Response to B. Lougheed

> *I have read your manuscript with great interest, as I am myself have written a Python package for calculating past irradiance (https://github.com/bryanlougheed/orbital chime/), although I haven't gotten around to finishing a manuscript yet due to my current employment situation. Therefore, I had a look through your package and found that it works as described and will be helpful for researchers using Python. In addition to the helpful comments from reviewers Marie-France Loutre and Michel Crucifix, I have some further minor comments:*

These comments are indeed useful and I thank Bryan Lougheed for taking the time to look at the software package and write his comments.

On referring to as "climatic precession" (ϖ).

> *In your manuscript, longitude of perihelion (ϖ) is referred to as "climatic precession" (line 23, 92 and 472). However, ϖ is not in itself precession (climatic or otherwise), but rather the orbital angle corresponding to longitude of perihelion, which is governed by changes in general precession (the combination of axial and apsidal precession), as you note in the second half of line 23. I would suggest to refer to ϖ as "longitude of perihelion", as is usually done in the literature.*

As always, the vocabulary might be problematic and may differ from authors to authors. Using usual notations (eg. Berger 1978), we have: $\varpi = \psi + \Pi$
where indeed ϖ is the "longitude of perihelion" but only with respect to the moving vernal point. Π is also the "longitude of perihelion", but this time with respect to the fixed stars. In the astronomical community, "longitude of perihelion" refers unambiguously to this angle Π, not to ϖ. The terminology "longitude of perihelion" is therefore unsufficient and likely to be ambiguous. Besides, ψ is also usually called the "general precession" and corresponds to the precession of equinoxes. In order to avoid confusion (many students do have difficulties to understand the difference between general precession, precession of the perihelion and climatic precession), it is therefore critical to specify that climatic precession is associated to the angle ϖ (and combines both the precession of the perihelion measured by Π and the precession of equinoxes measured by ψ). And to be more precise, "precession" is the phenomenon associated with the change of these different angles. So, indeed, it would be important to add the word "angle". I should therefore state either :
- ϖ is the longitude of perihelion with respect to the moving vernal point.
or :
- ϖ is the angle measuring climatic precession.
I will specify "climatic precession angle" in the revised manuscript.

> *The term "climatic precession" has typically been used in the palaeoclimate literature for e*sin(ϖ), an index that visualises both eccentricity and longitude of perihelion in tandem (Imbrie and Imbrie, 1982; Vernekar, 1972; most likely also earlier works), which you have described in line 476 as the "climatic precession parameter", but can be simply described as "climatic precession".*

I disagree, since e*sin(ϖ) is a mathematical mixture of two quantities: eccentricity e and "the angle measuring climatic precession" ϖ and it is important to identify correctly the individual astronomical parameters. In mechanics, "precession" is always a moving angle. When taking the sine and multiplying by eccentricity, this cannot be called "precession" anymore. Besides, it is quite a convention to take the sine (taking the cosine is also perfectly acceptable), therefore the word "parameter" seems useful to me to underline that this is an arbitrary choice. I did not invent the terminology "climatic precession parameter" and it is also widely used in the paleoclimate community, though indeed many people in the community tend to use "climatic precession" and even more often "precession" as a shortcut. The aim of this manuscript is to explain a few subtleties of insolation quantities and I therefore want to be as precise as possible in the terminology.

Elliptical vs numerical integration

> *Regarding your text that I have quoted below:*
>
> > *The simplest way to produce the corresponding time series for Quaternary climates is to compute the daily insolation for every day of the year, then select and sum the ones above the chosen threshold. This might not be the most efficient way. Besides, discretizing the year into an integer number of days (360 or 365 days) is not entirely rigorous. Discretization itself might lead to slight numerical errors.*
>
> *Here, you are discussing the disadvantages of using an a numerical integration instead of an elliptical integration. A numerical integration would indeed involve calculating irradiance across discrete intervals, but typically this does not involve breaking down the year into 360 or 365 discrete days. Mean daily irradiance can be calculated at any desired discrete interval resolution of the year (e.g. 1/10000th of a year resolution, or better), after which numerical integration can be carried out on the intervals using, e.g., a midpoint, trapezoidal, etc., approach. Berger et al. (2010) compare the results of elliptical and numerical integration for various irradiation quantities in their Tables 1a-c, 2a-c, where differences between the two methods are found to be near-negligible.*
>
> *Therefore, I would argue that the "numerical errors" or lack of "rigour" are not practical reasons for using elliptical integration instead of numerical integration. In any case, there is already a very small loss of precision in palaeo-irradiance calculations due a number of approximations, including, for example, the assumption of the orbital paramaters remaining constant throughout the tropical year (in particular in the case of ϖ). Therefore, precision becomes a somewhat irrelevant reason for choosing either elliptical or numerical integration.*

I agree that numerical integration can be as precise as desired when increasing the time step and that is is indeed possible to used "as many days as necessary" to get a good precision when, for instance, computing integrated insolation. Still, when computing the sum of daily irradiance, people generally use the actual duration of actual days, ie. 86400 seconds (eg. see Huybers, 2006: $J = \sum_i W_i * 86400$) since it looks much more natural, even if summing up 36500 days of 864 seconds would indeed be more precise. What I had in mind is not so much the possible computation methodologies, but much more the actual usual practice.

> *You mention efficiency, and I agree that this could be a decisive motivation for using elliptical integration, depending on computational resources and the size of the dataset (e.g. number of kyr and/or latitudes) needing to be computed.*

Efficiency is indeed important in many cases and algorithms for computing elliptic integrals are very efficient. But having an analytical expression for insolation series in general has also many other benefits, like a better theoretical understanding, or the possibility to apply other analytical methods (equations, derivatives, integrals, …). I certainly prefer to express results as analytic expressions, whenever possible, not only for efficiency.

Approach used to solve the Kepler equation for E

> *When solving the Kepler equation for E, there is no closed-form algebraic solution for E, so other methods must be used, typically involving an iterative approach that converges on the correct value for E to within a desired precision. I saw that in your py files you make use of the Halley's method option within SciPy's optimize.newton function. Perhaps you could consider citing the necessary literature as well as the SciPy package itself, seeing as solving the Kepler equation is of such importance for many irradiance calcuations.*

I am not sure that citing Edmund Halley would be very useful for the reader:
Halley (1694) "Methodus nova accurata & facilis inveniendi radices æqnationum quarumcumque generaliter, sine praviæ reductione". Philosophical Transactions of the Royal Society (in Latin).
But indeed, I need to (and will) cite the SciPy package.
Pauli Virtanen, et al. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17(3), 261-272. DOI: 10.1038/s41592-019-0686-2.

Python documentation strings

> *You have provided some documentation for your python functions as commented out text (using #) in your py files, above the def line of the functions. This documentation can only be accessed by the user by opening the py file in a text editor. You could instead include the documentation as a python docstring block (bookended using """) just below the def line, which would also make your documentation callable to the user's python environment using the help(functionname) or ?functionname commands.*

Thank you for the suggestion!
I will gladly implement this type of comments in the next release of the software.

> *I look forward to seeing the work published in Climate of the Past and hope that my comments have been of some help.*

Yes, they are certainly useful. Thanks again.