Authors' Response to Reviews of

NorSand4AI: A Comprehensive Triaxial Test Simulation Database for NorSand Constitutive Model Materials

Ozelim et al.
Geoscientific Model Development - GMD, egusphere-2023-1690

RC: ***Reviewers' Comment***, AR: Authors' Response, ☐ Manuscript Text

Dear Prof. Dr. Le Yu, Handling Topic Editor of GMD

May this letter find you well.

We thank the valuable comments provided by the reviewers. A complete reply to each of the questions raised is hereby presented. The changes suggested by the reviewers were crucial to enhance the quality of the paper. One may notice, for example, that the manuscript total length went from 13 to 25 pages. We hope you find our manuscript suitable for publication and look forward to hearing from you in due course.

1. Reviewer #1

RC: ***Dear authors,***

This work aims to provide a useful synthetic dataset in assessing the liquefaction potential of soils for machine learning and deep learning tasks. I consider that more validation and scrutiny are required for such essential groundwork.

Table 1 shows the sampling range of the Nor-Sand model's input. However, justifications for such ranges are not mentioned in the manuscript. In the conclusion section, the authors noted that the Nor-sand model is used to assess the liquefaction potential of soils. The utility of this work should be highlighted throughout the article (not just in the conclusion session) to identify the target audience better and improve readership. Then, the ranges of the parameters can be justified.

AR: Dear reviewer,

About Table 1, indeed the original preprint lacked some clarifications on why the ranges adopted were of interest. The ranges adopted come from literature results on the behavior of real granular materials. An initial version of such ranges was first presented by Jefferies and Shuttle (2002) and has been updated ever since. The ranges presented in the paper are based on the latest compilation available, thus Table 1 reflects the information presented in the book by Jefferies and Been (2015). As mentioned, those authors have collected several triaxial tests carried out on a diverse set of granular soils, which eventually led to the creation of Table 1. In the updated version of the manuscript, we will include such information, describing why the values presented are of interest. The following text has been inserted right before Table 1

> The input parameters of the NorSand model are presented in Table 1. The sampling ranges adopted

come from literature results on the behavior of real granular materials. An initial version of such ranges was first presented by Jefferies and Shuttle (2002) and has been updated ever since. The ranges presented in Table 1 reflect the latest compilation available and reported by Jefferies and Been (2015). This way, practitioners will especially benefit from the datasets generated, since the parameters involved have been chosen as to represent real granular materials. Table 1 also present the meaning of each parameter in the column "Description".

Regarding the application to liquefaction modelling, indeed we had not presented this aspect throughout the article. In the updated version we completely reformulated the introduction to account for that, highlighting how the NorSand is used in that context to make the reader aware of the benefits of our approach. These changes can be seen on Sections 1 and 3, which are a completely rewritten Introduction and a section dedicated to the NorSand model, respectively.

RC: ***The output of the Nor-Sand model spreadsheet can be technically sound. However, the authors should test the sensitivity of the sampled dataset. For example, why would such sampling be the best dataset to represent the Nor-Sand model? Can one represent the model better with fewer samples, or more samples are required? Without showing a particular use (e.g., surrogate modeling, machine learning) or arguing the representativeness of the dataset, it is difficult to evaluate the value of such a dataset.***

AR: We are glad the reviewer pointed that out. At first, we performed a number of empirical simulations to check how big would the dataset be in order to represent the true behavior of the NorSand model. For simplicity, we ended up no including these studies in the manuscript.

On the other hand, after reading the comments, we had to devise a proper methodology (and not present just a series of empirical tests) to demonstrate in a robust and reproducible way that the dataset presented suffices to represent the NorSand model. This way, a completely new methodology has been proposed and applied to assess the quality of the sample size.

As suggested by the reviewer, the best way to show that the sample size is sufficient is to study how a model calibrated (or trained) on such dataset performs. So, we chose the most direct (and actually most important) learning task one could face while working with the dataset generated: back-calculation of the constitutive parameters of the model based solely on the triaxial test results. In short, from the triaxial tests we will learn the values of the parameters which govern the behavior of the material.

