RadLib: a radiative heat transfer model library for CFD

Victoria B. Stephens, Sally Jensen, David O. Lignell*

Department of Chemical Engineering, Brigham Young University, Provo, UT 84602, United States

Abstract

Ca. 100 words

Keywords: radiative heat transfer, reacting flows, CFD

Required Metadata

Current code version

Ancillary data table required for subversion of the codebase. Kindly replace examples in right column with the correct information about your current code, and leave the left column as it is.

Nr.	Code metadata description	Please fill in this column
C1	Current code version	TODO 2.1
C2	Permanent link to code/repository used for this code version	*github.com/BYUignite/RadLib*
C3	Code Ocean compute capsule	N/A
C4	Legal Code License	MIT license (MIT)
C5	Code versioning system used	Git
C6	Software code languages, tools, and services used	C++, Python 3
C7	Compilation requirements, operating environments & dependencies	TODO
C8	If available Link to developer documentation/manual	TODO
C9	Support email for questions	davidlignell@byu.edu

Table 1: Code metadata (mandatory)

*Corresponding author.

Email address: `davidlignell@byu.edu` (David O. Lignell)

1. Motivation and significance

Why did we make this? Radiation is hard to deal with in CFD. It's complicated. You can't really get analytic solutions to practical problems. In lots of cases, that's fine because radiation can be safely neglected; it's not usually the dominant mode of heat transfer. However, we do combustion CFD. Radiation isn't always important to combustion problems, but when it is, simulations can get grossly inaccurate. So we need a nice way to account for radiation in our CFD simulations.

When is radiation important to combustion simulations? In high-temperature systems, radiation can often be neglected in the beginning and middle of the simulation (assuming, say, a jet flame or a counterflow flame or something like that). At those times (which I'm going to call "early flame"), convective heat transfer dominates pretty heavily. However, radiation becomes more important for late-stage flame phenomena, either after some time has gone by (radiation time scales?), or physically high up in a flame where convection doesn't dominate as much as below, or just in areas that are relatively far away from the flame sheet itself where the reactions are happening. There's also soot, which involves radiation heavily, but soot particles are so big and slow that their time and length scales (and their radiative time scales) mean that radiation doesn't become important until late in the flame evolution.

So what's the problem? Why hasn't this been done already? Radiation is complicated. So is combustion. Combustion in particular can be so complex that direct simulations are too computationally expensive for us to simulate configurations that are practical for engineering systems. Direct simulations are usually used as a research tool. Other modeling approaches exist (LES, ODT, etc.) that lower the computational cost and allow us to simulate practical things. So far so good.

Radiation is a little like combustion in that its core mechanisms are complex, physically and mathematically. In addition, it's directional AND depends on the wavelength of the energy involved (other heat transfer doesn't have the wavelength dependence); its governing equations integrate over direction and wavelength, which makes things extra complicated. RADIATION GOVERNING EQUATIONS HERE. Convective heat transfer follows easy rules, but radiation doesn't. There are simple systems in which we can boil things down to analytic solutions, but most of the time, the simple equations don't apply, and that's often true in combustion systems where there are so many different length and time scales involved. The fundamental equations of radiative heat transfer are big and mathematically complex; they don't have analytic solutions except in the simplest geometries. So what do we do? One of two things. One, we do ray tracing (sometimes referred to as Monte Carlo simulations. These can be extremely accurate, but super computationally expensive. Essentially the equivalent of a direct numerical solution. Two, we simplify the the equations. We make assumptions. And so on. This is potentially less accurate, but more practical. By simplifying the fundamental governing equations, we create models that apply to various situations, systems, and geometries (i.e. black body assumptions, directional assumptions, etc.). There are models developed for CFD and specific combustion systems. These are the models that we've put into practice here in such a way that they're easy to apply to various systems. Modular organization for this purpose.

So far there isn't an easy way to access and use radiation models. There are simple ones in Cantera (check this, there might not be any), but they make too many assumptions or don't work well for combustion or what have you. RadLib is a radiation library of models that you can apply to any simulation type, and we'd like to add it to Cantera, too. Basically, there isn't a reliable way to do radiation calculations in CFD simulations without coding the models yourself, which is extra hard because they're complicated and hard. So we've done it for you and put them in a library that easy for anyone to use.

