Computational Geometry 24w5229

Alexander Kasprzyk (Nottingham University),
Alessio Corti (Imperial College London),*
Elana Kalashnikov (University of Waterloo),
Marta Panizzut (UiT, Tromsø),
Emre Can Sertöz (Leiden University)

June 23–June 28, 2024

1 Executive summary

1.1 Background

The focus of the meeting was the use of the computer in research in algebraic and arithmetic geometry, on a large scale.

There is no question that the use of computers has historically played a pivotal role in research in arithmetic geometry. For example, the Birch and Swinnerton-Dyer conjecture was made on the basis of pioneering computer experiments by Swinnerton-Dyer in the early 1960s on one of the first computers, located in the basement of the DPMMS in Cambridge in the UK. Today, many algebraic and arithmetic geometers engage in daily conversation with computational algebra systems (CAS) to aid their research in many essential ways.

(At least many of our colleagues do. This is necessarily anecdotal because we as a community do not feel obligated to formally acknowledge the essential work that is done by those who develop computational algebra software.)

This is not what the meeting was about: our focus was on the use of computer resource *on a large scale*. The use of High Performance Clusters (HPC) in algebraic and arithmetic geometry, and in neighbouring mathematical fields, is a relatively new phenomenon. This is for several reasons, on which I elaborate below. There are, and there have been, a few teams in the world that are attempting it. In my view, it has the potential to transform our fields and the way that we work, by uncovering patterns, truths, that become manifest only at a large enough scale. I am suggesting that the use of large scale computational resources in algebraic and arithmetic geometry could transform those fields in a similar way that the telescope transformed astronomy or the microscope biology. A separate consideration is that undoubtedly generative AI and LLMs will need, at some point, an ecosystem of interoperable software that contains the Lean proof assistant and CAS and that runs on HPC.

Those of us who have attempted, and are attempting, to work with large computers face high barriers to entry and, when one has entered, formidable challenges at every step of the way:

*This report was written by Alessio Corti with some assistance from the other organizers and the participants. His is the "I" in the text.

(a) There is no community to help one learn the ropes. All of us end up re-inventing the wheel several times.

(b) There are no established pathways to fundraising for accessing the equipment. Research in pure mathematics has very specific needs and the existing pathways that work for applied mathematics and science don't easily work for us. Funding bodies are not alert to the fact that we, too, have a legitimate stake here.

(c) Even when one has, somehow or other, secured access to the equipment, there is the issues of software. There is very little expertise on installing and running a CAS (for example MAGMA) on an HPC. The few who have been there have had to design and implement from scratch the higher-level software needed to manage computations with CAS running on HPC, see for example `pcas.xyz`. Needless to say, there is no support available for this: you are totally on your own. This type of work needs an ecosystem of stable, properly supported and maintained, CAS – something that is very far from the current reality on the ground.

(d) When one has successfully negotiated all of the above, new problems arise: persuade the Department – or the School of Science, or the University – that we need to maintain a small cluster to run experiments in order to be able to bid for time on larger national facilities; manage a larger team with postdocs that leave for permanent academic positions but still need access to the equiment, creating tension with university IT departments, etc.

(e) Finally there is the larger question of sustainability. Projects of this type typically produce software and large databases. How to maintain the databases, how to serve them to the community in a way that the community can keep computing with them effectively? To compute effectively with a database of the order of 100 terabytes one probably needs the computer to be physically next to the data, hence, at least initially, these computation have to run on the computers of the host institution. So we get into persuading the University IT department (or somebody, anybody who can and will) to allow external actors to run jobs on the University computers.

1.2 High-level objectives

The activities of the Workshop had two high-level objectives:

(1) Fact-finding. Bring enough people together, from algebraic and arithmetic geometry, who are using or have used large computational resources for research. Find out who we are, how we are getting on, what we need.

(2) Begin to create a community where problems and practices can be shared, outward-facing and capable to offer advise to those – especially young researchers – who are on the cusp of wanting to use large computational resources for research, and able to lobby institutional actors in different national funding environments.

1.3 Program structure

The program consisted of

(1) 12 *research talks*, see brief summary below.

