The Pencil Code Newsletter

Issue 2025/2

December 29, 2025, Revision: 1.55

Contents

We wish all readers a happy New Year 2026!

1 PCUM 2025 at CERN

The Pencil Code User Meeting (PCUM) 2025 was a great success combining both a school on early Universe physics and gravitational waves. (first week, October 20–24) and the actual user meeting (second week, October 27–31); see `https://indico.cern.ch/event/1548935/` for the program, group photos, slides of the talks, and their recordings. CERN provided a scenic setting for this event; see Figure 1. On the technical side, one of the insights from the meeting concerned the "Merge branch 'master'" commits; see §3 below.

Figure 1: View from the lecture room at CERN toward the mountains to the left of Mont Blanc.

2 PCUM 2026 in Alicante

The 22nd PCUM will be organized by Clara Dehman at the University of Alicante, Spain; see Figure 2. A poll to determine the preferred week for the meeting is currently open; interested participants are kindly invited to cast their vote at the following link: `https://docs.google.com/forms/d/e/1FAIpQLSddf3fDBfNivf9afTM37StIBoiEpTRwgNWvF1EQqcuzEZOS-A/viewform?usp=header`. Further details regarding the schedule will be communicated later. If there are questions, please check with Clara `<clara.dehman@ua.es>` via email.

3 Linear History and Git Usage

During the last PC user meeting it was unanimously decided by the participants that we like to preserve the linear commit history in the PC repository's `Master` branch. The PC steering committee was tasked to obtain more feedback on how to implement this from the

Figure 2: Poster ad of the 22nd PCUM at the University of Alicante, Spain. The time is still to be decided. Photo credit: University of Alicante webpage and GetYourGuide.

whole community. In the following, we reproduce those views.

3.1 By Philippe Bourdin

<Philippe.Bourdin@oeaw.ac.at>

received 28 December 2025

Options to prevent `Merge branch 'Master'` commits are: 1) on the server, 2) on the user's local checkouts, or 3) make only suggestions to users. Possible mechanisms we found on the internet so far would only check the text of the commit message, which we consider insufficient because users could change that text easily. GitHub seemingly offers such a prevention mechanism, where we do not know how this is implemented and if it could be tricked by altering the commit message.

The root cause of the issue is that someone executed a `git pull` without rebasing while there are remote changes and local commits. This is very often the case in a busy repository like ours. In principle, an alternative to `git pull --rebase` is to use `git fetch` and `git rebase origin/master` instead. There are lengthy discussions on the internet about this[1],[2]. We will not go into more detail here.

Looking at Fig. 3, another issue here is that `git blame` would erroneously indicate that Matthias is now the author of many lines in those files that were originally written by Fred and Axel in our example, which makes tracing these changes unnecessarily difficult. The reason is that basically Matthias committed Fred's and Axel's changes again in Matthias' own `Merge branch 'Master'` commit, of which he is of course the committer but not the original author. This problem scales with the length of the deviating history.

This whole topic has nothing to do with Subversion or the `subgit` software synchronizing our GIT and SVN repositories that live fully independently on our server. Instead, the problem exists purely in GIT alone in the sense that a previously linear history of the `master` branch appears to be rewritten and the local `master` branch relocated to a so-called 'anony-

[1] https://stackoverflow.com/questions/292357/what-is-the-difference-between-git-pull-and-git-fetch

