

Geosci. Model Dev. Discuss., referee comment RC1
<https://doi.org/10.5194/gmd-2022-22-RC1>, 2022
© Author(s) 2022. This work is distributed under
the Creative Commons Attribution 4.0 License.

Comment on gmd-2022-22

Anonymous Referee #1

Referee comment on "DFN Generator v2.0: A new tool to model the growth of large-scale natural fracture networks using fundamental geomechanics" by Michael John Welch et al., Geosci. Model Dev. Discuss., <https://doi.org/10.5194/gmd-2022-22-RC1>, 2022

The purpose of this manuscript is not clear to me. The stated aim is to give an overview of a software designed to generate discrete fracture networks based on algorithms previously introduced (or unified?) by the authors in Welch 2020. To do this, it is necessary to give some information of the algorithms, although the authors express a wish to limit repetition of material (see also remarks regarding how much information to introduce). Despite this, there are constant referrals to previous works of the authors, moreover, the given examples are reproduced from previous works. Since most of the manuscript essentially provides a summary of previous work, it seems fair to ask what is the novel contribution of the manuscript? The goal of introducing a new software is not reached - note that beyond a few flow charts, and overall programming environment, there is little information on software aspects, specifically, the methodology is introduced in an algorithmic fashion. This must all be made much clearer for this manuscript to be brought to the standards of a traditional journal paper. To be clear, this is not to say that I did not find the content interesting; it provides (or recaptures?) a plausible solution approach to a problem which is both important and difficult.

Is it possible to couple the structure generator and DFN Generator iteratively, to mimic how fractures are often generated as a response to larger-scale features (faults)?

L153: Here a list of examples would be useful, in particular of open source alternatives.

L221-> Calculation of in situ stress: This is one (of many) examples of what to me is insufficient information. How exactly is the stress calculated? What effects are included

and which are excluded? Do you account for stress rotations due to fractures under shear stress? How about stress release due to slippage? Is there communication between fractures in different gridblocks? Probably, some of the effects I rattle off are completely irrelevant for the processes under investigation, but the reader cannot judge this without knowing more about the approach; to be referred to read a chapter in a separate book is extremely unsatisfactory.

L234: Why 0.2% and not some other number? Is the algorithm sensitive to this number? Can the user control it?

The relation to the book (research monograph, I presume) Welch 2020 is problematic. It is understandable that the authors do not want to repeat information already given in that publication - the current manuscript already is on the long side. Still, the current manuscript is also very far from being self-contained, there is no way that I can understand the manuscript without looking up a lot of formulas in Welch 2020; indeed, from the frequent referrals, I have the feeling it would be necessary to study substantial parts of it. Another way to put this is that many of my questions and comments relate to algorithmic aspects that were supposed not to be the focus of the manuscript. Whether it is possible to give the necessary information in a manuscript that is fit for the journal seems like a question for the authors to answer, possibly in collaboration with the editor.

L242: How are the growth parameters defined? What do they describe, and which role do they play in the wider algorithm? Answers to these questions may be in previous publications, but the information as it is given now mainly serves to frustrate the reader.

L349 (and other places): The concept of a 'realistic' fracture network is both interesting and, to me, somewhat problematic. One can think of at least three ways in which a network can be more or less realistic:

- * The methodology used to generate the network can adhere to physical processes.

- * The geometric pattern

- * The network's physical properties.

Section 2.2.1 The paper is arguably on the long side, and it is natural to look for parts that can be shortened. In that context, one can easily cut 1-11/2 page from this section by streamlining the presentation (treat x and y / u and v jointly, treat mappings between coordinate systems in a unified manner etc.), without sacrificing accuracy in the presentation.

L434: What if the angle is larger, either because that's the orientation in which the fracture was generated, or that the grid block is rotated later on due to larger-scale formation? Please comment on how this is handled, and on the expected accuracy of the implementation in such cases.

Can you accommodate fractures shaped as ellipses?

I see how processing each gridblock independently allows for extremely efficient computations. Still, will this not give issues with discontinuities between gridblocks (clearly there is no reason to expect the fracture distribution to be homogeneous, but still, artificially introduced parameter heterogeneities are seldom a good thing).

L516: Such references to specific equations in previous works are not compatible with a manuscript being self-contained.

L572: And if the strikes are not perpendicular - what happens then?

L580: Can the stress shadow of a fracture extend beyond a grid block? If not (and this seems to be the case), this must create artifacts in the generated networks.

L590: How are such cases detected?

L618: Is the strike uniform throughout the gridblock? This will surely simplify calculations, but it would also limit the range of applicability for the modeling tool. As a very general question: Can such a range of validity be identified?

L647: I believe the link to the GitHub project is wrong (should be DFNGenerator without hyphen?)

Figure 9: The figure quality is too low, I need to zoom to 200% to be able to read it. This applies to most of the figures presenting results.

How is the integrity of the code ensured, in particular if external developers are to contribute code (or bugfixes)? Is there a test framework? What is the process of approval for suggested changes? I found no information on the GitHub page. This is important

information for anyone considering to contribute to an open-source project - before I know what are the ground rules for a project, I would not invest time in it.

P30, first bullet point: If my understanding of how the tests are set up, they confirm that the conversion from an implicitly to an explicitly defined DFN is correctly implemented, but there is no testing of the implicit representation in the first place. Moreover, the word confirm is unfortunate here (following Popper). Many formulations can be chosen without going too far into logic, for instance that the test support that the implementation gives a consistent conversion.

~L745: It seems to me that the size of gridblocks and timesteps are important parameters that control the balance between different form of accuracy (however that is to be defined) in the model; and presumably also between model property and computational cost. Have these balances been tested in any way? Are the guidelines for how to progress?

Section 4.2.1, and in particular Figure 11: Which metrics should be used to compare the figures (and which was used to identify the best fit)? What is meant by 'accurate representation'? Without knowing this, it is hard to know what to think about the performance of the method. Also, is the goal to create a replica of the data (so that figures B and C ideally should be identical), or should figure C be thought of as one realization in a stochastic framework? I might have overlooked this information somewhere earlier, but it certainly is a very important point that probably should be emphasized more strongly.

Figure 14: Again, how should the figure be understood - in which sense is the aim to reproduce known data? The purple and blue lines are generated fractures - should they match the observed fractures (the cyan structure(?))? If so, I would say they are very far from doing so (almost regardless of which metric is being employed). If not, then what is the correct interpretation of the test?

L834: There is always room to do stochastic modeling, but I agree that if the data is very sparse, the resulting uncertainty may be too large to draw any meaningful conclusion beyond 'anything can happen'. This however, brings up the question of which type of uncertainty the model can handle, or reproduce. Fracture length, orientation, network topology, density etc. Please comment.

L839: Geologically consistent in which sense?

Conclusion: Depending on the methodology chosen, upscaling of permeability and (non-linear) elastic parameters will be rather large extensions. Essentially these are software projects on their own, and indeed many packages exist that aim to do such calculations. To me it seems a better approach to couple DFN generator to such existing software.