(2) 3 *software presentations* (by David Roe, John Voight and Andreas Paffenholz) introducing software capabilities and services: the *L-Function and Modular Form Database* (LMFDB), the *Magma Computational Algebra System* (MAGMA), and the *Database for Discrete Geometric Objects* (polyDB).

(3) An *open problem session*.

(4) Two *open discussions* – the first chaired by Frank Sottile and the second jointly by Alessio Corti (present) and John Voight (remote) – followed by an out-of-hours *committee on action points*.

The backbone of the program and the true *raison d'être* of the whole workshop were the *open discussions* and the *committee on action points*.

The entire event – excluding the out-of-hours committee on action points – was run in hybrid mode. Four out of 11 research talks, and 1 out of 3 software presentations were given by online participants. It is a known fact (at least anecdotally) that hybrid meetings are especially challenging to run successfully, more challenging in fact than purely online meetings. This workshop worked exceptionally well as a hybrid meeting. The open discussions, in particular, were extremely successful. Some of it may be down to the choices made by the people who chaired them, but I think that the key reason is that the participants *really cared* about the subject and what we were trying to do with it.

1.4 Conclusion

I think that the meeting was successful. The atmosphere was electric, and many people made passionate and generous contributions to the discussions and the action points.

There were several young researchers at the meeting, who work in computational geometry. As I say again below, "computational geometry" is not considered a field on its own, and there is a lack of recognition of the value of large-scale computations particularly in algebraic geometry. Researchers in algebraic geometry who do a lot of work that is computational/experimental rightly feel that they do not fit in the usual career path of a pure mathematician. Many of the young researchers at the workshop fit this description, and it was palpable that they felt happy to just have a chance to be with each other and support each other. I hope that the Workshop is a step in building a research community around the use of large-scale computation in algebraic and arithmetic geometry.

2 Program Highlights

2.1 Research talks

Virtually all the research talks at the workshop showed mathematical results in algebraic or arithmetic geometry, where the research made use of large-scale computing resources in an essential way. This is only a small cross-section of problems in algebraic and arithmetic geometry that can benefit from large-scale computation.

I just briefly summarize some highlights:

Frank Sottile *Frontiers of Arithmetic in Enumerative Geometry.* The talk summarizes 25 years of work on different projects in the general area of the Schubert calculus of enumerative geometry. The talk in particular describes the different computational resources that were used and the challenges met.

S. Veneziale *Machine learning detects terminal singularities*. This was an extremely well-received, highly innovative talk on some application of machine learning to explore questions in algebraic geometry, specifically in the context of the study of $\mathbb{Q}$-Fano varieties.

Homotopy Theory Three talks were based on some implementation of homotopies of continuous paths (Bruin, Brysiewicz, Ren).

Solving polynomial equations A problem of a more applied flavour is finding solutions of polynomial equations numerically. (This is after all what Algebraic Geometry is all about.) Talks by Cummings and Yahl fall under this general umbrella. Cummings spoke about finding implicitizations numerically, and the surprising simplifications that occur in the presence of a large torus symmetry. Yahl spoke about surprising phenomena with Galois groups of polynomial systems that he discovered by computer experiments.

2.2 Software demonstrations

I talk in some detail about the software demonstrations.

MAGMA [1] John Voight has just moved to the University of Sidney where he will lead the development of the MAGMA computational algebra system. MAGMA is one of a handful of leading CAS in existence today (others include Macaulay2, OSCAR, PARI/GP, and SAGE) and it may be the more easily sustainable, and hence very interesting for potential use on HPCs. The (online) talk by John Voight was an introduction on parallelization already implemented in MAGMA, leading up to our second open discussion.

LMFDB [2] In his talk on *Finite Groups and K3 surfaces in the LMFDB*, David Roe spoke about current and future offerings on finite groups in the LMFDB. The LMFDB is the probably the best example, certainly the best funded one, to date of the typical output of a large computational project: a large queryable database of mathematical objects "served" to the community. The discussion at the end of the talk raised issues of longer term financial sustainability and viability of the project, and how can the community run large computations on the data in the medium-to-longer term.

PolyDB [3] This project, led by Andreas Paffenholz, is another example of an output of a large computational project in the form of a large queryable database of mathematical objects "served" to the community. There was again a discussion of longer term financial sustainability and viability of the project, and how can the community run large computations on the data in the medium-to-longer term.