[2] https://longair.net/blog/2009/04/16/git-fetch-and-merge/

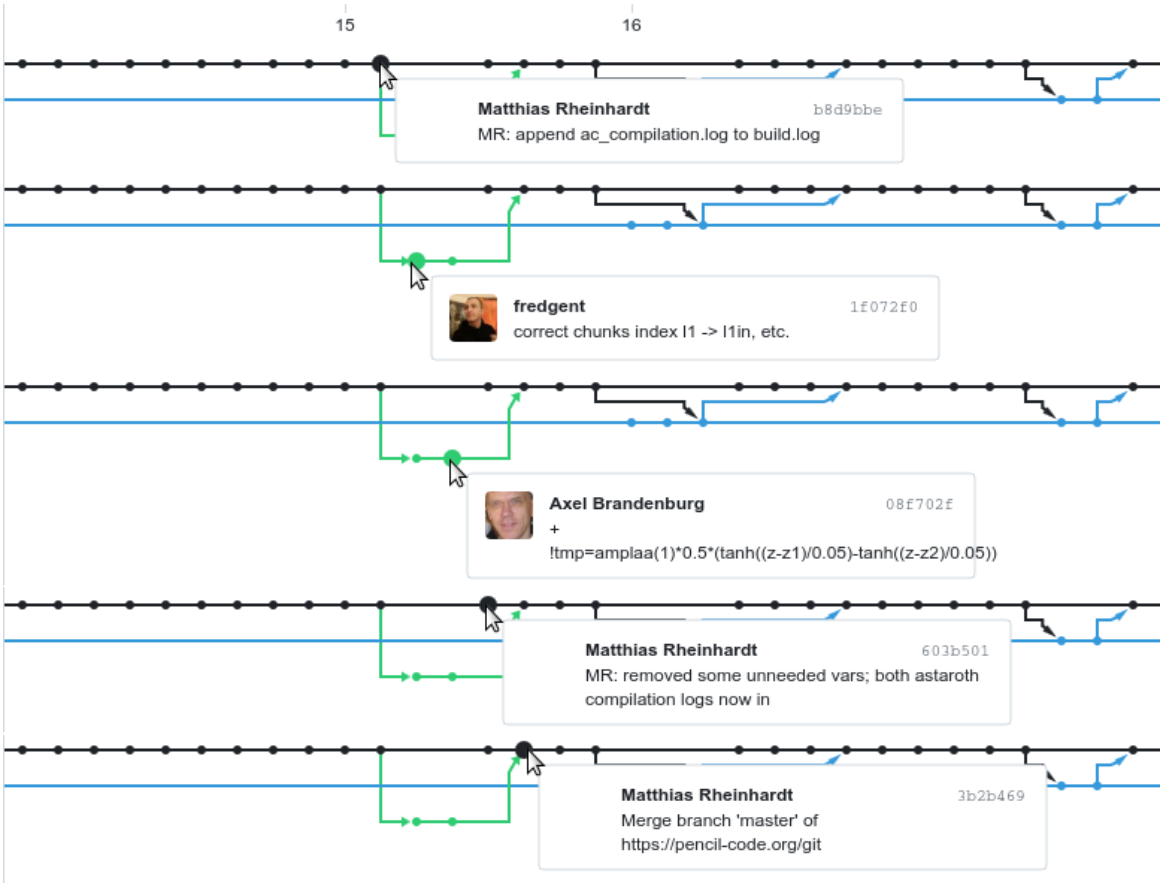

Figure 3: History graph of 15th of October 2025 as shown by GitHub network. From top: a) point of divergence, b+c) linear history of commit 1+2 in the original remote 'Master', d) a commit in Matthias' local 'Master', and finally e) erroneous merge is pushed and marks Matthias' local history now as the new linear history of the remote 'Master'. The green and black lines indicate the linear history of the remote 'Master' branch in its original and latest state, respectively. The blue line is from a remote branch that is not relevant here.

mous' branch when the `Merge branch 'master'` happens; see Fig. 3.

This relocation of some commits to an anonymous branch makes it particularly difficult for SVN users to find these commits, even though they are stored in the `shelves` branch of SVN that is explicitly created for the anonymous GIT commits – which can still be inspected with a checkout of all branches like this:

```
svn co https://pencil-code.org/svn/ pc-all
cd pc-all
svn log
```

My personal opinion is that the behavior of `git pull` not rebasing by default is a debatable design feature of GIT – actually, I would recommend not to use `git pull` at all. This is also the opinion of many commentators on the internet discussion in response to the links above. If we can, we should prevent users from making this mistake by all means possible: technical means on the server, on the user's local checkout, and with documentation.

