

# GDTk CHANGELOG 2022-q4

*A newsletter for the GDTk Community*

23 October 2022

Welcome to the inaugural newsletter for the GDTk community! There's been a lot happening in the past few months and it seemed the timing was right to record and broadcast those happenings in an informal newsletter. Our intent is to release on a quarterly basis. The newsletter will be principally written and compiled by the developers, but we will also readily welcome contributions from the community. Without further ado, read on to find out what's been happening lately amongst the group of users.

Since this is our first newsletter, I'll take the liberty of a loose definition for the period of a quarter. Let's share some of the activities going back half a year or so. We've had some exchanges and visits, some serious paper writing and some even more serious code writing resulting in a program called Chicken.

## *Eilmer in the archive*

The dev team has recently published a paper in the archival literature on the D language version of Eilmer (colloquially called Eilmer 4). The paper appears in *Computer Physics Communications*.<sup>1</sup> For a limited time, we've been given a link to share the work freely.<sup>2</sup> After that time, you can likely get to the paper through your institution's access to Elsevier journals.

Eilmer is in the archive in more ways than one now. *Computer Physics Communications* are serious about the reproducibility of results derived from simulation programs, so they archive the source code as well. As part of the publication process, Nick Gibbons had to ensure that the code built, installed and ran as advertised. This was done in a container environment that the article reviewers were able to inspect. The source code now has its own DOI

<https://doi.org/10.17632/gy2ds2fyxm.1>

and a location on Code Ocean

<https://codeocean.com/capsule/7226427>.

The paper covers the formulation and numerics in Eilmer, our development process, and gives several examples of applications. I wrote more about the story behind the paper on our GDTk blog.

<https://gdtk.uqcloud.net/blog/the-paper-on-eilmer-v4-is-out.-go-read-it>

## *Hacking at Chickens*

Putting aside the clickbait-inspired gore of the title, what we're talking about here is writing code (hacking) for a new flow solver called Chicken.

We're really excited that the team has been selected to attend a GPU Hackathon in Canberra, hosted by National Compute Infrastructure Australia. The team will get to work with NVIDIA



Figure 1: The Eilmer paper is available in CPC volume 282.

<sup>1</sup> N. N. Gibbons, K. A. Damm, P. A. Jacobs, and R. J. Gollan. Eilmer: an open-source multi-physics hypersonic flow solver. *Computer Physics Communications*, 282(108551), 2023. DOI: 10.1016/j.cpc.2022.108551

<sup>2</sup> <https://authors.elsevier.com/1fsPB20InkAiF>



Figure 2: A postcard for Easter time by Wally Fialkowska with hens pulling a cart, first sold in 1914. <https://tuckdbpostcards.org/items/35560>

employees to look at compute performance improvements and tuning on NVIDIA HPC-grade GPUs. I should explain that GPUs are graphics processing units. They have been growing in dominance in the high-performance computing world over the past decade. The current world-leading supercomputer, Frontier, set the record for exaflop compute by stitching together 37,000 GPUs (albeit using ones from rival company AMD).<sup>3</sup>

To get the most benefit out of the hackathon, the team needs a fairly complete compressible flow solver written for GPUs. To answer that call, Peter Jacobs bought himself a kiddies NVIDIA card<sup>4</sup> to start building yet another CFD code. That code is called *Chicken*. The name *Chicken* was chosen in part to respond to an old debate in supercomputing, and that is, whether one would prefer few but powerful compute units or many small processors to get the work done. The debate has ended: many small processors has won in the supercomputing space. The actual words that inspired the name came from John Rollwagen<sup>5</sup> who was CEO of Cray Research. At a CTAC meeting in the early 1990s, Peter Jacobs attended a talk given by Rollwagen.<sup>6</sup> The opening to Rollwagen's talk proposed a hypothetical situation, paraphrased based on recollection from 30 years ago, and it went something like this...

Say that you are an early settler on the east coast of America and are about to start your journey into the wild west with all of your family and belongings in your wagon. What are you going to trust to pull that wagon: Four hefty bullocks or 1024 chickens?

So you can see that our answer, some 30 years later, is: "Chickens, please, lots of them."

For those following the Git repository, *Chicken* began life with commit e359113b:

```
commit e359113b6486589830b594d1d72492660b6a8e75
Author: Peter Jacobs <peterj@mech.uq.edu.au>
Date: Sun Sep 11 15:18:16 2022 +1000
```

The start of an experiment to write a 3D flow solver in CUDA C++.

