The Pencil Code Newsletter

Issue 2024/2

October 7, 2024, Revision: 1.25

Contents

1 PCUM 2024 in Barcelona

The Pencil Code User Meeting (PCUM) of 2024 was well received by all participants. We had a number of newcomers who have heard about the code, but haven't actually used it yet. They all acknowledged the personal contacts that developed. This provided help that could not be replaced by any zoom meeting.

An important highlight was the tutorial led by Jenny Schober (`https://jennifer-schober.com/`). She provided hands-on experience with the code and its basics; see `https://jennifer-schober.com/teaching/pcum2024/` for her material.

1.1 15 on-site participants

When people introduced themselves on Monday morning (Figure 1), they also indicated since which year they started using the code. Out of the 15 on-site participants, there were 5 newcomers to the code. Most of the other 10 participants started using the code in the last 10 years, and a few others even well before then; see Figure 2 for a histogram. There were also some who have been exposed to the code since earlier years, but didn't really work with it until opportunities arose to get help from others either during the tutorials or during some of the coding sessions.

Figure 1: Monday morning: introduction.

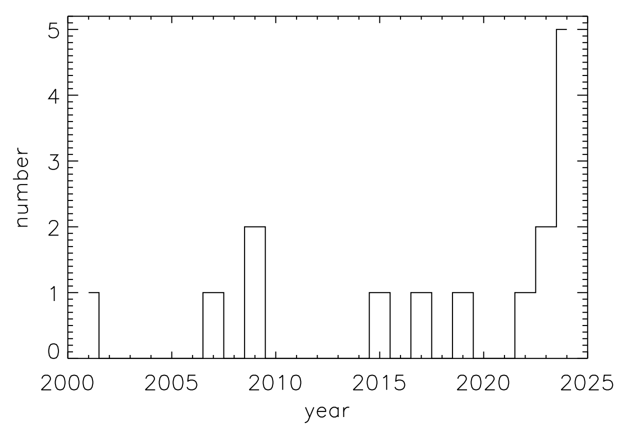

Figure 2: Number of users versus the year when they started using the code.

1.2 Mac problems

Macs used to be the system of choice for many users. This has changed and new upgrades often result in trouble. These types of problems are usually unheard of for Linux users. Fortunately, we had some experts.

Below, we reproduce two recent posts that are intended to help with MacOS problems. The instructions by *Deepen Garg* (`https://www.dgarg.com/`) are now posted on `http://pencil-code.nordita.org/doc/MacOS/Pencil_Code_MacOS_Writeup.pdf`. This text is also listed on `http://pencil-code.nordita.org/doc.php` and reproduced below in Sec. 1.3.

On September 30, Clara Dehman posted the following on our pencil-code-discuss list (`https://groups.google.com/g/pencil-code-discuss/c/iZd8pG5B_DM/m/YEWC6bkUAwAJ`):

I'm reaching out regarding a solution I found for getting MPI to work properly on my MacBook, which might be helpful for anyone facing similar issues.

After updating my MacBook to the latest MacOS (Sequoia 15.0) and Xcode (16.0), I encountered problems running the Pencil Code with multiple processors due to MPI no longer functioning. I attempted various fixes, including reverting to the previous Xcode version (15.3), which had worked before, and uninstalling and reinstalling both Xcode and Homebrew (which I used to install MPI). However, the problem persisted.

I then decided to use MacPorts instead of Homebrew. While installing MacPorts, I was prompted to update to the latest version of Xcode (16.0). Once I did that, installing MPI via MacPorts successfully resolved the issue.

Figure 3: Tuesday evening dinner: a great treat!

1.3 Tips for MacOS installation

By Deepen Garg (`https://www.dgarg.com/`)

1. Download Xcode Command Line Tools, by terminal command:

   ```
   xcode-select --install
   ```

   or, if that doesn't work, by downloading from `https://developer.apple.com/download/all/`. Version 15.4 has been seen to run on Macbooks with the Apple M series chips as well as the Intel chips (you can check the chip in *About This Mac* from the Apple icon on the top left). You do not need the full Xcode, but that should work as well (Be careful with the large install size of the full Xcode). This step installs `git` and a C compiler along with some other tools. The default C compiler that ships with v15.4 is `clang` (you can check the version with the command: `clang --version`). For internal purposes, Apple, however, also "installs" another compiler called `gcc` which is running `clang` under the hood. You can check this with: `gcc --version`, which should show the same result as the clang version.