Section 4.2 of the updated draft presents all the details of the new framework considered. It now reads:

4.2 Sample size validation

The samples generated using the methods in the last subsection need to be sufficiently large in order to represent the general behavior of the NorSand model. The best way to show that the sample size is sufficient is to study how a model calibrated (or trained) on a given dataset performs. So, we chose the most direct (and actually most important) learning task one could face while working with the datasets generated: back-calculation of the constitutive parameters of the model based solely on the triaxial test results. In short, from the triaxial tests we will learn the values of the parameters which govern the behavior of the material.

This way, it is possible to recall that a total of 14 parameters (10 constitutive and 4 related to test conditions) are used to generate the triaxial test results (4000×10 array where 4000 denotes the number of time steps of the loading process and 10 is the number of quantities monitored during the test). From

last subsection's notation, Let In_i (shape 1x14) be the i-th row of the In matrix, which contains the constitutive parameters, and let ttu_i and ttd_i be the results of the triaxial test under undrained and drained conditions, respectively (4000x10 arrays, each) obtained by using these parameters on the NorSandTXL routine.

We will consider the following learning problem: From a sample of input parameters $In = In_{n,m}$, which considers n different types of soil and m different test configuration (therefore with nm rows), we will use the ttu_i (or ttd_i), for $i = 1, ..., nm$, to learn the vectors of parameters In_i, for $i = 1, ..., nm$. We wish to investigate what are the values of n and m that suffice to produce an accurate representation of the model. In order to do so, following standard learning tasks in a Machine Learning context, we need training, validation and testing data. It is worth noticing that our methodology needs to be robust, so we indeed need the validation dataset because hyperparameter tuning will be performed.

The dataset obtained by following the methods of the first subsection was generated by a Latin Hypercube Sampling (LHS) algorithm, which is known to provide low-discrepancy sequences of values (i.e., the samples are spread in the domain of the sampled variables). Despite being a really powerful technique, LHS does not have an interesting property: sequences obtained by LHS are not extensible. To put it simply, being extensible means that a sample of size j contains the values of the sample of size k, $j > k$. This way, it would not be possible to sub-sample from our original sample In in order to build smaller datasets without loosing the space-filling capability of the dataset. This way, we needed to consider another sampling scheme to perform our investigation.

We chose to combine two quasi-Monte Carlo low discrepancy sequence generation techniques (Sobol (Sobol, 1967) and Halton (Halton, 1960)), which are also extensible, to perform our tests. In that case, we generated a dataset with $n = 2048$ and $m = 42$ using Sobol sampling for the constitutive parameters (10 parameters) and Halton sampling for the experimental test condition variables (4 variables) using the SciPy Python package (Virtanen et al., 2020). Both sequences have been scrambled (Owen and Rudolf, 2021) to improve their robustness for space filling. By using these parameters, we ran the NorSandTXL routine in the same manner as described in the first subsection and obtained the corresponding triaxial test results for both drained and undrained cases. Let us call this new dataset and $qIn_{2048,42}$.

By using the extensibility property of the sequences considered, 49 sub-samples were taken: $qIn_{n,m}$ for n in [32, 64, 128, 256, 512, 1024, 2048] and m in [6, 12, 18, 24, 30, 36, 42]. One may see that powers of 2 were used as sample sizes for the Sobol sampling scheme, which is standard and derives from its implementation in *scipy.stats*. It is worth noticing that, in general, none of the entries of $In_{n,m}$ will be in $qIn_{n,m}$, which indicates that using $qIn_{n,m}$ for training and validation and $In_{n,m}$ for testing does not allow for any data "leakage". Besides, there is a clear benefit in using $In_{n,m}$ as a test set: all the models will be tested on the same dataset.

For the learning task considered, we used the *scikit-learn* Python package (Pedregosa et al., 2011) and chose 4 algorithms: Ridge Regressor, KNeighbors Regressor and two variants of the Ridge Regressor which incorporate nonlinear mappings of the input and output values. The first two algorithms mentioned belong to two different classes: linear and neighbors-based regressors. They were chosen to illustrate how different types of algorithms learn our chosen task. The variants of the Ridge Regressor were chosen to account for nonlinearities by using the kernel trick. Considering the high dimensionality of the input datasets, using traditional kernels is not computationally feasible, so we used Nystroem kernels (Yang et al., 2012), which approximate a kernel map using a subset of the training data. By combining Nystroem kernels and Ridge Regressors, we can map the inputs to a nonlinear feature space

and then consider a linear regression on these features. This is a similar approach as the one considered to build Support Vector Machine Regressors, but with a slightly different regularization for the decision boundary.