Peter has been aided by others in the team in the form of bug hunting, code review and a general elevation of banter that arises when Peter starts a new code. Those others include Christine Mittler, Kyle Damm, Jamie Border, Rob Watt and Nick Gibbons. Unfortunately, the team to attend the hackathon is capped at 4 members. Peter, Christine, Nick and Rob will travel to Canberra for the in-person event held over 2,3,4 November.

*Chicken* has a somewhat niche application: highly-resolved calculations on geometries with simple topological arrangements. Think: direct numerical simulation (DNS) of transitional/turbulent hypersonic boundary layers. The goal is to incorporate Christine Mittler's thesis work to get the high order accuracy into the spatial reconstruction as an enabler for DNS.

We look forward to getting a report back from the team later in November.

## *Pretzels and Vegemite*

Some of our Australian team recently got the chance to sample pretzels in Germany. Meanwhile, we have had a few visitors from Germany to the group. I cannot say how well or otherwise they are taking to the Vegemite.

<sup>3</sup> <https://www.ornl.gov/news/frontier-supercomputer-debuts-world-s-fastest-breaking-exascale-barrier>

<sup>4</sup> Kiddies is not meant in a perjorative sense. Rather, the graphics cards used to power the gaming computers for kids and adults alike are perfectly suitable for scientific code development. It is a graphics card of this type that Peter bought.

<sup>5</sup> <https://www.computerhistory.org/revolution/supercomputers/10/68/273>

<sup>6</sup> The context for this bullocks or chickens analogy relates to the supercomputing scene of the early 1990s. At the time there were quite a few parallel-computer companies competing for your supercomputer dollars. Cray were making multiprocessor machines with up to 4 of their vector-processors (hefty bullocks) while companies such as MasPar and Connection Machines made a computer with many small processors (chickens). For example MasPar made a computer with 1024 processing-elements that operated on a single instruction stream. <https://www.cpushack.com/2014/09/05/maspar-massively-parallel-computers-32-cores-on-a-chip/>

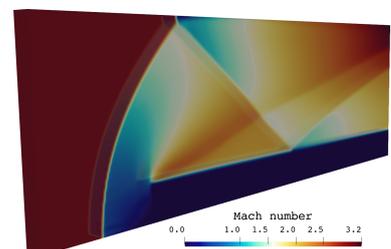


Figure 3: *Chicken* simulation of flow over a forward-facing step with incoming flow approximately Mach 3. Perspective view shows the 3D simulation of an essentially 2D flow. This simulation is 2 cells deep in z-direction.

With borders re-opening and the availability of international flights growing, the PhD students in our TUM-UQ project on DNS of hypersonic transitional flows have had the chance to complete the exchange portion of the project. From April to June, we had Giuseppe Chiapparino visit the Australian group from the Technical University of Munich. Then from June through to September, Lachlan Whyborn and Christine Mittler went to Munich for an exchange visit to TUM. We thank Christian Stemmer and hypersonic CFD team there for hosting Lachlan and Christine. While in Germany, Lachlan presented some of his recent work at the ESA organised 2nd International Conference on Flight Vehicles, Aerothermodynamics and Re-entry Missions Engineering, held in Heilbronn.

Then in September, postdoc Nick Gibbons joined in the pretzel taste testing on a trip to Belgium and Germany. In Bruges, Nick presented simulation work on scramjet combustion at the 2nd International Conference on High Speed Vehicle Science & Technology. I've heard that Nick got to catch up with long-time Eilmer user Jimmy-John Hoste over a Belgian beer or two. Following that, a different kind of high-speed vehicle took Nick to Munich to visit TUM for several days.

Later in September, we received a visit from Christian Stemmer to the group, and shortly after, Friedrich Ulrich arrived. Friedrich is also completing his exchange portion of the TUM-UQ project and will be with us until December.

## *Sneak peek: Eilmer v5*

If you are watching the repository closely, you might have noticed this commit:

```
commit 32c20b27d221849e5f100e57d2627bcea8006de2
Author: Rowan J. Gollan <r.gollan@uq.edu.au>
Date: Sat Oct 22 23:43:18 2022 +1000
```

```
lmr5: first commit in public master
```

Don't panic! v4 of Eilmer is not going away anytime soon.