3.2 By Illa R. Losada

`<illa.rivero.losada@gmail.com>`

received 26 December 2025

During the Pencil Code User Meeting 2025, we discussed at length recurring problems related to non-linear Git history on the `master` branch, in particular commits of the form `Merge branch 'master'`.

Such commits are not inherently wrong in Git, nor

are they necessarily the result of deliberate misuse. In many cases, they are produced automatically by Git when running a plain `git pull`, which by default performs a `fetch` followed by a `merge`. If this command is executed on a branch that has local commits and has diverged from the remote, Git will create a merge commit without the user explicitly intending to merge branches.

This behavior can occur in any pure Git environment. However, its consequences are significantly amplified in our repository setup, which accepts both Git and SVN commits. Git represents history as a directed acyclic graph (DAG), while SVN enforces a linear revision history. When a large or unintended Git merge is mirrored through the Git–SVN bridge, the resulting linearization can obscure ancestry, reintroduce outdated snapshots, and make recovery difficult or impossible.

As a concrete example, consider commit `60cf8c763`. This commit merged a branch last updated in 2017 (commit `316e7d608`) into the then-current `master` (commit `eff679045`). The merge was performed using a simple `git pull` by a user on the old branch, resulting in a single merge commit with the auto-generated message:

```
Merge branch 'master' of
https://github.com/pencil-code/pencil-code
```

Because the local branch had diverged from `master` for two years, Git attempted to reconcile two vastly different trees. This resulted in a merge commit affecting 1,415 files, with over 150,000 lines added and 91,000 lines removed. While technically valid from Git's perspective, this commit effectively reintroduced or deleted large portions of repository history, creating significant disruption and making subsequent merges more error-prone.

This example demonstrates that:

- Long-lived branches or edits on a very old checkout, combined with plain `git pull`, can produce massive, unintended merge commits.
- Auto-generated merge messages such as `Merge branch 'master' ...` are a warning sign of such operations.
- Even without `--force` pushes or rebase, repository history can be disrupted in ways that are difficult to undo.
- Hybrid Git+SVN servers exacerbate the risks because Git and SVN handle branching, merges, and history differently, making the consequences of large unintended merges significantly more severe.

In a standard Git environment, each merge commit records its full ancestry as part of Git's directed acyclic graph (DAG). Even after a large or unintended merge, both parent histories remain intact and inspectable, making such events visible and, in most cases, recoverable using standard Git tools (e.g., inspecting parents, reverting merges, or reconstructing history). When the same merge is mirrored into an SVN repo, this DAG structure must be flattened into SVN's strictly linear revision model. As a result, parent relationships are lost, merge semantics are discarded, and the commit appears in SVN as a single large linear change. This loss of structural information makes it difficult to determine the origin of changes, complicates reverts or bisection, and significantly hampers recovery.

The meeting consensus was that the primary mitigation should be **clear guidance and documentation**, rather than relying solely on technical enforcement mechanisms. Contributors are strongly encouraged to:

- use `git pull --rebase` instead of plain `git pull`,
- avoid merge commits on the `master` branch unless explicitly intended,
- use SVN directly if you are not familiar with Git's subtleties described in this newsletter.

These recommendations, along with a detailed explanation of the underlying problem, are now documented in the Pencil Code Read the Docs pages:

- History Rewriting Hazard (Git + SVN Server)
- Git on an SVN-Backed Server: A Post-Mortem

Community feedback is always welcome, especially regarding additional safeguards or workflow improvements. However, until reliable server-side blocking mechanisms are identified, the current emphasis is on informed and careful usage.