We also considered mapping the output values (14 parameters, in our case) to the [0,1] range by combining the *scikit-learn* implementations of TransformedTargetRegressor and QuantileTransformer, which transforms the target values (outputs of the pipeline) to follow a uniform distribution. Therefore, for a given component, this transformation tends to spread out the most frequent values. It also reduces the impact of (marginal) outliers (Pedregosa et al., 2011). For all the algorithms considered, we also used a QuantileTransformer to preprocess the input values.

This way, Figure 1 presents the methodology proposed and applied to assess the quality of the sample size. In the present paper, the LHS-generated dataset with $n_{soils} = 2000$ and $n_{conditions} = 40$, whose input parameter matrix is $In_{2000,40}$, will have its sufficiency assessed.

It is possible to describe the workflow in Figure 1 as:

For n in [32,64,128,256,512,1024,2048]:

For m in [6,12,18,24,30,36,42]:

- For each simulated triaxial test corresponding to the parameters matrix $qIn_{n,m}$, select only the columns corresponding to ϵ_1, p', q and e (axial strain, mean effective stress, deviatoric stress and void ratio, respectively), which are the variables commonly measured and reported. The other 7 columns are manipulations of these three. This reduced simulation dataset is of shape 4000x4.
- Each triaxial test simulation may have different start/end values for ϵ_1, so it is important to "align" all the test considered. By alignment we mean that all the tests will have measurements for the same values of ϵ_1. This will enable us to use this variable as an index and, therefore, decrease the dimensionality of each triaxial test simulation from 4000x4 to 4000x3.
- Downsample the 4000 timesteps to 40, by using evenly spaced values on a logarithmic scale (function *logspace* from Python package *numpy*: more values in the beginning of the time steps, where more changes are observed). This reduces each simulated triaxial test corresponding to the parameters matrix $qIn_{n,m}$ from 4000x10 to 40x3. The concatenation of all triaxial test results corresponding to the parameters matrix $qIn_{n,m}$ shall be named $qInN_{n,m}$ and is of size $(nm, 40, 3)$.
- Perform a GroupKFold cross-validation scheme to find the best hyperparameters of an algorithm A using $qInN_{n,m}$ and inputs and $qIn_{n,m}$ as outputs. The loss function considered during the GroupKFold cross-validation is the mean absolute percentage error across all folds;
- Retrain the algorithm A using all $qInN_{n,m}$ and $qIn_{n,m}$ after fixing the hyperparameters as the optimal ones obtained during the cross-validation scheme;
- Test the trained algorithm A_t on In_{n_h,m_h}, where n_h and m_h are the hypothesized sufficient number of materials and test conditions, respectively;
- Obtain the mean absolute percentage error in the predictions of all the 14 input parameters corresponding to In_{n_h,m_h};

- Get the overall mean error, corresponding to all the input parameters.

As described, for training and validation, we considered a GroupKFold cross validation technique, which is a K-fold iterator variant with non-overlapping groups (Pedregosa et al., 2011). This approach makes sure no material (group) is present both in train and validation set, which would lead to data "leakage".

A Bayesian optimization was performed to look for the best hyperparameters using the cross-validation folds generated. This process was carried out using the *Hyperopt* Python package (Bergstra et al., 2015), which considers Tree-structured Parzen Estimators. The search space for the Ridge and KNeighbors Regressors are the ones considered in the *Hyperopt-Sklearn* Python package (Komer et al., 2014). For the Nystroem kernel, a custom search space was defined and consisted of: 'gamma' parameter uniformly on [0,1]; 'n_components' parameter as a random equi-probable choice among [600,1200,1800]; 'kernel' parameter as a random equi-probable choice among ["additive_chi2", "chi2", "cosine", "linear", "poly", "polynomial", "rbf","laplacian", "sigmoid"]; 'degree' parameter as the integer value truncation of an uniform random variable on [1, 10] and 'coef0' parameter uniformly on [0,1].

Finally, after the best hyperparameters are found, they are fixed and the algorithm A is retrained with the full dataset $qInN_{n,m}$. This calibrated version is then used to test the quality of the model on the triaxial test results corresponding to the dataset In_{n_h,m_h}. Then, the errors obtained for each model are plotted and analyzed. The reader may find the complete codes used to implement the steps above in (Ozelim et al., 2023b).