2. Install Homebrew by running the following command in the terminal (check with `https://brew.sh/` for the latest installation steps):

   ```
   /bin/bash -c "$(curl -fsSL
       https://raw.githubusercontent.com/
       Homebrew/install/HEAD/install.sh)"
   ```

   Homebrew is a package manager for MacOS (and Linux), which we are going to use to obtain the various compilers required in the following steps.

3. Install C and Fortran compilers using the command `brew install gcc`. The gcc package contains various GNU compilers. The ones we need are `gfortran` for Fortran, and `gcc` for C (Be careful not to confuse the gcc package with the gcc compiler). However, the gcc compiler installed by brew is actually not called `gcc` because of the conflict with `gcc` distributed by Apple. They are usually called as `gcc-xx` where xx represents the major version number. Checking the `gfortran` version number (`gfortran --version`) should show Homebrew as the source and the major version number. Then, running `gcc-xx --version` should also show the same version number as `gfortran` and Homebrew as the source.

4. Install the openmpi compiler required for parallelization by running `brew install open-mpi`

5. If you are lucky, this setup should work as-is. However, there have been multiple reports of it not

working, and the culprit seems to be the fact that Pencil Code (PC) uses the gcc compiler, which is `clang` under the hood for Macs, and the gfortran compiler, which is installed from Homebrew, and these two compilers from different sources are not compatible sometimes. So we need to make PC use the gcc-xx compiler that was also sourced from Homebrew. There are a few solutions depending upon the needs of the user:

(a) Open the `GNU-gcc.conf` file[1] in any text editor and change the line `CC = gcc` to `CC = gcc-xx` where xx is the version number you found in step 3 above. This will change the compiler PC is using, and everything should work fine again now. When you pull a newer version of PC, there is a small chance that this file could be overwritten, in which case you would need to change this again.

(b) Check the directory listed when running `which gfortran` and make sure that this directory comes before the directory listed by `which gcc` when you run `echo $PATH`. If required, change the `$PATH` variable to make sure this is the case. Then, run this command:

```
cd `dirname \`which gfortran\`` &&
ln -s `ls -1 --color=never
   gcc-* | grep -Ev --color=never
   "\-[^\-\s]+\-" | tail -1` gcc
```

This command navigates to the directory where `gfortran` is installed (basically the Homebrew directory) and creates a soft link called `gcc` for the last version of gcc it can find. Alternatively, you can do it manually by navigating to the directory listed under `which gfortran` and then running `ln -s gcc-xx gcc`. Restart your shell by either closing and opening the terminal or running `exec zsh` (or `exec bash` if you are using bash), and then run `gcc --version`. It should now show Homebrew as the source.

Run PC again; this should resolve the compilation issues.

1.4 Dinners and lunches

Dinners and lunches provided great opportunities for discussing problems of scientific interest; see Figures 3 and 4. In the rest of this newsletter, we elaborate on some of the points that emerged during the meeting. When the Friday group picture was taken (Figure 5), some people had left already, but others continued.

Figure 4: Thursday luncheon: all problems solved.

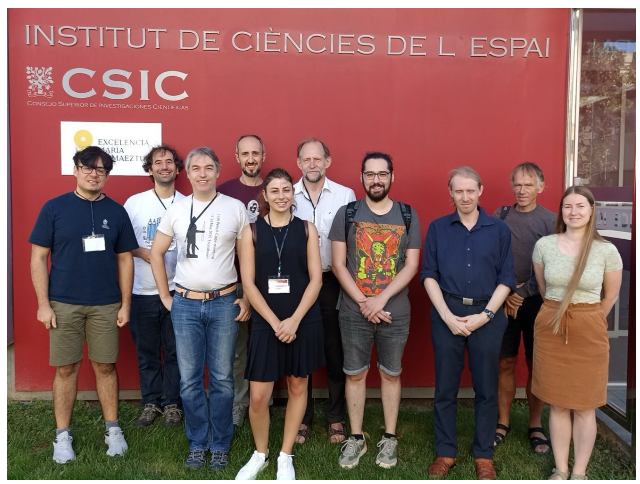

Figure 5: Friday group picture: some have left.

