. Both the manuscript and model code will be very useful to the climate and ecological modelling communities. Below I have listed some general comments followed by more specific comments. For this review, I ran the FORTRAN version of SPLASH using the San Francisco test data described in the manuscript. I also ran global 10-minute simulations using 30-year mean pseudo-daily climate data. The FOR-

TRAN code is well documented and was easy to compile (I used an INTEL FORTRAN compiler) and run.

As currently written, the manuscript presents model results for one grid point near San Francisco, California (USA). I would suggest expanding the manuscript to include figures displaying model output for the globe (these could be added to the text or placed in a supplement) as well as some evaluation of the results. As noted in the text, this model can (and will) be applied to spatial grids so it is important to provide evidence that the model works across the range of global climate conditions. Evaluating the model output using observed data is also important. SPLASH uses a set of simplified equations and so the model output will not necessarily closely match observed data. Instead the importance of the evaluation is that it will provide some indication of the extent to which SPLASH may over- or under-predict certain variables and whether there are any spatial biases in the model performance.

As you note (page 2, lines 31-34), SPLASH has been designed so that it can be used for palaeoclimate applications. The code is currently set up so that the orbital parameters used for palaeoclimate simulations are not input as variables but are specified in the code as parameters. I would suggest adding text to the manuscript (and a comment in the code) that describes for the user exactly what variables need to be modified when running the model for palaeo time periods (e.g., obliquity, eccentricity, longitude of perihelion). It would also be useful to describe for the reader any modifications that are required to the input data when doing palaeoclimate simulations. For example, for a palaeoclimate simulation, how should a user specify the input year so that it works with the Julian day subroutine (i.e., get_julian_day)? If a user was running a simulation for 4 ka (∼2000 BCE) would they need to specify the input year as -2000 to get the correct Julian day? Does the Meeus (1991) Julian day algorithm used in the model work for negative numbers?

Page 1, line 11: You say that the climatic drivers for the model include "either fraction of bright sunshine hours or fractional cloud cover," which makes the two variables sound interchangeable as input. The equations in the code are set up to use sunshine data as input, not cloud cover. I would suggest deleting the reference in this sentence to fractional cloud cover to prevent a user from misinterpreting this sentence as indicating that cloud cover data can be used directly without having to first convert them to sunshine data.

Page 3, lines 26-27, Equation 1: As described in Section 2.6, runoff is subtracted from the water balance in calculating daily soil moisture so I would suggest including runoff as a term in Equation 1 (i.e., subtracting RO in the right hand side of the equation).

Page 8, lines 24-30: You list some objections to using observed soil properties to estimate maximum soil moisture capacity, including that doing so does not change the seasonal course of soil moisture. I do not think there is any problem with specifying a constant size (in this case, 150 mm) for the soil moisture "bucket" as for some research questions it could be important to hold bucket size constant. However, the seasonal pattern of soil moisture does change depending on the bucket size. I ran SPLASH for the San Francisco grid point using a 100 mm bucket and the soil moisture reached saturation sooner in the year and also was depleted earlier in the spring. Similarly, using a 200 mm bucket, the soil moisture reached saturation later in the year and more soil moisture was available later in the growing season. These seasonal shifts in soil moisture availability may be significant because, for example, they can affect calculations of annual water deficit. I would suggest revising the text stating that the course of seasonal soil moisture is insensitive to the bucket size, as well as the text indicating that soil type-dependent values for the bucket would not improve the model accuracy.

Page 13, line 13: Add the name of the specific WATCH data set you used.

Page 17, line 10: I would add text describing the typographical error (e.g., Eq. 7 of Gallego-Sala et al. (2010) used 273.3 instead of 237.3).

Code: The SPLASH code is currently available for download from an online Git repos-

Printer-friendly version

Discussion paper

itory. I would suggest providing the code in a supplemental file accompanying the manuscript so that the code is documented in case the Git repository is unavailable in the future. The user could still be directed to the online Git repository for the most current version of the code.

The following comments refer to the FORTRAN version of the code, although the same issues may be present in the C++, R, and Python versions.

1. There are a number of debugging comments in the code I downloaded from the Git repository, such as "consistency check – XXX PROBLEM: THIS LEADS TO DIFFERENCE WITH OTHER VERSIONS XXX." To prevent confusion for the user, remove these comments if the issues have been resolved. If the issues have not been resolved, provide enough detail in the code comment so that the user can determine how the issue may affect their results.

2. In various places in the code the user is referred to particular equations in the documentation file (splash_doc.pdf) that accompanies the code. However, in some cases the equation referenced in the code does not match the equation in the documentation. For example, the calculation of daily photosynthetic photon flux density (PPFD) in the FORTRAN code refers the reader to Equation 57 in the documentation file, which is an equation for the bulk modulus of water. Check that the references in the code to the splash_doc.pdf file are correct.

3. It would help the user if all of the variable names were defined in the code. For example, in the code where PPFD is defined as a type real variable the accompanying comment defines PPFD as "daily PPFD (mol/mˆ2)" instead of "daily photosynthetic photon flux density (mol/mˆ2)."

Figure 3: Change "CRU TS" to "CRU TS3.21" in the caption text.

Interactive comment on Geosci. Model Dev. Discuss., doi:10.5194/gmd-2016-49, 2016.