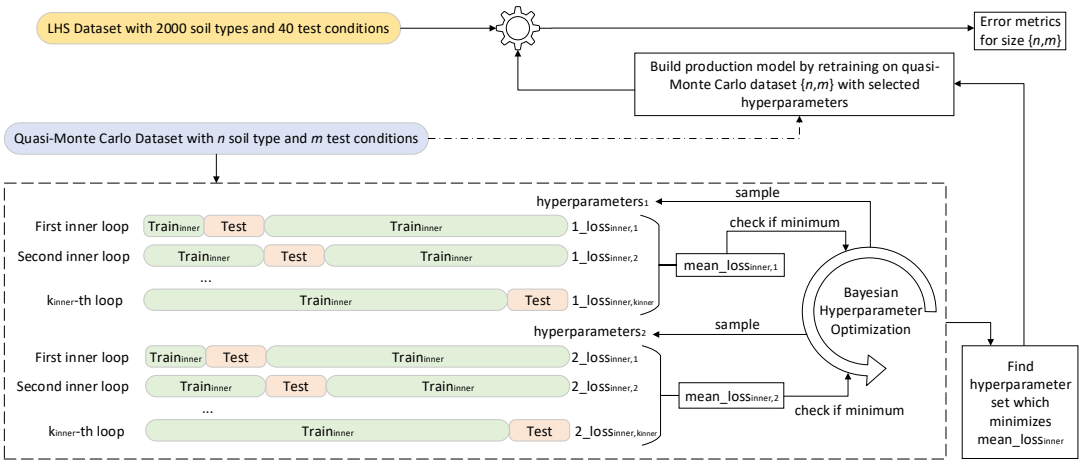

Figure 1: Methodology used to assess the sufficiency of the dataset containing 2000 soil types and 40 test conditions to represent the general behavior of the NorSand model

The results of applying this new methodological framework to assess the sufficiency of the datasets led to the results now presented in Section 6 of the updated draft, which follow below:

6. Technical Validation

Considering that the engine running the triaxial test simulations is the Excel spreadsheet presented in the book by Jefferies and Been (2015) a and that such spreadsheet has been extensively validated by both academia and industry, there is no need to discuss the technical quality of the dataset. On the other hand, it is necessary to show that $In_{2000,40}$ suffices to cover the general behavior of the NorSand models.

By following the methods previously described and plotting the mean absolute percentage error (MAPE) result of the 49 models (each trained and validated with samples of different sizes subsampled from $qIn_{2048,42}$) Figure 2 and 3 were obtained for drained and undrained conditions, respectively. The 4 algorithms considered were Ridge, KNeighbors, Ridge-K (with nonlinear kernel on inputs) and Ridge-KT (with nonlinear kernel on inputs and also QuantileTransformer on the outputs). It is clear in the figures that, for contours of 0.5% gains in MAPE, the sample size of 2000x40 is actually more than enough for the learning task considered. This can be stated by noticing that the contours with lower error encompass samples with an exponential range of sizes (the x-axis is in log scale). This indicates a really small gradient on the error in the nxm space, implying a good sample size. This happens for all 4 algorithms, indicating that not only linear and neighbors-based regressors have reached their maximum ability to learn, but also the nonlinear variants considered. It can be seen that the two nonlinear transformations applied (to inputs and to both inputs and outputs) present a similar behavior, although with considerably smaller MAPEs.

Due to the space-filling qualities of both $In_{2000,40}$ and $qIn_{2048,42}$, $qIn_{2048,42}$ can also be considered a sufficient dataset to represent the NorSand model.

RC: ***Since there are too many possible ways to improve the manuscript, I leave the authors to decide which aspects they would like to work on. I do not recommend the manuscript for publication at this stage.***

AR: We sincerely thank the reviewer for pointing out such interesting and important issues. We believe the updated version of the manuscript covers all the issues raised, and hope the paper is now suitable for publication. The new dataset as well as the codes used to perform the analyses are all available in Zenodo.

