It was a pleasure to read through and learn about the new model Sed Trace, in the paper "**SedTrace 1.0: a Julia-based framework for generating and running reactive-transport models of marine sediment diagenesis specializing in trace elements and isotopes"** by Jianghui Du.

The model being described is a 1D (pseudo-)steady-state implementation of a sediment early diagenesis model, that is able to include carbon and nutrients, trace elements, and isotopes. This type of model is notoriously difficult to setup, apply and master as users must have a good command of numerical methods, and many sub-disciplines of (computational) geochemistry, and experience with approaches to validate and assess. Yet, the model scope is broadly relevant to many topical areas of geoscience, and developing new methods and tools to reduce the barrier to entry is of great interest to the community. The paragraph starting line 62 may benefit from the paper by Paraska et al (2014) which similarly highlights the challenge after doing a meta-analysis of models.

As suggested in the title, the software developed in this case is quite unique, as the user can setup a tailored simulation relatively simply in an xlsx spreadsheet, and then this information is parsed to create full-featured simulation in the Julia language. Outputs are similarly written to xlsx for subsequent plotting/analysis. As someone who has spent much time wrestling with fortran versions of similar models (including CANDI), I found the idea quite clever and from a user experience point of view the approach of having a lot of complex configuration information nicely summarised in a xls spreadsheet was a very clean and flexible way to do this.

At the heart of the model is an implementation of a reactive-transport code following the traditional approaches introduced in the foundational models of CANDI, STEADYSED etc, and the numerical methods for discretisation and ODE solution seem closest to CANDI, though updated (e.g. using the robust CVODE). In my view, it is elegantly described and I found it hard to fault, with care and attention to detail in the equations.

The model allows for a grid transformation (e.g. changing resolution over depth). The opening of section 3.1 may benefit from an improved opening/contextual statement setting the scene for this; whilst experienced users will get this concept, it might be confusing new users unclear what they should do or consider when making a grid and options available. Also, our experience with CANDI numerical code was that user's need to be aware of setting the grid parameters, and in some cases they would lose some mass conservation – this was albeit in more unsteady conditions, but this made me wonder if mass balance testing had been done on this code, or could be reported to the user to check for grid issues?

The model includes options for both kinetic and equilibrium chemical reactions, and is quite flexible in allowing users to formulate their reactions and their stoichiometry. The parser is able to allow users to easily tailor their settings and reaction network in Excel rather than having to edit code, and includes options to check stoichiometric balance. The approach to allow for Omega on reactions allows for precipitation/dissolution, and the code adopts a logistic rather than Heaviside style to smooth the numerical solution when quantities are close to equilibrium.

The pH solution is also well implemented, essentially conserving charge by adopting the concept of the equilibrium invariant concentration (similar to a component definition in PHREEQC). This can be a headache when dealing with boundary conditions, but this is addressed in the model. A comment here how this relates to or is different from traditionally used approaches may be warranted.

Overall, I found the model description excellent, and the software is full-featured and should be published. My main comment is that there is a fair amount of assumed knowledge required to digest these options and methods, and in the spirit of making this type of modelling easier for a wider audience, then the paper may also benefit from some careful additions of some leading or contextualising sentences before diving into some of the detail detail, though I realise this may make it longer than it already is.

On reading the paper I was keen to get using this and test the examples. I am not a Julia user, but have a reasonable level of skill with programming and modelling. I started with the GitHub repository and the associated documentation.
https://jianghuidu.github.io/SedTrace.jl/dev/guide/

I was able to install Julia, and follow the installation instructions to add the SedTrace package. I realise it is not the job of the author to teach users basics of Julia, but I somewhat embarrassingly was then unsure of how to get to the next step. I am not exactly sure where the installation put the git repo or how to run a Julia script. Whilst you have the workflow section I was not clear on what to type at the prompt. After an hour of training myself on Julia I decide to use the Visual Studio Code Julia plugin, and then click “Run” on the main script. I’m running the phbb example: main.pHBB1991.jl. This gave a few package issues, which I was able to resolve as:

```
  import Pkg; Pkg.add("NonlinearSolve")
  import Pkg; Pkg.add("DataFrames")
```

then many other errors, meant that the only step that complete was creating “model_parameter_template.pHBB1991.uniform.xlsx”. At that point I got the error:

```
Fatal error:
ERROR: MethodError: Cannot `convert` an object of type
JLD2.ReconstructedTypes.var"##Base.InvasiveLinkedList{Task}#332" to an
object of type Base.IntrusiveLinkedList{Task}

Closest candidates are:
  convert(::Type{T}, ::T) where T
   @ Base Base.jl:64
```

This experience was on a Mac Silicon 13.4 Julia 1.9.1.

I am sure I have just done something obvious wrong as a non-Julia (and Mac!) person. However, again in the spirit of making this type of modelling easier, I would really benefit from another section in the documentation, which gives exact steps and allows a new user to get running the examples using only the downloaded GitHub example repo.

References:

Paraska, D.W., Hipsey, M.R. and Salmon, S.U., 2014. Sediment diagenesis models: Review of approaches, challenges and opportunities. *Environmental modelling & software*, *61*, pp.297-325.