2. Model descriptions

RadLib includes three models of varying complexity and accuracy to calculate the radiation absorption coefficients and their weighting factors for each gas considered by the model. Radiation absorption coefficients are typically calculated using correlations relating them local properties such as temperature, total pressure, or species partial pressure, depending on the model. Correlations come from curve fits to high-resolution radiation property databases. At present, RadLib considers up to four gas species (H_2O, CO, CO_2, and CH_4) and, optionally, soot volume fraction in its calculation of absorption coefficients and weighting factors.

Once the radiative absorption coefficients are calculated, they are then used to solve the radiative transfer equation (RTE), which depends on the simulation configuration and assumptions. Solving the RTE is not the focus of this software, but RadLib's example cases do employ a simple implicit trapezoid method solver to calculate the radiative heat flux and volumetric heat source profiles between two parallel planes. [MORE DESCRIPTION OF SOLVER GOES HERE?]

[OTHER THINGS THAT APPLY TO ALL MODELS GO HERE]

[MAYBE INCLUDE SOME BASIC INFO ON LBL?]

2.1. Planck Mean absorption coefficients

When we refer to the Plank Mean model, what we're using is actually the Planck Mean absorption coefficients, calculated from the correlations given on the TNF workshop site [1]. Their correlations (temperature dependent) are based on the RADCAL model in [2]. The TNF radiation model is also documented in [3].

This model is commonly used because it's relatively simple, easy to implement, low in computational expense, and provides reasonably accurate results in many cases.

Quote from TNF site page: "The characteristics of some flames selected for the workshop are such that a model based upon the assumption of optically thin radiative heat loss should yield reasonable accuracy. This has been demonstrated for the simple hydrogen jet flames [4]. However, there is some evidence that the optically thin model significantly over predicts radiative losses from the CH4 flames in the TNF library, due to strong absorption by the 4.3-micron band of CO2 [5, 6, 7]." Basically, it works well for flames that are actually optically thin, but not so well for flames that aren't. This is especially true of sooting flames.

2.2. Weighted sum of gray gases (WSGG)

The basic assumption of weighted sum of gray gases (WSGG) models in general is that the non-gray behavior of gas mixtures, in this case H_2O and CO_2, can be modeled by a weighted sum of several gray gases and one transparent gas (which represents the spectral windows between absorption bands). RadLib uses the WSGG model presented by Bordbar et al. [8, 9], which uses correlations based on the HITEMP 2010 database [10]. RadLib, in accordance with the Bordbar et al. WSGG method cited above, uses a mixture of four gray gases and one transparent gas. Absorption coefficients are calculated by

$$K_i = \sum_{k=0}^{4} d_{i,k} M_r^k, \tag{1}$$

where K_i is the absorption coefficient for species i, $d_{i,k}$ is a species-specific correlated model coefficient, and M_r is the molar ratio $Y_{\mathrm{H_2O}}/Y_{\mathrm{CO_2}}$. The weight factors are calculated by

$$a_i = \sum_{j=0}^{4} b_{i,j} T_r^j, \tag{2}$$

where a_i is the weighting factor for species i and T_r is a normalized temperature equal to T/T_{ref} with T_{ref} = 1200K. The value of $b_{i,j}$ is calculated by

$$b_{i,j} = \sum_{k=0}^{4} C_{i,j} M_r^k, \tag{3}$$

where $C_{i,j}$ is another correlated model coefficient and M_r is the molar ratio $Y_{\mathrm{H_2O}}/Y_{\mathrm{CO_2}}$ as above. The model coefficients $d_{i,k}$ and $C_{i,j}$ can be found in the literature [8, 9]. RadLib uses the updated model coefficients from [9], which extends the model presented in [8] to include all possible values of the H_2O-CO_2 molar ratio.

2.3. Rank Correlated SLW (RCSLW)

The Spectral Line Weighted-sum-of-gray gases (SLW) model represents a family of global approaches to radiative heat transfer in high-temperature gases that also includes Absorption Distribution Function (ADF) and Full Spectrum k-distribution (FSK) models, all of which are based on the same fundamental principle in modeling the gas absorption spectrum [11]. In order to extend their spectral models from uniform conditions (isothermal, homogeneous gases) to nonuniform conditions (non-isothermal, non-homogeneous gases), these models take a reference approach in which local gas states are corrected relative to a reference state. Reference approaches, however, generally lack consistent reference states and can yield significant errors in cases with large spatial temperature gradients [12]. The Rank Correlated SLW (RCSLW) model is a unique extension of the generalized SLW model that does not require a specified gas reference state and preserves the emission term of the spectrally integrated RTE. Recent comparison of advanced SLW modeling approaches revealed that "the Rank Correlated SLW model is the most robust of all models, and demonstrates that it can achieve accurate solutions with as few as 3–5 gray gases" [13]. A brief overview of the general SLW and RCSLW models will be given here; detailed discussion can be found in the literature [14, 15, 16, 17, 18, 11, 12, 19].