2. Reviewer #2

RC: ***General comments This paper tried to establish a comprehensive triaxial test simulation database for soil science. It is attractive to develop this kind of model, but I have some questions about this approach. Especially, what is the advantage improved from the previous approach needs to be clearly introduced and explained in this study. In addition, for better presentation quality, I strongly suggest reorganizing the manuscript because the current manuscript contains so many paragraphs. Please address the following questions.***

AR: Dear reviewer, We recognize the structure of the paper needed enhancement. In the updated version we incorporated the suggestions presented. About the paper, overall, there are two main advantages of using our results and datasets. The first one is that there are no known implementations of the NorSand model in Python. So, we built a bridge which connects a well-known VBA implementation to the Python environment. This allows other researchers to consider our code as a step in their Pipelines, allowing them to use the full power of Python packages (such as sklearn, TensorFlow, Pytorch etc) during their analyses. This has been included in the manuscript as:

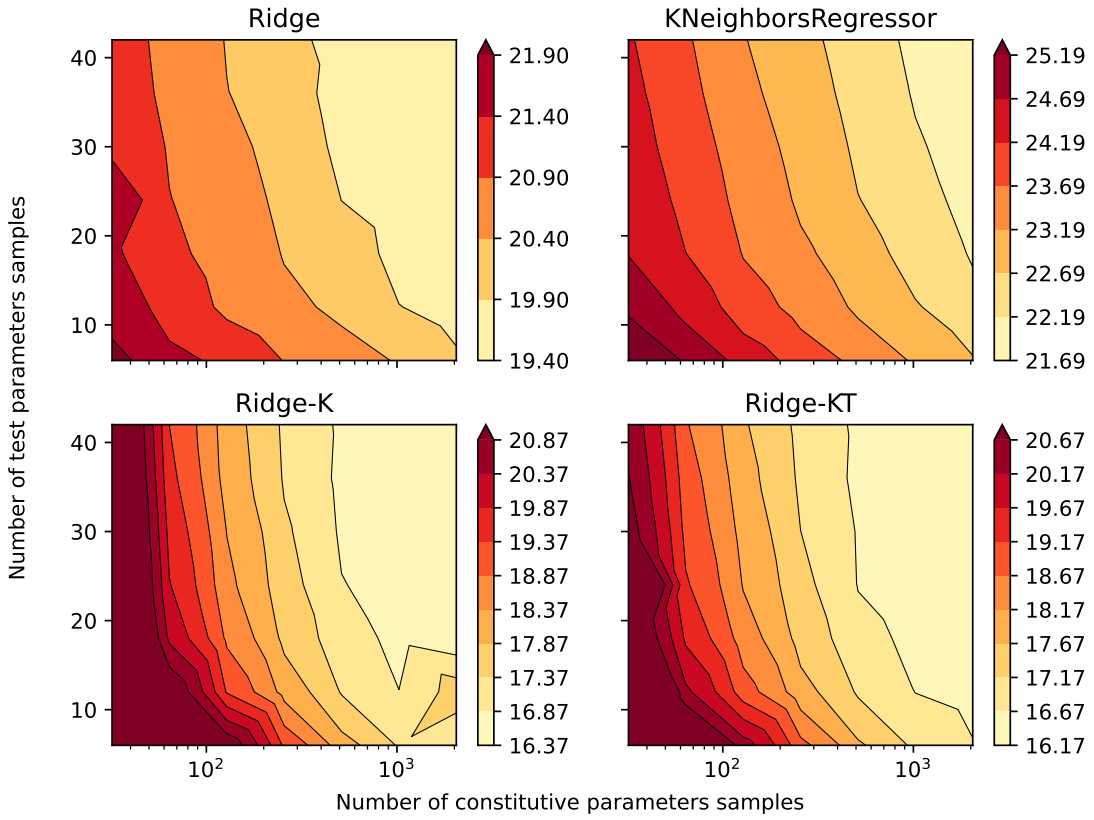

Figure 2: Mean absolute percentage error for all the 14 parameters after being back-calculated solely from drained triaxial test results.

> Also, only recently the NorSand method has been implemented in commercial Finite Element softwares (Rocscience, 2022; Itasca Consulting Group, 2023; Bentley, 2022). Besides, regarding open-source distributions, only the Visual Basic (VBA) implementation presented by Jefferies and Been (2015) is available. Thus, another open-source implementation easily integrated into ML and DL modelling pipelines is desirable. [...] Thus, the current paper aims to address three main issues: the quantity and complexity of synthetic datasets for nonlinear constitutive modeling of soils and the availability of open-source implementations of the NorSand constitutive model. [...] Then, the third aspect is considered by presenting an implementation which connects the well-known VBA implementation to the Python environment. We will use the VBA code as the "processing kernel" of our Python implementation, taking advantage of the years of tests and validation of the algorithm provided by Jefferies and Been (2015). This new Python code allows other researchers to use the full power of Python packages during their analyses involving NorSand.