2 Tutorials from "Bernoulli"

On an earlier occasion, at the Bernoulli Center at EPFL in Lausanne (`https://indico.cern.ch/event/1334236/overview`), Jenny Schober and Alberto Roper Pol presented a tutorial with hands-on exercises using Pencil Code. It can be accessed via Jenny's website [2], where it is protected under the password BernoulliMHD2024.

[1] `$PENCIL_HOME/config/compilers/separate/GNU-gcc.conf`

[2] `https://jennifer-schober.com/teaching/numerical-simulations-of-magnetohydrodynamics-in-the-early-universe`

This course attracted a number of interested users, in particular those interested in MHD and GWs in the early Universe. You can find a few pictures from the course and the workshop attached!

Figure 6: Pencil Code tutorial at the Bernoulli Center at EPFL in Lausanne.

3 GPU readiness

During the PCUM24, Matthias Rheinhardt explained how to include the Astaroth libraries in the Pencil Code so as to gain significant speedup though GPU accelerated computing. His presentation is now also listed on `http://pencil-code.nordita.org/doc.php` and available on `http://pencil-code.nordita.org/doc/GPUs/Presentation3.pdf`.

4 Dispersive and diffusive errors

In simulations, dispersive errors result from truncation errors in the spatial discretization, while diffusive errors result from truncation errors in the temporal discretization.

When someone asks about numerical diffusion of a code, one does usually not mean the diffusive error of the discretization scheme, but a numerical device that acts like explicit diffusion in places where it is "needed", e.g., "slope-limited diffusion". Below, we review dispersive and diffusive errors.

4.1 Spatial discretization

The spatial discretization with the 6th order scheme yields an error that is proportional to δx^6, where δx is the mesh spacing. Specifically, we have

$$(f')_{\rm num}^{\rm 6th} = f' + 7\times10^{-3}\delta x^6\, f^{\rm (vii)}; \qquad (1)$$

see Eq. (9.42) of Brandenburg (2003)[3]. For an advection equation of the form

$$\dot{f} = -Uf', \qquad (2)$$

where $U=$ const is the advection speed, and $f\propto e^{{\rm i}(kx-\omega t)}$, we obtain for the dispersion relation

$$-{\rm i}\omega = -{\rm i}kU\left[1-7\times10^{-3}(k\delta x)^6+\ldots\right], \qquad (3)$$

so ω is real, the error is dispersive and there is no diffusive error.

A dispersive error corresponds to a phase error, because the phase speed of a wave, ω/k, amounts no longer to the actual value of $(kU)/k=U$, but to a smaller value, $(k_{\rm eff}U)/k$, where $k_{\rm eff}$ is the effective wavenumber, which is a function of k that is no longer linear at large k. The highest wavenumber is the Nyquist frequency, $k_{\rm Ny}=\pi/\delta x$, which corresponds to a zic-zac function where a wave period is described by just two points.

At $k=k_{\rm Ny}$, the value of $k_{\rm eff}$ vanishes for centered differences, so the error is infinite. Computing $k_{\rm eff}(k)$ is an exercise in the Pencil Code course; see `http://norlx65.nordita.org/~brandenb/teach/PencilCode/keff.html`. We see from Eq. (3) that the leading correction is proportional to $(k\delta x)^6$, which is almost 10^3 at the Nyquist wavenumber, so $7\times10^{-3}(k\delta x)^6$ exceeds unity. This shows that there are higher order terms that become important at and near the Nyquist wavenumber.

4.2 Diffusive error

A diffusive error results from the temporal error of the scheme in an advection equation such as Eq. (2). Analogous to Eq. (1), the temporal error in the Pencil Code is in our standard scheme

$$(\dot{f})_{\rm num}^{\rm 3rd} = \dot{f} + \epsilon^{\rm (3rd)}\delta t^3\,{\rm d}^4 f/{\rm d}t^4. \qquad (4)$$

where $\epsilon^{\rm (3rd)}=0.042$ is a non-dimensional coefficient quantifying the error of the third order timestepping scheme. To estimate ${\rm d}^4 f/{\rm d}t^4$, we differentiate Eq. (2) three times (using $U=$ const). After the first differentiation, we find $\ddot{f}=-U\dot{f}'=+U^2f''$, so we pick up a diffusion-like term. Differentiating another two times, we obtain ${\rm d}^4 f/{\rm d}t^4=U^4{\rm d}^4 f/{\rm d}x^4$, corresponding to fourth order hyperdiffusion. Inserting this into Eq. (4) yields an advection-hyperdiffusion equation,