The radiative transfer equation (RTE) for an absorbing, emitting, and non-scattering medium along a given path length s in a direction $\hat{\Omega}$ is given by

$$\frac{dI_\eta}{ds} = -\kappa_\eta I_\eta + \kappa_\eta I_{b\eta}, \tag{4}$$

where I_η is the radiative spectral intensity, $I_{b\eta}$ is the Planck spectral distribution of blackbody intensity, and κ_η is the spectral absorption coefficient of the medium. Integrating with respect to wavenumber and subdividing the absorption cross-section into gray gases gives the SLW form of the RTE:

$$\frac{dI_j}{ds} = -\kappa_j I_j + a_j \kappa_j I_b, \qquad j = 0, 1, ..., n. \tag{5}$$

Here, n is the number of gray gases in the model and I_j is the intensity of gray gas j. κ_j is the gray gas absorption coefficient, which can be calculated as $\kappa_j = N\sqrt{C_{j-1}C_j}$, where C_{j-1} and C_j are supplemental absorption cross-sections used to discretize the absorption spectrum [15]. The gray gas weights corresponding to each absorption coefficient are given by $a_j = F(C_j) - F(C_{j-1})$, where $F(C_j)$ is the absorption line blackbody distribution function (ALBDF) for species j, calculated from the detailed absorption spectrum of that species. The absorption cross-sections are chosen with respect to a thermodynamic reference state, which makes generalized SLW methods reference approaches. Once Equation 5 is solved for each gray gas species, to total radiative intensity I can be calculated by summing the gray gas intensities:

$$I = \int_{\eta=0}^{\infty} I_\eta d\eta = \sum_{j=1}^{n} I_j. \tag{6}$$

The Rank-Correlated SLW (RCSLW) model avoids specifying a reference state by relating thermodynamic states to one another instead. If two arbitrary absorption cross-sections define identical wavenumber intervals, the two cross-sections (neither of which is designated a reference cross-section) can be said to be rank-correlated. The inverse ALBDF, defined such that $C[F(C,\phi_g,T_b),\phi_g,T_b] = C$ and $F[C(F,\phi_g,T_b),\phi_g,T_b] = F$ for a given gas thermodynamic state ϕ_g and blackbody temperature T_b, is proven to have this property and can also be called the rank-correlated reordered absorption cross-sections [12]. As a result, the inverse ALBDF can be used to construct cross-section intervals for the RCSLW model that do not rely on a defined reference state or spectrum. RadLib's RCSLW model uses Method 1.2.2 as defined and recommended in [12]

3. Software Description

RadLib is an object-oriented C++ class library, including both C++ and Python interfaces. The RadLib package contains five subdirectories (upon initial download): `source` contains the RadLib source code; `build` contains installation files; `examples` contains instructive example cases, including a simple interface and solver for a parallel planes geometry; `data` contains ALBDF data tables required for the RCSLW model; and `docs` contains files used to generate code documentation with Doxygen (optional).

The `source` and `examples` directories are further divided into `C++` and `Python` subdirectories to differentiate between interfaces. There are three interface options for using this code: C++, Python, and Cython-wrapped Python. C++ interfaces are located within the `examples/c++` folder, while

Python interfaces are located within the `examples/python` folder. When running examples with Python, the Cython-wrapped version is the default; to run the regular Python version without the Cython wrapper, edit the comments near the top of the example files (i.e `ex_S1.py`). The C++ interface produces the fastest-running code, followed by the Cython-wrapped Python interface and then the regular Python interface. [INSERT RUNTIME COMPARISONS HERE]

RadLib installation is automated with CMake. First, navigate to the `radlib/build` directory. If the user requires an installation location other than the default `radlib/installed` directory, edit the `user_config` file before running CMake. To compile the package, run the command `cmake -C user_config ../source`. Upon successful completion, run `make` and then `make install` to complete the process. The generated C++ library file is located at `radlib/installled/lib/libradlib.a`.