2.3 Open problem session

We had a session on open problems in computational algebraic and arithmetic geometry, led by Yue Ren. There was a good balance of problems of a more technical nature and fundamental problems of wide general interest. A short summary follows.

Y. Ren Can we deal with hierarchical data like evolutionary trees?

T. Brysiewicz Develop a technique which can make rigorous deductions from the trace tests in numerical algebraic geometry.

M. Zach Use his own "scheme" data-type in OSCAR to compute cohomology groups of coherent sheaves on projective varieties, i.e., $H^i(X, \mathcal{F})$, and apply to a hard geometric problem.

N. Bruin Homotopy continuation. He needs to track roots of polynomials approaching discriminant

F. Sottile Certification of a solution to a system of polynomial equations requires a square Jacobian. However, in pure maths, overdetermined systems are quite common. Problem: find methods to certify over-determined systems. Possible answers: (i) least squared? (This does not work: fixed points of Newton are not zeros.) Can you massage a rank condition to make it square?

E. Sertöz Numerical methods for(!) algebraic geometry. Really use numerics toward prestigious AG problems. For example, problems on linear systems on curves. Is the natural map $\mathrm{Sym}^2\mathrm{H}^0(\mathrm{C},\mathrm{L}) \to \mathrm{H}^0(\mathrm{C},\mathrm{L}^{\otimes 2})$ of maximal rank? Answer on a single (general) curve using numerical rank computations.

D. Roe Has curves (genus up to 24) represented as canonical models, and he wants to construct good planar models for the LMFDB.

[1] http://magma.maths.usyd.edu.au/magma/
[2] https://www.lmfdb.org/
[3] https://polydb.org/

2.4 Open discussions

The open discussions were popular with the participants. Minutes of the discussion were taken and shared with the participants. I limit myself to make a list of the subjects that were discussed with just a few hints at the contents of those discussions.

Community aspects The main point is that computational geometry is not considered part of the scientific computing/computational/numerical community; it is not considered a field on its own, and there is a lack of recognition of the value of large-scale computations in pure mathematics. There is a need to build a community of researchers working in this field to share knowledge and build up the credibility of the field.

As a consequence, researchers whose work is heavily computational/experimental do not fit in the usual academic career path for pure mathematicians.

Access to computing hardware The discussion centred on various issues related to accessing computing hardware. For example: there is both a need for large clusters, and smaller (easy to access) platforms where one can carry on experiments. Buying hardware is cheap, while maintaining and installing are expensive (and often institution-specific): there is usually no money from grants to pay for a research scientist to support with this. University-level clusters are often not set up for mathematicians, since they are often batch-oriented: this is potentially problematic in pure mathematics where sometimes you do not know how long a computation will take to run. Different solutions such as Google colab, or running jobs locally instead of in a university-managed cluster, have their own problems.

Data storage and access Overall, the conclusion is that purchasing equipment is rarely the problem (storage space is relatively inexpensive). The real problem is maintenance and long-term storage solutions. Often universities have centrally managed IT services or they outsource them, making it hard to receive support tailored to mathematics research. Transferring large data sets is very difficult, can software circumvent this by better communicating with data repositories? Another problem is to have large amounts of data available on the same machine (for example if there is the need for pre computations): Google colab might offer a solution to this (but it costs more). The aim is to have data that is instantly accessible by anyone, anywhere, at anytime: so querying the data here is the problem.

Software development Many CASs rely on volunteers and community: but if you want to do this on an HPC scale you need professional teams of people. CASs could be more professional and better address the communities needs with such support. At the moment, there is no avenue to secure funding via grants or otherwise to support open-source software projects. Software development needs to move towards parallelisation. However not everyone can be an expert in parallelisation: there needs to be a more transparent front-end in computer algebra systems.

Grants and Funding There is not much funding for the purpose of maintaining and providing accessibility to mathematical data and funding is not given to pay research scientists to support research groups. This is different from the applied sciences where grants are used to fund both equipment and long-term lab technicians with stable jobs.

