# Teaching Linear Algebra in a Mechanized Environment

Robert M. Corless
joint work with David Jeffrey and Azar Shakoori

The Ontario Research Centre for Computer Algebra, The Rotman Institute of Philosophy, and The School of Mathematical and Statistical Sciences, Western University, Canada
Cheriton School of Computer Science, University of Waterloo, Canada
Editor-in-Chief, Maple Transactions

# Announcement: Maple Transactions

**Maple Transactions**
an open access journal with no page charges
mapletransactions.org

We welcome expositions on topics of interest to the Maple community, including in computer-assisted research in mathematics, education, and applications. Student papers especially welcome.

# Context

- Teaching *Computational Mathematics* is increasingly important (Data Science, Visualization, Machine Learning, . . .)
- This is *difficult* because computational mathematics involves several things at once: mathematics, programming, complexity, and numerical stability, because of the *compromises* needed for efficiency.
- **Incorporating new things means removing old things** because we have only finite time to teach, and the students are learning other things as well.
- **Active Learning** is by now recognized as being by far the most effective way to learn.
- This talk will focus on Linear Algebra.

# Is technology disheartening?

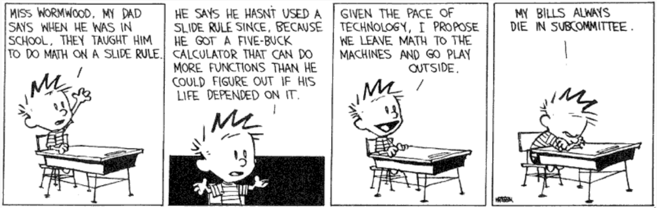


Figure 1: Hard to compete with a machine

# Isn’t technology taken into account already?

No, at least not everywhere.

There have been arguments for at least fifty years about this. Maybe the best paper on the subject is Bruno Buchberger’s 1990 paper “Should Students Learn Integration Rules?” [Link] where he summarizes a resolution of previous fierce discussions with colleagues into what is known now as the “White Box/Black Box Principle”. [I will give details shortly.]

But not everyone knows this. Let’s look first at a **recent bad example** (from my own University)

# How not to do it

We find the following question from a 2022 Math 1600 exam at Western (no calculators or cell phones allowed):

Find the inverse of the matrix

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}. \tag{1}$$

This question is obsolete in at least two ways. First, "nearly anything that can be done with the matrix inverse can be done without it." Second, who inverts 3 by 3 matrices by hand nowadays?

Maybe a better question: Does this matrix factor $A = LU$ without pivoting? **[can't tell without doing it]** Matrix factorings are *far* more useful than matrix inverses.

> *"Linear algebra is the first course where the student encounters algebra, analysis, and geometry all together at once."*

—William (Velvel) Kahan,
to RMC at the 4th SIAM Linear Algebra Conference in Minneapolis 1991

See Kahan's paper Mathematics Written in Sand [Link] for an early and prescient view of the use of computational environments as "computational laboratories."

He envisaged the student as being an active explorer.

# Challenges from floating-point arithmetic

- "Admit, for instance, the existence of a minimum magnitude, and you will find that the minimum which you have introduced, small as it is, causes the greatest truths of mathematics to totter." — Aristotle, "On the Heavens"
- Floats are *not associative*: $a + (b + c) \neq (a + b) + c$ necessarily. For instance $-M + (M + 1) = 0$ while $(-M + M) + 1 = 1$ if $M = 3.14 \cdot 10^{17}$.
- This (and other features such as UINT32s) break students' models of how the world works.
- We have to enable students to deal with floats and UINT32s.
- One has to rethink proofs in these contexts.

# Proofs in a first Linear Algebra course

Here are some things one might like to prove in the first course.

- Cramer's Rule
- that the Normal Equations $A^H Ax = A^H b$ give the least-squares solution to $Ax = b$
- the matrix determinantal lemma $\det(A + uv^H) = (1 + v^H A^{-1} u)\det(A)$ under certain conditions

[Link] Keith Devlin: What is a Mathematical Proof? "Proofs are stories that convince suitably qualified others that a certain statement is true."