Users are therefore encouraged to read the documentation carefully and to follow the recommended workflows when contributing to the Pencil Code repository.[3]

[3] *Note added by Matthias:* As a technical measure for avoiding `git pull` to act without rebasing, applying `source sourceme.sh`

LCOV - code coverage report

			Coverage	Total	Hit	Missed
Current view:	top level - src	Lines:	44.2 %	131117	58001	73116
Test:	pencil-codecov.info	Functions:	52.5 %	6756	3545	3211
Test Date:	2025-12-29 07:17:31					

File	Line Coverage				Function Coverage			
	Rate	Total	Hit	Missed	Rate	Total	Hit	Missed
NSCBC.f90	39.8 %	1565	623	942	70.8 %	24	17	7
ascalar.f90	80.5 %	257	207	50	100.0 %	13	13	
bfield.f90	54.0 %	594	321	273	63.4 %	41	26	15

Figure 4: Code coverage report for the Fortran source files.

4 Code coverage reports

by G. Kishore, Inter-University Centre for Astronomy & Astrophysics, Pune, India <am25r009@smail.iitm.ac.in>

received 29 December 2025

We now have the ability to generate code coverage reports for both the Fortran and Python parts of the code. This allows one to view which functions/lines in a particular source code file are covered by the current set of tests.

4.1 Fortran

Make sure that `lcov` is installed,[4] and run

```
pc_auto-test -C --max-level=3 --coverage
pc_codecov --output-dir lcov_html
```

To view the code coverage report, open `lcov_html/index.html` in your web browser.

4.2 Python

Make sure you have a Python interpreter in your PATH and that the `tox` Python package (version >= 4) is installed.

```
python/tests/test-python-modules.py --full
```

now sets in `.git/config`

```
[pull]
        rebase = true
```

and warns about `rebase = false` settings in branch-specific entries.

[4] use `sudo apt install lcov` for ubuntu

```
1185          1889 :       call farray_register_pde('aa',iaa,vector=3)
1186          1889 :       iax = iaa; iay = iaa+1; iaz = iaa+2
1187               :!
1188               :!  If we want to evolve the current density.
1189               :!
1190          1889 :       if (lohm_evolve) then
1191   MIS       0 :         call farray_register_pde('jj',ijj,vector=3)
1192             0 :         ijx = ijj; ijy = ijj+1; ijz = ijj+2
1193               :       endif
1194               :!
1195               :!  Identify version number.
1196               :!
1197   HIT    1889 :       if (lroot) call svn_id( &
1198            727 :           "$Id$")
1199               :!
1200               :!  Writing files for use with IDL
1201               :!
1202          1889 :       if (lroot) then
1203               :         if (maux == 0) then
1204           229 :           if (nvar < mvar) write(4,*) ',aa $'
1205           229 :           if (nvar == mvar) write(4,*) ',aa'
1206               :         else
1207           498 :           write(4,*) ',aa $'
1208               :         endif
1209           727 :         write(15,*) 'aa = fltarr(mx,my,mz,3)*one'
1210               :       endif
1211               :!
1212               :!  Register EE as auxilliary array if asked for.
1213               :!  This must not be involved when the displacement current is being
```

Figure 5: Code coverage report for a single Fortran source file.

will then run the Python tests and generate a HTML report, `report.html`. This report will contain a link to a code coverage report for Pencil's Python module.

5 Python Postproc Meetings

by Illa R. Losada <illa.rivero.losada@gmail.com>

received 26 December 2025

During the last Pencil Code User Meeting at CERN (see Sec. 1), we decided to resume periodic meetings focused on the development and usage of Python as a postprocessing tool for the Pencil Code.

Full information about past meetings, decisions, outcomes, and upcoming sessions can now be found on the Python Postprocessing Meetings page.