The second advantage is that, each evaluation of the NorSand Model in the VBA code takes some time and effort to be completed. So, by providing massive simulation results, we save a considerable number of hours (even days) from other researchers which need such datasets.

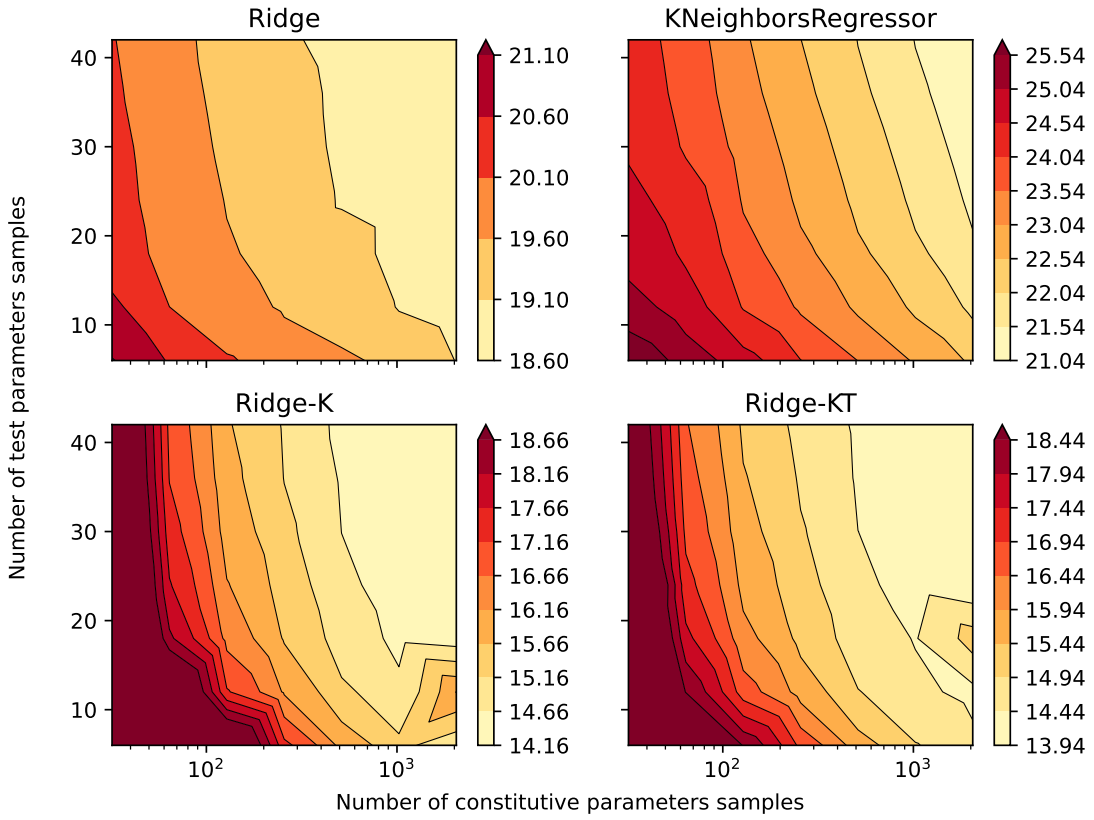

Figure 3: Mean absolute percentage error for all the 14 parameters after being back-calculated solely from undrained triaxial test results.

RC: ***Specific comments***

Many paragraphs throughout the manuscript: Repeatedly, there are lots of paragraphs, and some paragraphs seem to be merged. Please reorganize for better readability.

AR: We completely restructured the paper to account for such suggestion. The reviewer may check that all the section had insertions/changes.

RC: ***L26-34: These two paragraphs started with 'Montans et al. (2019) emphasize...' and ended with '(Montans et al., 2019)'. It is unclear what is authors' statements and what is referred statements. These two paragraphs may be merged.***

AR: Indeed, the paragraphs could be merged and the reference could be better placed. We implemented that in the new version of the manuscript.