Figure 1 illustrates the basic structure use of the RadLib package within a generic example using the provided interfaces. The RadLib library generates absorption coefficients and their weighting factors for use within the appropriate RTE, but does not specify any particular geometry, making it a versatile tool for any simulation that requires radiative heat transfer, regardless of configuration. The interfaces, solver, and examples included with the library serve to illustrate its use and validate the implementation and results.

4. Illustrative Examples

Simple comparisons of each of these models to some ray-tracing results (equivalent to a direct numerical solution) to establish validity. Illustrates when to use various models and why you might choose one over another. Plots and discussion go here. Five example cases demonstrated with this code in both C++ and Python. Compared to line-by-line calculations (essentially the radiation version of Monte Carlo).

5. Impact

As far as I'm aware, there isn't an easy way to incorporate radiative heat transfer into a CFD simulation. Most cases use optically thin assumption (which works in some cases, but not in others) or neglect radiation entirely (also applicable sometimes, but not always). [Refer to TNF website radiation page here.] Unfortunately, this means that when you simulation some configuration in which radiation might be important to the overall heat transfer, you won't get accurate simulation results. Furthermore, if your study is

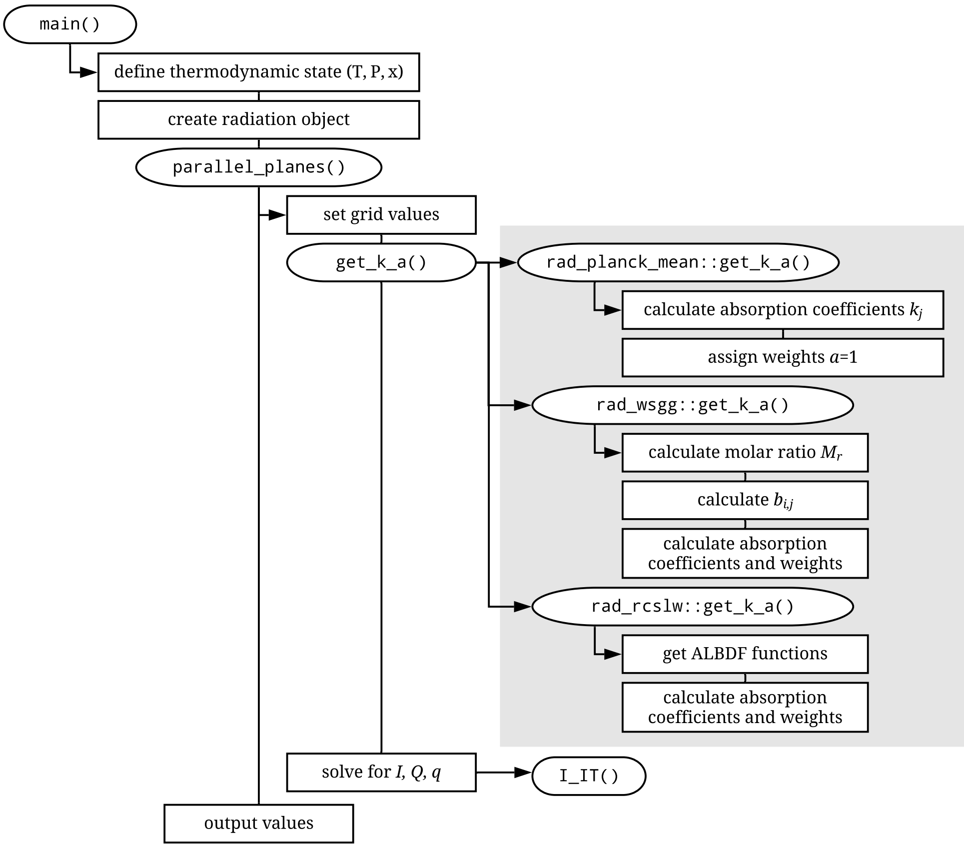

Figure 1: Example workflow diagram. Highlighted areas are part of the RadLib library; other areas represent example infrastructure for using the package.

about something else entirely, your radiation model (or lack thereof) becomes a source of error that may be very difficult to separate from other sources of error in your simulation study (i.e. soot modeling studies). As of right now, if you want any detailed radiation treatment, you have to code it yourself, which is difficult and requires external validation.