Figure 6: HTML report for the Python tests

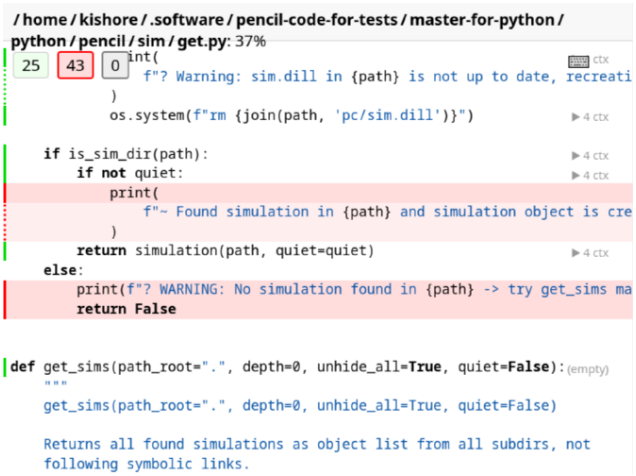

Figure 7: Code coverage report for a single Python source file.

As a follow-up, the first Python postprocessing meeting was held on Nov 26. During this meeting, the group discussed the meeting format and schedule, agreeing to hold Python postprocessing meetings every two months with rotating organizers. Key topics included testing and automated code checks, library installation via pip, updating and centralizing tutorials in ReadTheDocs, and improving real-time simulation visualization in Jupyter notebooks. Several action items were assigned, such as uploading Pencil Code Python as a pip package, enhancing documentation, and standardizing data formats.

The next meeting is planned for January (exact date TBD). We encourage all members of the Pencil Code community to participate, contribute ideas, and share experiences with Python postprocessing.

6 New ReadTheDocs Additions

by Illa R. Losada <illa.rivero.losada@gmail.com>
received 26 December 2025

The Pencil Code ReadTheDocs pages have recently been expanded with several autogenerated sections directly extracted from the source code, providing comprehensive reference material for developers and users alike.

- **Fortran Modules:** Description of all Fortran files with filtering options for easy navigation. See the full list at

  Fortran Modules.

- **Startup and Run-time Parameters:** Detailed list of parameters with their placement in configuration files (e.g., `start.in`, `run.in`, `print.in`), the associated module, default values, and descriptions if available, with filtering options for quick lookup, see:

  Startup and Run-time Parameters.

- **Boundary Conditions:** Description of available boundary conditions per module, with explanations of their meaning and usage, see

  Boundary Conditions.

- **Bin Scripts:** List of executable scripts in the `bin` directory, with short descriptions of their functionality, see

  Bin Scripts.

Additionally, the **Scientific Usage of the Pencil Code** document has been added, partly autogenerated to catalog publications making use of the code. The page includes references to journal articles and conference proceedings, classified by year and research topic. The primary applications lie in astrophysics, with additional work in meteorology and combustion. Explore the page here: Scientific Usage of the Pencil Code.

For guidance on how to add citations in the documentation, see the separate guide at Adding Citations.

These additions make it significantly easier for users to explore the code structure, runtime options, and scientific impact of Pencil Code.

7 PCUM25 developments

Already during the first coding session, a new dark matter autotest was constructed

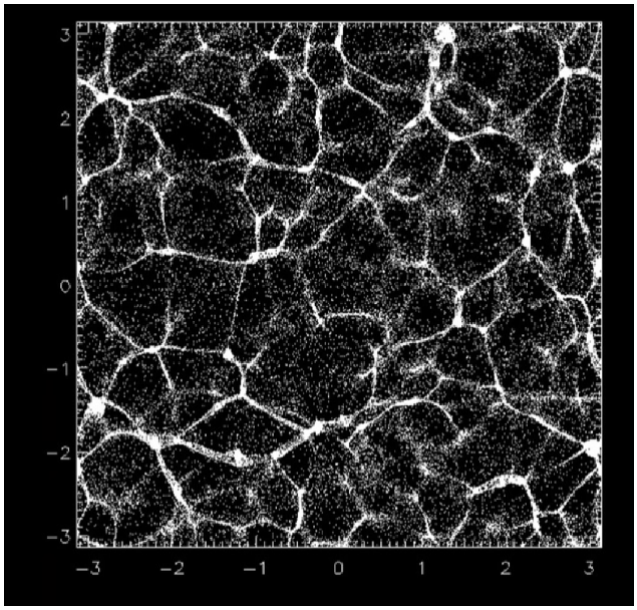

Figure 8: First Pencil Code dark matter trial during the coding session on the Monday of week 2.