It's our job to train people so that they are qualified.

# Programming versus Proof

But in North America, generic students are no longer exposed to proofs in high school. They need motivation when they *do* encounter proofs (usually in the first Linear Algebra course). We contend that **programming can provide such a motivation**. Ed Barbeau (U. Toronto) says that "there should be no proof without doubt." Meaning that if the students don't doubt the statement, proving it is counterproductive.

Having them write programs, and encounter bugs, can be very motivating. They might be interested in proving their programs correct, after that experience. They should try, by hand. We think that *only then* will the concept of an automatic prover have any meaning for them.

# Expanding on that a bit

- A classic proof of existence is somewhat akin to proving that a program terminates.
- Proving that a method gives the correct answer is akin to proving that a program terminates with the correct answer.
- What is frequently needed is a proof that when a program terminates, it gives the exact answer to a nearby question. This involves the original non-mathematical context.

# Buchberger's White Box/Black Box

In his 1990 paper Buchberger outlines the *White Box/Black Box* principle for teaching mathematics in a mechanized environment.

- When learning a particular concept the 1st time, say determinant, the student is not allowed to use the `Determinant` routine
- When using the concept in learning more advanced things (eg Cramer's Rule) they *are* allowed to use it
- Why? People need a certain amount of human action to internalize a concept
- At that point we say the concept has become an **answer** and not a **question**

Sometimes the rule can be broken profitably; one can use Black Box for a while, then open up the box and see what's inside it.

Again, Buchberger is thinking of the student as being *active*.

# We have already changed

Because of Wolfram Alpha [Link] homework assignments already have to be different. Then there's Chegg and MathOverflow and Maple Primes and many more. This puts more weight on exams. I am not even going to talk about ChatGPT!

Examining *with* technology is more stressful for the student. We have had lab equipment failures, software failures, and lots more. It's harder to invigilate, too.

But going the no-technology route requires banning cell phones (illegal in some countries). And limits the kind of question you can ask.

# A possibly better exam question

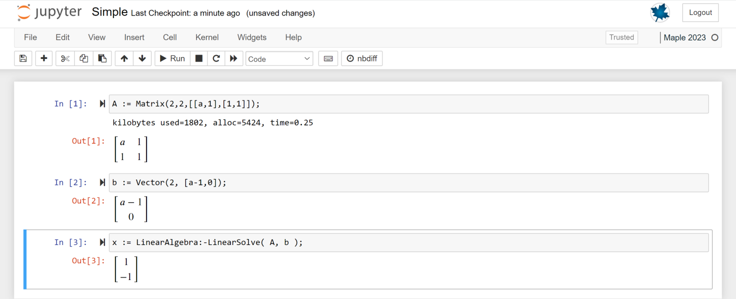


**Figure 2:** Question: Maple gives the output above. Is it correct? Is it correct *and complete*?

Take the inductive proof of the modified Gram-Schmidt process[1] and write a program to compute the *QR* factoring of an arbitrary rectangular matrix with complex entries. Prove your program is correct. Test it on matrices with complex entries. Is the output of your program *continuous* with respect to the input data[2]?

[1] I have a YouTube video on this induction for Modified Gram-Schmidt [Link]; one of my more popular videos, in fact.

[2] Probably not, and this is a very hard problem.

# Reflections

- A student quotation: “What *good* does [the technology] do? I mean, I *liked* plug-and-chug. Now I have to think about what the answer *means*!”
- The problem of case dissection [link] is *hard*; there can be a combinatorial explosion of cases. The program needs the user to tell it about the assumptions on the variables.
- Using technology requires training. (syntax was not at all obvious to me before I learned it)
- We need to re-think our exam questions in the light of student needs, which have changed with the changing environment.

The case of calculus is discussed in more detail in my 2004 paper Computer-Mediated Thinking [link].