Permanent lab-technicians To maintain expensive hardware and improve software, large-scale computation in mathematics needs the infrastructure of permanent positions for coders and technicians. Without the room to offer a stable career path for people with the right aptitude, the subject will struggle to take off. The technical work is difficult and time consuming, taking time away from what it takes to get a professorial position.

Action points These are summarized in the section on *Outcomes of the meeting*.

3 Outcomes of the meeting

In my view the most significant impact of this meeting will come not so much from individual scientific research outputs (papers, software, databases), but from higher-level outcomes.

Below I quote, from the notes that were taken during the discussion session, the *action points* that came out of those discussions. Here I briefly elaborate on two key outcomes of the workshop:

Community building We created a Zulip group to collect and share materials and resources, ask and give advice. The main goal of this group is to reduce the barriers to entry and it is a first step in building a community.

White papers Alessio Corti and Frank Sottile are writing a white paper that they hope to publish on the *Notices of the AMS* (or a similar journal) to raise awareness about large scale computations in algebraic and arithmetic geometry. A different group is working on a second, more technical, white paper.

In an ideal world we know what we want: (a) there is a community that those who want to get started can turn to for help (b) national funding bodies appreciate that we have a legitimate want to use the equipment; (c) that some CAS is supported sustainably (possibly by government funding but Simons maybe solving this problem for us with Magma); (d) that Departments and Universities also appreciate this (there is a tendency in Universities towards centralization of IT resources. This could work if they didnt charge the Department exorbitant fees. I think we prefer Departmental-level facilities, but we may have to make do with University-level facilities.) (e) that Universities accept that there are certain collaborations where facilities need to be shared with outside scientists on an informal free-of-charge basis.

A white paper is a way for us to begin to lobby for some of these things.

Action points

Community aspects:

- Build an online presence (a 'stackoverflow'-type community), bringing together current researchers working with large-scale computations in algebraic geometry and number theory.
- There needs to be an interest in training the next generation of researchers in large scale computations and experimental mathematics. It would be useful to collect examples of good problems for the new generations to work on alongside examples where the ability of running large scale computations either found counterexamples to conjectures or helped find and prove new phenomena.
- Organise conferences and events centred around software and software development to drive excitement.

Data storage and access:

- Grow mathbases.org for future researchers.

Software development

- In order to support people that develop software (especially open-source software) one might think of setting up payment options such that someone could contribute to a project using money coming from a grant.

Grants and funding

- Write a white paper (a sales pitch) about the importance of large scale computations in pure mathematics. This would be a useful document to cite in funding proposals and send to funding agencies to concisely make them aware of the value of computations. This should be thought of as a manifesto for mathematical discovery through advanced computing.
- Raise awareness among funding agencies by sitting in committees, give feedback at meetings, talk with people at grants interviews. Remember: ultimately funding agencies want to fund the next big thing!
- Potentially we could draw some connections to machine learning and AI (computer algebra systems for AI?).
- Is there a need for a new high level journal for computational mathematics?

Appendix: list of speakers and titles

The table below lists speakers and titles. The abstract and video recordings are of course available from the BIRS page of the event.[4]

N. Bruin	*Tools for rigorous computation on algebraic Riemann surfaces*
T. Brysiewicz	*Monodromy Coordinates – An iterator for large solution sets to polynomial systems*
J. Cummings	*Multigraded Implicitization*
A. Degtyarev	*At most 800 conics on a smooth quartic surface*
T. Dokchitser	*Reduction types of algebraic curves*
A. Frühbis-Krüger	*Exploiting natural parallelism in algebraic geometric structures*
A. Paffenholz	*polyDB: A Database for Discrete Geometric Objects*
Y. Ren	*Tropical homotopies served two ways in OSCAR*
D. Roe	*Finite Groups and K3 surfaces in the LMFDB*
K. Schaller	*Computing Newton–Okounkov bodies*
F. Sottile	*Frontiers of Arithmetic in Enumerative Geometry*
S. Veneziale	Machine learning detects terminal singularities
J. Voight	*Future of parallel computation and Magma*
T. Yahl	*Galois groups of purely lacunary polynomial systems*
M. Zach	*On the hunt for an Enriques Surface automorphism with minimal entropy*

[4] `https://www.birs.ca/events/2024/5-day-workshops/24w5229`