based on Wlad's earlier sample called `2d-tests/selfgrav-shearwave-dust-par`. All that needed to be done was to include a scale factor $a(t)$ on the rhs of the Poisson equation, expressed in *supercomoving coordinates* as indicated by tildes, which now reads[5]

$$\widetilde{\nabla}^2\widetilde{\phi} = 6a\left(\frac{\widetilde{\rho}_{\rm tot}}{\rho_0} - 1\right). \quad (1)$$

The result shown in Figure 8 looks impressive for a first trial.

8 Relevant job opportunities

8.1 PhD at Hamburg Observatory

We are looking for a PhD student to work on Pencil Code star-in-a-box simulations of intermediate mass stars in the pre-main sequence phase as a part of the DFG project "Fossil fields in Ap/Bp stars: an origin from star formation and protostellar evolution" in Hamburg Observatory. The project is an international collaboration involving Robi Banerjee (Hamburg Observatory), Dominik Schleicher (Sapienza University, Rome), Petri Käpylä (Institute for Solar Physics, Freiburg), and Daniele Galli (INAF Astrophysical Observatory of Florence). The PhD candidate will work in Hamburg where Prof. Banerjee will act as the formal supervisor, but the project entails collaboration with and visits to all of the participating institutes. We ask the senior members of the Pencil Code community to distribute the news to potentially interested students especially if they already have experience in using the code. Interested students should contact Petri Käpylä (`<pkapyla@leibniz-kis.de>`) for more details.

[5]see Martel, H. & Shapiro, P. R., "A convenient set of comoving cosmological variables and their application," *Month. Not. Roy. Astron. Soc.* **297**, 467–485 (1998).

8.2 COSMOMAG postdoc positions

Four postdoc positions are available, with deadline 20 January: two at CERN and LPENS, and two at Nordita:

1. `https://inspirehep.net/jobs/3094257`
2. `https://inspirehep.net/jobs/3094314`

We would be grateful if you could forward this announcement to potentially interested candidates. For details regarding the COSMOMAG project, see `https://cosmosimfrazza.eu/cosmomag`.

9 Announcements

During 12–23 January 2026, there will be the Nordita Winter School 2026, which is on Cosmological Magnetic Fields: Generation, Observation, and Modeling; see `https://indico.fysik.su.se/event/8554/`. While the deadline has now passed, there is still the possibility of online participation via zoom on `https://stockholmuniversity.zoom.us/j/62081336983`. The second week will feature lectures and exercises involving the Pencil Code. Teaching material is currently being assembled under the link `https://norlx65.nordita.org/~brandenb/teach/PencilCode/COSMOMAG2026/`.

10 Papers since Sept 2025

As usual, we look here at new papers that make use of the Pencil Code. Since the last newsletter of August 2025, six new papers have appeared on the arXiv, plus nine others, some of which had been just preprints and now have been published with a journal reference on ADS. We list both here, altogether 15. A browsable ADS list of all Pencil Code papers can be found on: `https://ui.adsabs.harvard.edu/public-libraries/`

iGR7N57OSy6AlhDMQRTe_A. If something is missing in those entries, you can also include it yourself in: https://github.com/pencil-code/pencil-code/blob/master/doc/citations/ref.bib, or otherwise just email brandenb@nordita.org. A compiled version of this file is available as https://github.com/pencil-code/website/blob/master/doc/citations.pdf, where we also list a total of now 136 code comparison papers in the last section "Code comparison & reference". Those are not included in our list below, nor among the now total number of 764 research papers that use the Pencil Code.