$$\dot{f} = -Uf' - \kappa_2\,{\rm d}^4 f/{\rm d}x^4, \qquad (5)$$

[3] `http://norlx65.nordita.org/~brandenb/Own_Papers/B/2003/Bran_comp03.pdf`

with a hyperdiffusion coefficient $\kappa_2 = \epsilon^{(3\text{rd})}\delta t^3 U^4$. Already during the PCUM in 2009 in Heidelberg, it was demonstrated that the advection of the sine wave with wavenumber $k = 1$ yields diffusive behavior with an exponential decay proportional to $e^{-\lambda t}$ at the rate $\lambda = 1.06 \times 10^{-6}$ for 128 mesh points when $U = 1$. This agrees with the value of $\kappa_2 k^4$ obtained from the calculation above.

However, when applying upwinding, both discretizations contribute to both errors, for first order in t and x:

$$\begin{aligned} \dot{f} + Uf' = & \frac{U\delta x}{2}(1 - \mathrm{Cou})f'' \\ & + \frac{U(\delta x)^2}{6}(3\mathrm{Cou} - 2\mathrm{Cou}^2 - 1)f''' \end{aligned} \tag{6}$$

with the Courant number $\mathrm{Cou} = U\delta t/\delta x$.

In general, upwinding can be described as as the sum of a centered difference scheme and a centered hyperdiffusive term; see Appendix B of Dobler et al. (2006); see `https://ui.adsabs.harvard.edu/abs/2006ApJ...638..336D/abstract`. For a sixth-order scheme, this hyperdiffusivity is $|U|\delta x^5/60$.

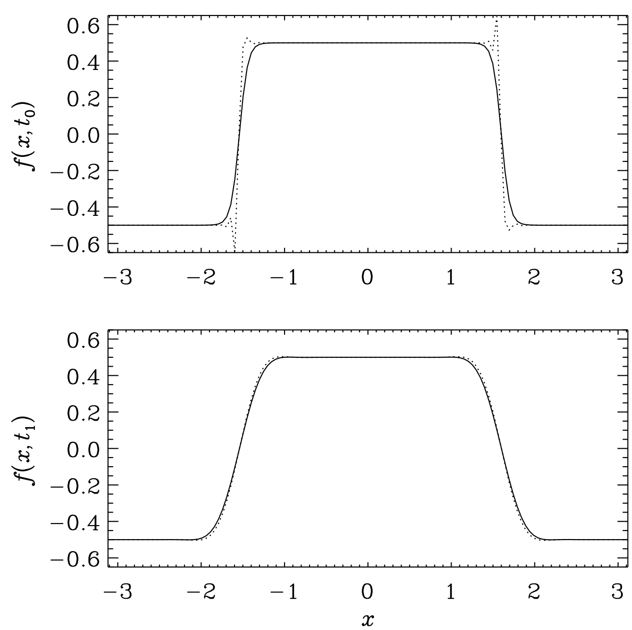

Figure 7: Demonstration of start-up errors using $w = 0.1$ (solid line) and $w = 0.01$ (dotted line) one timestep after the initial time t_0 (top) and after 5 "revolutions" through the periodic mesh at $t_1 = 10\pi$ (bottom).

4.3 Artificial viscosity in 2009

Those who attended the PCUM in 2009 and the years before will remember Mikaela Sundberg (Professor of Sociology at Stockholm University; see `https://www.su.se/english/profiles/msund-1.183862`), who studied the sociology of us during the meetings. But you may not know that she has written several related papers. Her work may attract new interest in view of related discussions during the PCUM in Barcelona.

tions with the code. With a critical tone of voice, one of the founders of the simulation code asked the audience if anyone was using "artificial viscosity." One of the other developers answered that he was and he defended it by saying that it was needed to solve "real physical problems," as opposed to solving problems under "very idealized conditions," which is what the developer considers the founder to be doing in his simulations. After some other users had reported on their inclusion of such terms too, the founder of the simulation code concluded the discussion by recommending that users be cautious with "unphysical things" in the code. These terms are certainly used by simulationists for pragmatic reasons, but they still cause principle debates.