RadLib can make researchers' lives easier by providing a library of prevalidated radiation models that can be switched out with no difficulty. Additional models can be added easily using the provided modular framework. Researchers can even use Radlib as a tool for comparing models. No need to code multiple different complex radiation models yourself just to decide which one works best for your simulation. And no need to puzzle through the literature to figure our which model(s) might be best for your simulation, either; instead, you can test them yourself. It saves tons of time and effort that researchers can now put toward results rather than code or model development.

By putting all these models side by side in a modular framework, RadLib also provides a structure on which new or altered radiation models can be tested against existing ones. Maybe more comparative studies can be done, especially for more complex simulation cases. We plan to add radiation models as appropriate to RadLib, too.

RadLib opens up new horizons for CFD simulations, especially in combustion cases. It is designed such that it can be easily incorporated into existing research codes. Our group plans to use it with ODT [CITE OTHER SOFTWAREX PAPER HERE?] alongside soot model library (in development) to study late-flame phenomena such as soot oxidation, flame extinction, and soot-flame breakthrough. These topics in particular are difficult because they require accurate simulation data over a relatively long computational time in addition to good models for both soot chemistry (which is an active research area) and radiation heat transfer (which is the difficult part that RadLib can solve). Using RadLib for such studies allows us to separate and quantify sources of error that may occur due to various models, which is super difficult if you only have one model to work with or it hasn't been validated well.

RadLib can also be used outside of combustion CFD research for anything involving radiative heat transfer. Potential research areas that could benefit include atmospheric and climate sciences, interstellar phenomena, improving efficiency of energy-producing processes, safety in chemical plant design, etc. We aren't experts in these areas, but radiative heat transfer is a universal phenomena that applies to any system or process that involves heat transfer.

This software has not been used outside of the current research group, and it is not used in any commercial settings at this time.

6. Conclusions

Set out the conclusion of this original software publication.

7. Conflict of Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgements

The authors extend special thanks to Hadi Bordbar for assistance with the WSGG model and to Vladimir Solovjov and Brent Webb of Brigham Young University for their insights and assistance with the RCSLW model.

References

[1] N. Smith, J. Gore, J. Kim, Q. Tang, Tnf workshop radiation models (2003).
URL `https://tnfworkshop.org/radiation/`

[2] W. L. Grosshandler, Radcal: A narrow-band model for radiation calculations in a combustion environment: Nist technical note 1402.

[3] R. S. Barlow, A. N. Karpetis, J. H. Frank, J.-Y. Chen, Scalar profiles and no formation in laminar opposed-flow partially premixed methane/air flames, Combustion and Flame 127 (3) (2001) 2102–2118. doi:10.1016/S0010-2180(01)00313-3.

[4] R. Barlow, N. Smith, J. Chen, R. Bilger, Nitric oxide formation in dilute hydrogen jet flames: isolation of the effects of radiation and turbulence-chemistry submodels, Combustion and Flame 117 (1-2) (1999) 4–31. doi:10.1016/S0010-2180(98)00071-6.
URL `http://www.sciencedirect.com/science/article/pii/S0010218098000716`

[5] J. H. Frank, R. S. Barlow, C. Lundquist, Radiation and nitric oxide formation in turbulent non-premixed jet flames, Proceedings of the Combustion Institute 28 (1) (2000) 447–454. doi:10.1016/S0082-0784(00)80242-8.

[6] X. Zhu, J. Gore, A. N. Karpetis, R. S. Barlow, The effects of self-absorption of radiation on an opposed flow partially premixed flame, Combustion and Flame 129 (3) (2002) 342–345. doi:10.1016/S0010-2180(02)00341-3.
URL `http://www.sciencedirect.com/science/article/pii/S0010218002003413`

[7] P. J. Coelho, O. J. Teerling, D. Roekaerts, Spectral radiative effects and turbulence-radiation-interaction in sandia flame d, in: Barlow, Robert S., Pope, Stephan B., Masri, Assaad R., Oefelein, Joseph C. (Ed.), The Proceedings of the Sixth International Workshop on Measurement and Computation of Turbulent Nonpremixed Flames, 2002.