References

Baehr, H., Rice, K., Yang, C.C. and Hall, C., On the Gravitational Collapse of Small Dust Grains in Self-gravitating Disk Structures. *Astrophys. J.*, 2025, **995**, 39.

Brandenburg, A., Ghosh, O., Vazza, F. and Neronov, A., Magnetic field spreading from stellar and galactic dynamos into the exterior. *arXiv e-prints*, 2025a, arXiv:2512.17770.

Brandenburg, A., Larsson, G., Del Sordo, F. and Käpylä, P.J., Magnetorotational Instability in a Solar Near-surface Mean-field Dynamo. *Astrophys. J.*, 2025b, **993**, 22.

Brandenburg, A. and Ntormousi, E., Magnetic Field Amplification during a Turbulent Collapse. *Astrophys. J.*, 2025, **990**, 223.

Elias-López, A., Numerical simulations of astrophysical dynamos and applications to giant planets. *arXiv e-prints*, 2025, arXiv:2512.09725.

Gent, F.A., Mac Low, M.M., Korpi-Lagg, M.J., Puro, T. and Reinhardt, M., Asymptotic behaviour of galactic small-scale dynamos at modest magnetic Prandtl number. *arXiv e-prints*, 2025, arXiv:2512.17885.

Ghosh, O., Brandenburg, A., Caprini, C., Neronov, A. and Vazza, F., Can galactic magnetic fields diffuse into the voids?. *arXiv e-prints*, 2025, arXiv:2510.26918.

Gurgenidze, M., Long, A.J., Roper Pol, A., Brandenburg, A. and Kahniashvili, T., Primordial magnetic field from chiral plasma instability with sourcing. *arXiv e-prints*, 2025, arXiv:2512.09177.

Lipatnikov, A.N. and Sabelnikov, V.A., A numerical study of velocity-pressure gradient and pressure-dilatation terms in transport equations for subfilter turbulent kinetic energy in premixed flames. *Phys. Fluids*, 2025, **37**, 105163.

Mtchedlidze, S., Vazza, F., Du, X., Carretti, E., Stuardi, C. and O'Sullivan, S.P., Simulated Rotation Measure Sky from Primordial Magnetic Fields. *arXiv e-prints*, 2025, arXiv:2511.19508.

Qazi, Y., Shukurov, A., Tharakkal, D. and Gent, F.A., Mathematical Aspects of Geophysical and Astrophysical Fluid Dynamics: Magnetic buoyancy instability in galaxies. *arXiv e-prints*, 2025, arXiv:2509.06173.

Singh, T., Sankar Maity, S., Chatterjee, P. and Pogorelov, N., Combining local and global magnetohydrodynamic simulation frameworks to understand the evolution of coronal mass ejections. *J. Astrophys. Astron.*, 2025, **46**, 87.

Skalidis, R., Tritsis, A., Beattie, J.R. and Hopkins, P.F., Residual energy in weakly compressible turbulence with a mean guide field. *arXiv e-prints*, 2025, arXiv:2512.11973.

Tschernitz, J. and Bourdin, P.A., Kinematic viscosity in solar convection simulations. *Astron. Astrophys.*, 2025, **701**, A185.

Zhang, J. and Brandenburg, A., Resistive Scaling in the Magnetic Helicity-Driven Inverse Cascade. *arXiv e-prints*, 2025, arXiv:2509.21141.

This Pencil Code Newsletter was edited by Axel Brandenburg <brandenb@nordita.org>, Nordita, KTH Royal Institute of Technology and Stockholm University, SE-10691 Stockholm, Sweden; and Matthias Rheinhardt <matthias.rheinhardt@aalto.fi>, Department of Computer Science, Aalto University, PO Box 15400, FI-00076 Aalto, Finland. See http://www.nordita.org/~brandenb/pencil-code/newsletter or https://github.com/pencil-code/website/tree/master/NewsLetters for the online version as well as back issues.