Figure 8: Excerpt from Sundberg's 2012 paper[4].

Her impressions from a discussion of numerical diffusion are reflected by the excerpt in Figure 8 . As back then, the community still seems divided.

4.4 Start-up errors

Start-up errors result when the initial condition is too sharp such that its derivatives cannot be represented accurately by the scheme. In the Pencil Code, advection tests can be done like in `http://norlx65.nordita.org/~brandenb/teach/PencilCode/advection.html`. A hatwave of the form

$$f = \tfrac{1}{2}[1 + \tanh(w^{-1}\cos kx)] \tag{7}$$

is invoked through

```
&pscalar_init_pars
initlncc='hatwave-x',ampllncc=1e-0,widthcc=.10
```

In Figure 7, we see that $w = 0.01$ is clearly too small and yields after one timestep sharp spikes in the tail of the flanks. After five revolutions through the periodic mesh, i.e., at a time of $t = 10\pi$ (for $k = U = 1$), the profiles for $w = 0.01$ and $w = 0.1$ look almost the same, however. In both cases, we used for the passive scalar diffusivity the value $\kappa = 5 \times 10^{-4}$. On our mesh with $\delta x = 2\pi/128 \approx 0.05$, this corresponds to $\kappa \approx 0.01\,U\delta x$, which was also the value used in Eq. (9.42) of Brandenburg (2003) for the sixth-order scheme.

[4]Sundberg 2012, *Science, Technology, & Human Values*, **37**, 64–87 (`https://doi.org/10.1177/0162243910385417`)

5 Random tips

5.1 Check-in notifications

Many of us receive check-in notifications after each check-in. This can be a lot of mail, but normally you would delete it anyway after having looked at it. Being aware of the check-in activity can be useful for anybody doing any changes to the code. You might, e.g., think that a particular `special` module of yours should only be of your concern, but this is not true. Somebody else might need to make changes to it because of some compatibility issues, for example.

Check-in notifications are published on `https://groups.google.com/g/pencil-code-commits` and one can subscribe to this list. Please email to brandenb@nordita.org.

5.2 New "Documentation" material

On `http://pencil-code.nordita.org/doc.php`, several new links have appeared. One is by Deepen Garg with Instructions for MacOS installation (Sep 29, 2024). Another is the GPU acceleration with the Pencil Code using Astaroth by Matthias Rheinhardt (25 Sep 2024). Finally, we have now a link to the teaching material of Jennifer Schober from the Barcelona User Meeting, where she explains how to simulate turbulent dynamos with the Pencil Code (Sep 25, 2024).

6 Next PC User Meeting

During the PCUM in Barcelona, there was an ad hoc steering committee meeting to discuss proposals for the next meeting. The current chair of the Pencil Code Steering Committee will ask the proposers for more detailed proposals. The minutes on the meeting will be published on `http://norlx65.nordita.org/~brandenb/pencil-code/PCSC/minutes/`.

7 Upcoming meetings

Several meetings may be of interest to the Pencil Code community. During December 2–4, there is the 19th MHD days in Potsdam; see `https://meetings.aip.de/event/1/`. Furthermore, there will be two programs at Nordita during 2025: the one on "Axions in Stockholm" during June 23 – July 11 (`https://indico.fysik.su.se/event/8808/`) with Pencil Code user Oksana Iarygina as an organizer and that on "Numerical simulations of early Universe sources of gravitational waves" during July 28 – August 15 (`https://indico.fysik.su.se/event/8805/`) with Pencil Code user Alberto Roper Pol as an organizer Both sites are still restricted though.

8 Papers since May 2024

As usual, we look here at new papers that make use of the Pencil Code. Since the last newsletter of May, 3rd, 11 new papers have appeared on the arXiv, plus 9 others, some of which had been just preprints and have now been published in a journal. We list both here, altogether 20. A browsable ADS list of all Pencil Code papers can be found on: `https://ui.adsabs.harvard.edu/public-libraries/iGR7N570Sy6AlhDMQRTe_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 119 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 722 research papers that use the Pencil Code.

References

Brandenburg, A., Neronov, A. and Vazza, F., Resistively controlled primordial magnetic turbulence decay. *Astron. Astrophys.*, 2024a, **687**, A186.

Brandenburg, A. and Banerjee, A., Turbulent magnetic decay controlled by two conserved quantities. *arXiv e-prints*, 2024, arXiv:2406.11798.

