Review of "Characteristics of earthquake ruptures and dynamic off-fault deformation on propagating faults." by *Preuss et al.*

Boris Kaus, Mainz

This is a very interesting and extremely well-written manuscript that employs a novel numerical approach that includes both long-term tectonic processes and short-term earthquake rupture to study off-fault deformation develops. A 2.5D approach is employed in which the effect of a viscous substratum is taken into account as well. Benchmarks are presented that demonstrate the robustness of the results w.r.t. numerical resolution, particularly for a newly proposed dynamic fault-width algorithm. Simulations are carefully performed and clearly described in which the complexity of the models is incrementally varied, such that readers get a clear understanding of the relative importance of the effects. The simulations reveal that there are quite some differences in model behavior depending on whether a rate weakening or rate strengthening rheology is employed. The most realistic results in terms of having successive earthquakes, the development of localized fault growth as well as distributed deformation is obtained with a rheology that transitions from rate strengthening to rate weakening. The implications of the models are discussed, as are the limitations and a potential application to the Anatolian fault system.
I believe that this is a very nice contribution that should certainly be published.

As you will see below, I have two moderate comments and a few minor remarks. I believe they would further contribute to the clarity of the manuscript and hope that the authors are willing to address them during the revision stage.

1) Moderate comments

Yield stress criteria
You correctly write that at yield $F=0$, and you employ a standard Drucker-Prager yield function (eq. 9). Yet, your expression for the yield function (eq. 8) is incorrect, which can be best illustrated graphically:
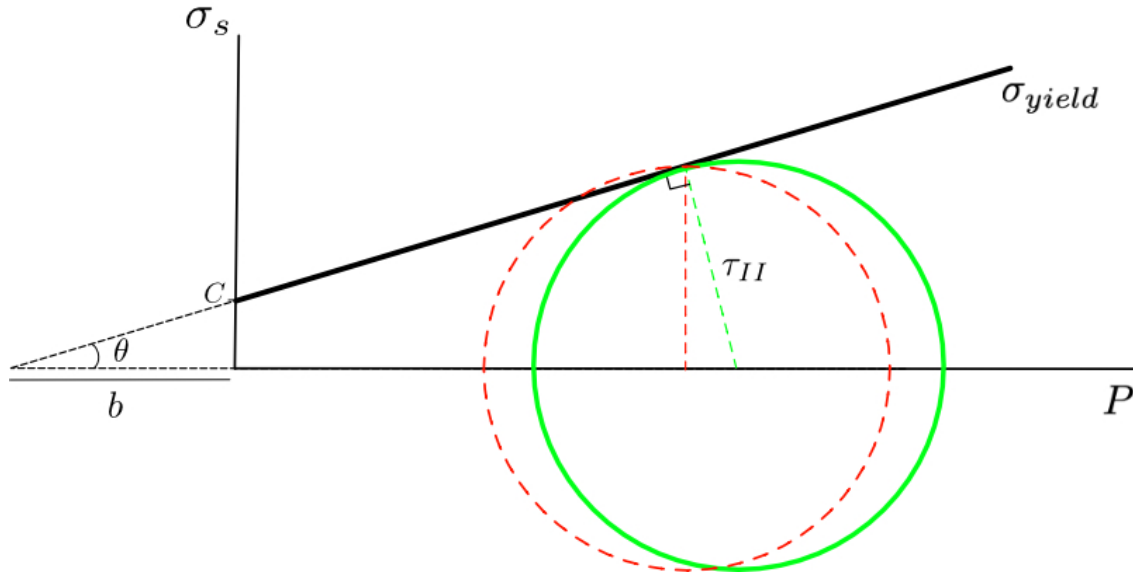
The plot shows the yield function (black) together with the Mohr-Circle (green circle, which has radius τ_{II}). At yielding ($F=0$), the Mohr-Circle exactly touches the yield stress function. This condition is *not* $\tau_{II} = \sigma_{yield}$ (your eq. 4), as this gives the red circle (which predicts a stress that is somewhat larger than the yield stress). Instead, we can use trigonometry to compute the condition for $F=0$. If we define the effective angle of friction θ as:

$$\tan \theta = \mu_l (1 - \lambda),$$

we can define the yield condition ($F=0$) as:
$$\tau_{II} = P \sin(\theta) + C \cos(\theta)$$

Since this is correctly described in the textbook of one of the co-authors of this manuscript, I suspect that it is incorporated correctly in the software. I also don't know how big an effect it will make on the results, even if this would not be the case (to be tested). Yet, in any case, it would be good if you can correct your description.

Elastic material parameters
Your choice of having the same values for bulk and shear moduli (table 1; both 50 GPa) results in a Poisson ratio of 0.125. That might be appropriate for already damaged rocks, but perhaps not so much for intact rocks. How sensitive are your results to the particular choice of Poisson ratio?

2) **Minor remarks:**
   - Table 1: I suppose that the host rock cohesion is 6 MPa, and not 6e6 MPa?
   - Fig. 3: you show an overview of several different models. Yet, are the snapshots chosen to have approximately the same plastic strain, deformation stage or time? Would be good to mention it.
   - Figure 4. It would be good to explain at the beginning of the figure caption that this figure concerns the RT model.

- Your movies are extremely large (some over 1 Gb!); it is certainly possible to create smaller movie-sizes from a set of pictures, and I believe that this is important for readers that do not have a high bandwidth connection. I personally use ffmpeg for this (free software for linux/mac). The following command creates a relatively modest-sized *.mp4 movie, if you have a series of pictures (here *.png named as MovieFrame.0001.png, MovieFrame.0002.png etc.):

```
ffmpeg -y -f image2 -framerate 10 -i MovieFrame.%04d.png  -vf pad="width=ceil(iw/2)*2:height=ceil(ih/2)*2" -f mp4 -vcodec libx264 -pix_fmt yuv420p Movie_loRes.mp4
```