RC: ***L57: From here, the introduction is suddenly changed to soil science. To fill the gap in the general introduction, some background information for soil science will be needed.***

AR: Indeed, the paper lacked a more comprehensive review on soil sciences. We added such information in the

updated version. Besides, a whole new section (Section 3 in the updated draft) has been created to discuss the NorSand model. We believe the new introduction is more complete and thank the reviewer for pointing out its previous deficiencies.

RC: ***L57: In addition to the above comment, it is hard to follow these previous studies. One idea is to prepare a brief summary for a clear introduction. Please reconsider.***

AR: This is a nice suggestion. We restructured the introduction to account for summaries on previous studies, focusing on soil sciences and liquefaction assessment.

RC: ***Table 1: A brief description of these parameters will be helpful for readers. The abbreviation of "OCR" should be explicitly defined within this manuscript. This might be covered within the previous studies, but why the sampling range can be set as listed in Table 1? Even though NorSandTXL has been already described in previous studies, a kinder introduction for Table 1 is required because this manuscript itself should be standalone.***

AR: About Table 1, indeed the original preprint lacked some clarifications on why the ranges adopted were of interest and also what each parameter means. For example, OCR stands for over consolidation ratio. A novel section (Section 3) has been added to the manuscript to better situate the reader with respect to the NorSand model. Besides, a new column ("Description") has been added to Table 1, clearly indicating the meaning of each parameter. The ranges adopted come from literature results on the behavior of real granular materials. An initial version of such ranges was first presented by Jefferies and Shuttle (2002) and has been updated ever since. The ranges presented in the paper are based on the latest compilation available, thus Table 1 reflects the information presented in the book by Jefferies and Been (2015). As mentioned, those authors have collected several triaxial tests carried out on a diverse set of granular soils, which eventually led to the creation of Table 1. In the updated version of the manuscript, we included include such information, describing why the values presented are of interest as well as what each parameter controls in the soil's behavior. This specific insertion reads as:

> The input parameters of the NorSand model are presented in Table 1. The sampling ranges adopted come from literature results on the behavior of real granular materials. An initial version of such ranges was first presented by Jefferies and Shuttle (2002) and has been updated ever since. The ranges presented in Table 1 reflect the latest compilation available and reported by Jefferies and Been (2015). This way, practitioners will especially benefit from the datasets generated, since the parameters involved have been chosen as to represent real granular materials. Table 1 also present the meaning of each parameter in the column "Description".

RC: ***L156-159 and L170-171: It is still unclear why this Python coding is needed and excel spreadsheet is not acceptable. This point seems to be argued in L102-107, but for practical use, how about calculating time by spreadsheet and Python, or how about the operational advantages?***

AR: About the Python coding, it is not that the excel spreadsheet is not acceptable. A first thing to notice is that there are no known implementations of the NorSand model in Python. So, we built a bridge which connects a well-known VBA implementation to the Python environment. In the end, we still rely on the VBA code as the "processing kernel" of our Python implementation. This new Python code allows, on the other hand, other researchers to use the full power of Python packages (such as sklearn, TensorFlow, Pytorch etc) during their analyses involving NorSand. The second advantage is that, each evaluation of the NorSand Model in the VBA code takes some time and effort to be completed (setting parameters, choosing simulation type, running the VBA macros and collecting results). So, by providing massive simulation results, we save a considerable number of hours (even days) from other researchers which need such datasets. The text added to account for

this specific issue was presented in the response to the first "General comment".

RC: ***L160 (Section 5): I was impressed that this section seems to be moved to the Appendix part.***

AR: We chose to move the coding part to the Appendix because we wanted to focus on the datasets in the main "body" of the manuscript. But we moved the codes back from the appendix and insert them in Section 7 (previously Section 5), as suggested. Besides, we included additional codes in the manuscript and created a github repo to make their sharing easier. We incorporated a new simplified code which simply outputs the simulation values instead of directly saving them to a .h5 file. This will make the incorporation of the code into existing Pipelines easier.

RC: ***Technical comments L51: Use "DNN". L89: No need to repeat "(Jefferies, 1993)" here. L105: Please insert after Table 1, and there maybe no need to change the paragraph here. Tables 1, 2, and 3: The caption should be placed at the top of the table.***

AR: We incorporated all the issues above in the final version of the manuscript. We sincerely thank the reviewer for such careful analysis on the paper, specially the code parts.

Sincerely,

Luan Carlos de Sena Monteiro Ozelim, D.Sc.
Corresponding author on behalf of all authors