Brandenburg, A., Iarygina, O., Sfakianakis, E.I. and Sharma, R., Magnetogenesis from axion-SU(2) inflation. *arXiv e-prints*, 2024b, arXiv:2408.17413.

Dehman, C. and Brandenburg, A., Reality of Inverse Cascading in Neutron Star Crusts. *arXiv e-prints*, 2024, arXiv:2408.08819.

Hidalgo, J.P., Käpylä, P.J., Schleicher, D.R.G., Ortiz-Rodríguez, C.A. and Navarrete, F.H., Magnetohydrodynamic simulations of A-type stars: Long-term evolution of core dynamo cycles. *arXiv e-prints*, 2024, arXiv:2409.18066.

Kishore, G., Singh, N.K., Käpylä, P. and Roth, M., Strengthening of the f mode due to subsurface magnetic fields in simulations of convection. *arXiv e-prints*, 2024, arXiv:2409.14840.

Korpi-Lagg, M.J., Mac Low, M.M. and Gent, F.A., Computational approaches to modeling dynamos in galaxies. *Liv. Rev. Comp. Astrophys.*, 2024, **10**, 3.

Lipatnikov, A.N., Towards large eddy simulations of premixed turbulent flames without a combustion model. *arXiv e-prints*, 2024, arXiv:2408.07668.

Lyra, W., Yang, C.C., Simon, J.B., Umurhan, O.M. and Youdin, A.N., Rapid Protoplanet Formation in Vortices: Three-dimensional Local Simulations with Self-gravity. *Astrophys. J. Lett.*, 2024, **970**, L19.

Maity, S.S., Sarkar, R., Chatterjee, P. and Srivastava, N., Changes in Photospheric Lorentz Force in Eruptive and Confined Solar Flares. *Astrophys. J.*, 2024, **962**, 86.

Mtchedlidze, S., Domínguez-Fernández, P., Du, X., Carretti, E., Vazza, F., O'Sullivan, S.P., Brandenburg, A. and Kahniashvili, T., Intergalactic medium rotation measure of primordial magnetic fields. *arXiv e-prints*, 2024, arXiv:2406.16230.

Saieed, A. and Hickey, J.P., Viscosity-modulated clustering of heated bidispersed particles in a turbulent gas. *J. Fluid Mech.*, 2024, **979**, A46.

Sankar Maity, S., Chatterjee, P., Sarkar, R. and Mytheen, I.S., Evolution of reconnection flux during eruption of magnetic flux ropes. *arXiv e-prints*, 2024, arXiv:2407.18188.

Schober, J., Rogachevskii, I. and Brandenburg, A., Efficiency of dynamos from an autonomous generation of chiral asymmetry. *Phys. Rev. D*, 2024, **110**, 043515.

Sengupta, D., Cuzzi, J.N., Umurhan, O.M. and Lyra, W., Length and Velocity Scales in Protoplanetary Disk Turbulence. *Astrophys. J.*, 2024, **966**, 90.

Vashishth, V., Hysteresis Near the Transition of the Large-Scale Dynamo in the Presence of the Small-Scale Dynamo. *Sol. Phys.*, 2024, **299**, 115.

Vemareddy, P., Simulating the formation and eruption of flux rope by magneto-friction model driven by time-dependent electric fields. *arXiv e-prints*, 2024, arXiv:2409.14045.

Wang, K., Wang, H., Zheng, J., Luo, K. and Fan, J., Particle-resolved numerical simulations of char particle combustion in isotropic turbulence. *Proc. Comb. Inst.*, 2024, **40**, 105315.

Warnecke, J., Korpi-Lagg, M.J., Rheinhard, M., Viviani, M. and Prabhu, A., Small-scale and large-scale dynamos in global convection simulations of solar-like stars. *arXiv e-prints*, 2024, arXiv:2406.08967.

Yuvraj, Im, H.G. and Chaudhuri, S., How "mixing" affects propagation and structure of intensely turbulent, lean, hydrogen-air premixed flames. *arXiv e-prints*, 2024, arXiv:2405.17197.

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. For a timeline off the beach of Bogatell on the last day of the PCUM24 in Barcelona; see Figure 9.

Figure 9: Timeline off the beach of Bogatell on the last day of the PCUM24 in Barcelona.

.