This commit brings the eilmer 5 work-in-progress code into the master branch on the public-facing github. This is so that I can work more closely with the developments in master.

Eilmer v5 is a change at the interface level, but uses almost all of the core files from Eilmer v4. The migration path is that new code goes in:

```
src/lmr
```

but uses a lot of code from src/eilmer.

There is an example of the new interface in:

```
examples/lmr/2D/convex-corner
```

I (Rowan) have been hacking away over the past few months at a re-built user and I/O interface for Eilmer. There is a primary driver for this work. Work on the steady-state solver began around Easter of 2016. It has matured considerably over that time thanks predominantly to Kyle Damm's efforts. We think it's mature enough now to include in the tagged releases. However, we also needed to settle



Figure 4: Pretzels or vegemite? Why not have both!

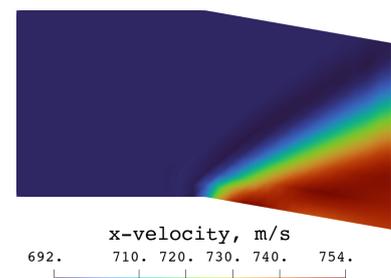


Figure 5: The very first simulation with Eilmer 5: an expansion fan (under-resolved). If you think it looks simple, you're right. I wanted to avoid any shocks in the flow field. The good news is that this simulation converges with a relative residual drop of 14 orders of magnitude in just 32 steps of Kyle's Jacobian-free Newton Krylov implementation.

on how the steady-state mode would work. In the early days, the steady-state solver got bootstrapped from the transient solver and re-used the infrastructure of user input, I/O and post-processing. It's time now though to build an interface that is most appropriate for the steady-state solver for the foreseeable long term.

The jump from Eilmer v4 to Eilmer v5 is *by no means as large* as when we went from v3 to v4. Eilmer v5 retains all of the core code from Eilmer v4. The changes are what I'm describing as superficial. Superficial in that the algorithms and functionality is unchanged. What will change is the command-line interaction, some parts of inputs (particularly associated with steady-state mode configuration), the organisation of files on disk and aspects of the post-processing. I call this the user interface. From my developer's perspective, that is superficial. A user might argue that it's anything but superficial if they have to change their input scripts. Let me allay your concerns, hopefully.

We have not made these changes without some considerable deliberation.<sup>7</sup> We believe these changes help the user in several ways. The input names are clearer and the behaviour and layout of the I/O makes more sense for running in a steady-state mode. In short, it is easier to describe how to use the new code for steady-state simulations, and easier to describe usually translates to shorter documentation and fewer misconceptions.

I have been rather abstract in describing the changes in Eilmer 5 and that's intentional. The code is not ready for general use, but I wanted to share the news here in case you spot it in the repository and wonder what it's about. I also want to give the warning that it is not ready for prime time.

#### HIC SUNT DRACONES

Our eventual goal is that all of Eilmer v4 functionality will be available in Eilmer v5. In the initial phase, Eilmer v5 will focus on the steady-state solver mode.

Finally, when Eilmer v5 is ready for mainstream use, it will be an opt-in kind of deal. Eilmer v4 will not go away until the maintenance burden on the interface overwhelms the hours we have in the day to work on it. In any case, I think there will be a low barrier to switching to Eilmer v5 when the time is right.

### What's coming up

Here are some things to look out for over the summer.

- Lachlan Whyborn and Robert Watt will each present papers at the Australasian Fluid Mechanics Conference in Sydney in the first week of December 2022.
- Kyle Damm, Jamie Border and Lachlan Whyborn will each present work at the AIAA SciTech conference in Washington DC in January 2023.
- Eilmer meet-ups: The October meet-up will be 10am (BNE time) on Thursday 27 October, with guest speaker Lachlan Whyborn. The November meet-up will be the final one for this year. It will be held on 25 November.
- GDTk Summer of Code: while not as grand as the Google summer of code, I expect the team will be able to ramp up some development activity once research and grant writing season begins.

<sup>7</sup> Although I did ambush the dev team with a first demo of Eilmer v5. I was hacking away on some refactoring work on the Exhibition holiday in August and got to the point that the only sensible way forward was to start a new directory and call it Eilmer v5.



Figure 6: The inscription of HIC SUNT DRACONES was placed on the Hunt-Lenox globe to warn about unknown territory. Similarly, Eilmer v5 should be considered unknown territory for a while. The development pace will be fast and the changes to interface frequent.