[Re] A three-layer model for the flow of particulate suspensions driven by sedimentation - ReScience #113

Revision and comments

I. Hmamou (@IbrahHm), O. Delestre, C. Lucas

October 7, 2026

We sincerely thank the two reviewers, R. Saleem (@theRizwan) and K. Hinsen (@khinsen), for their complete work and the numerous tests they ran in order to improve the reproducibility of our results and our code. We have revised the manuscript and the code, and we added the scripts to generate the figures, following their recommendations. In order to be consistent, we have carefully re-generated each figure with the corresponding script.
In the next sections, we set out in detail these changes point by point.

Response to R. Saleem (@theRizwan) comments: reproducibility

Reproducibility: Major comments

1. *'algorithm.py' cannot be run from what is distributed, although the data is published.*

Response: We agree with these observations regarding the dataset integration, and we have implemented all of your suggestions to ensure a seamless reproducibility workflow. First, we have updated the root `README.md` and the tutorial in `examples/01_suspension_flow_exp/` to explicitly provide the exact DOI link, offering readers a direct route to download the raw `Suspension-flow-experiments.xlsx` data.
To resolve the layout and nomenclature discrepancies between this raw spreadsheet and the solver's expected input, we opted to ship a dedicated pre-processing Python script (`convert_excel_format_data.py`) rather than duplicating or hand-reshaping the data. This script automatically reads the raw spreadsheet, handles the nomenclature discrepancies (e.g., matching `E = 1.125` to the article's `E = 1.12`), ignores unused data blocks, and reshapes the data into the exact 6-sheet, 8-row layout required. As you rightly pointed out, shipping this programmatic conversion eliminates the risk of human error or misaligned block matching.
As you mentioned, because the simulated front position is inherently discretized by the grid resolution ($\Delta x \approx 0.00546$ m), the objective function minimized by `scipy.optimize.minimize_scalar` is essentially a step function with flat valleys. Consequently, minor floating-point differences between compilers (GCC vs. Apple Clang) during intermediate evaluations can lead the optimization algorithm to converge at slightly different points within the same valid flat band, or occasionally miss a very narrow lower band. We agree with your recommendation. In the revised manuscript, we have explicitly discussed this grid resolution limitation (line 295). We clarify that the calibrated C_{sed} values are subject to an uncertainty range defined by the mesh size, and we provide the corresponding valid parameter bands to guarantee theoretical reproducibility across all architectures. We checked that the tolerance we used in the minimization algorithm is consistent with the grid resolution. Note that we do not know how the precision of the C_{sed} values is linked to the precision of the algorithm. Consequently, when running the minimization

algorithm, we also checked that the three digits in (6) are significative.
Finally, the hardcoded path in `algorithm.py` has been removed. The script now accepts the path to the converted dataset dynamically as a command-line argument. These changes remove the need for the user to manually rename files or edit the source code, preventing any immediate crashes and making the workflow adaptable to any environment.

2. *The Python installation procedure works verbatim on Ubuntu 24.04 and fails on a stock macOS machine, and on neither does the interpreter get the README names.*

Response: As suggested, we have updated the `README.md` to explicitly state that the minimum required version is `Python` $\geq$ `3.10` rather than demanding exactly `3.13.2`. We have also added a clarifying note in the virtual environment instructions specifically for macOS users, advising them to explicitly call `python3.10` or `python3.13` when creating the `venv` to bypass the older stock `3.9` version. These changes ensure the `requirements.txt` pins will resolve on all operating systems.

3. *Neither plotting script writes a file, and their documented invocation gives the reader no signal.*

Response: We have implemented the modifications suggested:

- In `script_plot_deposit.py`, the `plt.show()` call has been removed and replaced with a dynamic `plt.savefig()` that reliably outputs the final `.pdf` file.
- In `script_animation_deposit.py`, we have completely replaced `plt.show()` with `ani.save()`, enabling the automatic rendering and saving of the animation as an `.mp4` file.

Both scripts now function in headless modes and leave behind permanent graphical artifacts for the reader to inspect.

4. *No script in either repository produces the article's actual figures.*

Response: To ensure that readers can directly regenerate the overlay of the model and the experimental data, we have introduced a dedicated post-processing script (`plot_deposit_vs_exp.py`). This new script automatically reads the simulated results and the experimental dataset, plots them together, and exports the final comparison as a PDF. Furthermore, the exact run instructions for this replication-specific workflow are now thoroughly documented in the new `examples/01_suspension_flow_exp/README.md` step-by-step tutorial. Note that you will not be able to replot the blue lines of Figure 4 and Figure 5 as the original data were not made available by their authors.