[8] M. H. Bordbar, G. Wecel, T. Hyppanen, A line by line based weighted sum of gray gases model for inhomogeneous co2–h2o mixture in oxy-fired combustion, Combustion and Flame 161 (9) (2014) 2435–2445. doi:10.1016/j.combustflame.2014.03.013.

[9] H. Bordbar, G. C. Fraga, S. Hostikka, An extended weighted-sum-of-gray-gases model to account for all co2 – h2o molar fraction ratios in thermal radiation, International Communications in Heat and Mass Transfer 110 (2020) 104400. doi:10.1016/j.icheatmasstransfer.2019.104400.
URL `http://www.sciencedirect.com/science/article/pii/S0735193319302660`

[10] L. S. Rothman, I. E. Gordon, R. J. Barber, H. Dothe, R. R. Gamache, A. Goldman, V. I. Perevalov, S. A. Tashkun, J. Tennyson, Hitemp, the high-temperature molecular spectroscopic database, Journal of Quantitative Spectroscopy & Radiative Transfer 111 (15) (2010) 2139–2150. doi:10.1016/j.jqsrt.2010.05.001.
URL `http://www.sciencedirect.com/science/article/pii/S002240731000169X`

[11] V. P. Solovjov, F. André, D. Lemonnier, B. W. Webb, The generalized slw model, Journal of Physics: Conference Series 676 (2016) 1–36.

[12] V. P. Solovjov, F. André, D. Lemonnier, B. W. Webb, The rank correlated slw model of gas radiation in non-uniform media, Journal of Quantitative Spectroscopy & Radiative Transfer 197 (2017) 26–44. doi:10.1016/j.jqsrt.2017.01.034.

[13] J. Badger, B. W. Webb, V. P. Solovjov, An exploration of advanced slw modeling approaches in comprehensive combustion predictions, Combustion Science and Technology 012022 (676) (2019) 1–17. doi:10.1080/00102202.2019.1678907.

[14] V. P. Solovjov, B. W. Webb, Slw modeling of radiative transfer in multicomponent gas mixtures, Journal of Quantitative Spectroscopy and Radiative Transfer 65 (2000) 655–672.

[15] V. P. Solovjov, B. W. Webb, An efficient method for modeling radiative transfer in multicomponent gas mixtures with soot, Transactions of the ASME 123 (2001) 450–457.

[16] V. P. Solovjov, B. W. Webb, Multilayer modeling of radiative transfer by slw and cw methods in non-isothermal gaseous medium, Journal of Quantitative Spectroscopy and Radiative Transfer 109 (2) (2008) 245–257. doi:10.1016/j.jqsrt.2007.08.015.

[17] V. P. Solovjov, D. Lemonnier, B. W. Webb, The slw-1 model for efficient prediction of radiative transfer in high temperature gases, Journal of Quantitative Spectroscopy and Radiative Transfer 112 (7) (2011) 1205–1212. doi:10.1016/j.jqsrt.2010.08.009.

[18] V. P. Solovjov, D. Lemonnier, B. W. Webb, Extension of the exact slw model to non-isothermal gaseous media, Journal of Quantitative Spectroscopy and Radiative Transfer 143 (2014) 83–91. doi:10.1016/j.jqsrt.2013.10.008.

[19] B. W. Webb, V. P. Solovjov, F. André, An exploration of the influence of spectral model parameters on the accuracy of the rank correlated slw model, Journal of Quantitative Spectroscopy and Radiative Transfer 218 (2018) 161–170. doi:10.1016/j.jqsrt.2018.06.023.

Current executable software version

Ancillary data table required for sub version of the executable software: (x.1, x.2 etc.) kindly replace examples in right column with the correct information about your executables, and leave the left column as it is.

Nr.	**(Executable) software metadata description**	**Please fill in this column**
S1	Current software version	TODO 2.1
S2	Permanent link to executables of this version	TODO For example: $https : //github.com/combogenomics/DuctApe/releases/tag/DuctApe - 0.16.4$
S3	Legal Software License	MIT license (MIT)
S4	Computing platforms/Operating Systems	Linux, OS X, Microsoft Windows
S5	Installation requirements & dependencies	TODO
S6	If available, link to user manual - if formally published include a reference to the publication in the reference list	TODO For example: $http : //mozart.github.io/documentation/$
S7	Support email for questions	davidlignell@byu.edu

Table 2: Software metadata (optional)