White paper on Release Requirements for Legacy Model Codes

Mark Marley (NASA Ames Research Center)
Jonathan Fortney (Univ. of California at Santa Cruz), Richard Freedman (SETI Institute), Peter Gao (Univ. of California at Berkeley), Roxana Lupu (BAERI), Caroline Morley (Harvard University), Tyler Robinson (Northern Arizona University), Didier Saumon (Los Alamos National Lab)

The movement towards widespread open source release of astrophysics codes is a product of the 21st century. This white paper briefly addresses the issue of how to handle 20th century legacy code within the modern open source environment.

Many researchers still working today began their careers long before routine public release of codes was even imagined as a future possibility. Code development for astronomy was often done by researchers with little to no formal training in coding best practices, code management, change control, or other common modern procedures. Instead code was often assembled from modules passed down from mentor to mentee, collaborator to collaborator, and from obscure legacy sources. Improvements and modifications were made to bring the code up to the level of stability and verification required to address a given problem, but not sufficient to allow arbitrary other people, even other domain experts with their own comparable codes, to run. Code developers knew the limitations of their codes and did not attempt to work outside of those limitations. With limited funding and resources such codes have continued to be maintained—sometimes for decades—with a focus on producing incremental improvements to "what works" rather than producing polished code that would gracefully handle extraordinary requests (e.g., to compute qualities outside of known thermodynamic conditions where the code was known to work).

Our group's radiative-convective equilibrium code, which we internally call the 'EGP' code, exemplifies this type of development history. Various modules have been combined from sources as diverse as a 1960's era punchcard FORTRAN Mie scattering code, a Titan atmosphere code, a white dwarf atmosphere code, snippets contributed by various collaborators, and various public libraries including from Numerical Recipes and a NASA archived code library. In places comments range from highly detailed, to detailed, but in a foreign language (French), to sparse. Successful compilation requires that just the right compiler switches in Intel Fortran (but *not* gfortran) be used. While this might seem a Frankenstein horror to some, the code has been extensively validated against various analytic tests and other similar codes. Originally assembled by Marley with later important inputs from Jonathan Fortney and Caroline Morley, the code is robust, but has a steep learning curve to use. Despite these limitations science to which this code has contributed has garnished well in excess of 10,000 citations, which points to the robustness of the science produced despite the challenges in running the model.

While it is true that with considerable effort, likely at least one person-year of a professional programmer's time, the code could be cleaned up, made easier to use, and documented. Tests could be done to test all input parameters to be certain that they are physically consistent with the capabilities of the code, and a hundred other comparable improvements made. The cost of such work would conservatively exceed $400,000 in programmer time, benefits, and overhead (at least one year to translate the code to a

modern language, add extensive input/output and runtime verification checks and validate it). Historically our own group has instead focused on using grant funds to add capabilities to the code or apply it to various new problems to enable new and exciting science. It is extraordinarily unlikely that such funds would be made available to bring a legacy code up to modern standards solely to satisfy well intended desires for community code releases that would be interested to few—if any—astronomers trained in more modern coding environments such as Python. If forced to support such a conversion without new funding and instead of doing science such a requirement would simply be devastating to our small research group while yielding little to no measurable community benefit.

Furthermore a strong case can be made that for many legacy codes it is the input assumptions and output results that are of interest and which can be validated against other codes computing the same physics. Our code's output, for example, is a temperature-pressure profile of an atmosphere with various specified properties that is in radiative-convective equilibrium. Other researchers can make the same set of original assumptions and compare their results to our temperature profiles which we publish and make freely available. Having access to the code itself would have almost no value to others given its complex heritage and difficulty of use. Helping multiple outsiders to use this legacy code could potentially evolve into a large amount of time and again would doubtless not be supportable by any NASA funding program.

We stress that we are not opposed to releasing codes when feasible. The "Ackerman and Marley" cloud code is freely distributed by our group for example. This code was written in 2000, is well documented, and easily compiles and runs. The code has been distributed to multiple groups who have published several papers with neither Ackerman nor Marley as co-authors. This is an example of where making code freely available to the community makes perfect sense.

We are aware of numerous other legacy codes with similar complex histories and challenging execution. The specific examples given here are meant to be representative of the many other codes used by senior astronomers who are nevertheless still active in the field.

In summary, we are asking for a "common sense" approach to requiring the distribution of legacy codes that have been constructed over decades, dating back to times when it was never even imagined that the code would someday be released. One criteria might be codes that have a published history prior to 2000 be excepted from open release requirements. Such a practical approach would avoid the large cost in dollars, time, and effort that would be entailed to bring legacy code up to modern standards.