Reproducibility: Minor comments

5. *Further, Section 5 of the article relies on numerical results from the original authors in private communication, making any comparison completely unrecoverable from public artefacts. A simple acknowledgement in the article, or better yet, a release to print those curves along with the rest.*

Response: As we are not explicitly authorized to publicly redistribute the raw numerical data obtained via private communication from the original authors, we have followed your first suggestion. We have added a clear acknowledgement in Section 5 (line 329) of the revised manuscript stating that the original authors' numerical results are not included in the public artifact due to their private nature.

6. *The article states the optimisation searched over]0.5, 5[, while algorithm.py uses bounds=(1.0, 4.5). The reproduced values land inside both ranges, but the two statements should agree.*

Response: We have updated the domain in the article: $\Omega =]1.0, 4.5[$ in order to be consistent with the values of C_{sed} given in equation (6). Another choice of bounds can lead to other numerical approximations and to slightly different values of C_{sed}.

7. *According to the article, dx = 0.005 m, however the shipped code_parameters/params.ini you have is domain_length = 3.605 and cells_number = 660, giving dx = 0.00546 m, and that is the step that the error calibration moves in (comment 1). Which arrangement gave rise to Figure 2? As per the article, if params.ini is correct, it should have shipped with 721 cells; if params.ini is correct, the article text should say so.*

Response: The exact arrangement that gave rise to Figure 2 (and indeed all published results) is the one shipped in the `params.ini` file: 660 cells, which yields $\Delta x \approx 0.00546$ m. The value $\Delta x = 0.005$ m stated in the article was an rounding approximation of this true value. To maintain the integrity and reproducibility of the published results and figures, we must leave the codebase untouched at 660 cells. Consequently, we corrected the revised article (line 293) to accurately state $\Delta x \approx 0.00546$ m (660 cells), thereby aligning the manuscript with the computational reality.

Code clarity: Major comments

8. *The requirements that were presented in the README differ from what the build actually requires.*

Response: We apologize for this typographical error in the documentation. The true minimum required version is indeed `3.15`, exactly as stated in the `CMakeLists.txt` file. We have now corrected the README and the CHANGELOG to explicitly state `CMake` $\geq$ `3.15`, ensuring consistency across the repository and eliminating any artificial barriers.

Code clarity: Minor comments

9. *The CHANGELOG describes a "C++17 Finite Volume solver", but CMakeLists.txt sets CMAKE_CXX_STANDARD 20 and the README says C++20.*

Response: This observation is correct. In the spirit of maximizing cross-platform compatibility, we realized it is better practice to require the lowest standard that successfully compiles the codebase. Since our solver does not strictly require C++20 features, we have downgraded the requirement in `CMakeLists.txt` and the `README.md` to `C++17`. The codebase now uniformly enforces and documents the C++17 standard, lowering the barrier to entry for users with older compilers.

10. *CHANGELOG and the documentation banner both declare version 0.1.0, but no git tag exists, so there is no immutable reference to that version. Tagging would also let the article cite an exact code state.*

Response: We agree with this best practice. We have now formally tagged the repository with a new release version 1.0.0 (tag in git and reference in the article line 45) that encapsulates all the improvements and fixes made during this review process.

11. *CMake prints – Compilation completed successfully ! at the end of the configure step, before anything has been compiled. Moving it or rewording it would avoid a misleading message.*

Response: We agree. Printing "Compilation completed" at the end of the configuration phase (i.e., after running `cmake ..`) is indeed misleading, as the actual compilation only occurs when invoking `make`. We have reworded this statement in `CMakeLists.txt` to `message(STATUS "Configuration completed successfully ! Run 'make' to build the executable.")` to accurately reflect the build sequence.

12. *The README's example uses Csed = 3.1131 where equation (6) prints 3.113. Consistent, but easy to make identical.*

Response: We have updated the `README.md` example to use $C_{sed} = 3.113$, aligning it with the truncated value printed in equation (6) of the manuscript.

13. *In the article repository, .DS_Store, __pycache__/article.cpython-39.pyc and the LaTeX build files article.fdb_latexmk and article. fls are committed and worth removing.*

Response: We have removed these auxiliary and cache files from the article repository and added them to the `.gitignore` to ensure the repository remains clean.

Response to K. Hinsen (@khinsen) comments

Minor typos, pull request ***`Fix typos, minor stylistic improvements - # 1`***

Response: Thank you for pointing out these typos, we merged your changes.

I suggest adding the following paragraph (or something similar) to the beginning of your README.md:

Your computational environment must include:

- *a C++ compiler*
- *the build utilities make and cmake*
- *a Python 3 environment including the libraries NumPy, SciPy, Pandas, matplotlib, and openpyxl.*

Response: This paragraph has been added in the README in the `examples/01_suspension_flow_exp` directory.

A minor point: Eqs. 1a, 1b, 1c are described as the same as Eqs 10-14 of the original work. In the arXiv preprint, the numbers are 3.10 for 1c, 3.11 and 3.12 for 1a, 3.13 and 3.14 for 1b. The numbers may be different in the published version. Please check.

Response: We checked with the editor published version, the equation numbers are correct.