# But, *which* technology?

One wants to reduce, reuse, and recycle: instead of using a calculus tool in one course and a linear algebra tool in another and a statistical tool in yet another, one wants to present as much of a common front as possible. Jupyter notebooks are very popular just now, and perhaps one should "go with the flow."

Of course, there are many kernels one can connect to Jupyter notebooks: Maple, Mathematica, Matlab, Python, R, Sagemath, and so on.

Let the religious wars recommence…

# Mathematical Notions Strengthened by Programming

Several mathematical notions are strengthened by these exercises.

- One can use *mathematical induction* to prove correctness of a program
- Real analysis using $\varepsilon$–$\delta$ proofs works well with IEEE floats by means of the IEEE guarantees $\mathrm{fl}(x \mathrm{~op~} y) = (x \mathrm{~op~} y)(1+\delta)$ for some $|\delta| \leq \mathsf{u}$ where $\mathsf{u}$ is the *unit roundoff*, $2^{-53}$ for double precision
- Practice with functions is always useful
- Simply working with visualizations improves people's feel for geometry.

# Unintentional Power

Tampering with the first Linear Algebra course will have significant downstream effects. Who will benefit from the changes? Will anyone be harmed? And do we really *have* to change?

Old-fashioned lecturing Linear Algebra teaches students:

- To sit still for 50 minutes
- To take notes (gives practice writing or typing)
- Many other things (maybe gives them practice in listening to a chain of argument)

# But we maintain that the Big Ideas can be taught better with an active technology

- Functions
- Continuity (Carathéodory uses this to define differentiability)
- Limits and Convergence
- Reductionism (cf. Steven Strogatz' *Infinite Powers*)
- maybe, Proof?

# In Summary

- The environment in which our students use mathematics has changed
- The mathematics we teach must change with that
- Not only the methods we use to teach mathematics (videos seem extraordinarily popular) but also the **topics must change**
- Students need active training in the responsible use of technology
- That means we have to keep up with the technology
- The reactionaries are winning in some places now, but they should not be allowed to win overall

# Thank you

Thank you for listening!

MINISTERIO
DE CIENCIA
E INNOVACIÓN

This work was partially supported by NSERC grant RGPIN-2020-06438, and partially supported by the grant PID2020-113192GB-I00 (Mathematical Visualization: Foundations, Algorithms and Applications) from the Spanish MICINN. I also thank CUNEF Universidad for financial support.

# References

[1] Harold Abelson. Computation in the undergraduate curriculum. *International Journal of Mathematical Education in Science and Technology*, 7(2):127–131, 1976.

[2] Luis von Ahn. Duolingo: learn a language for free while helping to translate the web. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 1–2, 2013.

[3] Robert M Aiken and Richard G Epstein. Ethical guidelines for AI in education: Starting a conversation. *International Journal of Artificial Intelligence in Education*, 11:163–176, 2000.

[4] Aristotle. *On the Heavens*. 350BCE.

[5] Robert L Armacost and Julia Pet-Armacost. Using mastery-based grading to facilitate learning. In *33rd Annual Frontiers in Education, 2003. FIE 2003.*, volume 1, pages T3A–20. IEEE, 2003.

[6] Jack Betteridge, James H Davenport, Melina Freitag, Willem Heijltjes, Stef Kynaston, Gregory Sankaran, and Gunnar Traustason. Teaching of computing to mathematics students: Programming and discrete mathematics. In *Proceedings of the*

*3rd Conference on Computing Education Practice*, pages 1–4, 2019.

[7] Jack Betteridge, Eunice Y. S. Chan, Robert M. Corless, James H. Davenport, and James Grant. Teaching programming for mathematical scientists. In *Mathematics Education in the Age of Artificial Intelligence*, pages 251–276. Springer International Publishing, 2022. URL *https://doi.org/10.1007/978-3-030-86909-0_12*.

[8] P Bond. The era of mathematics–review findings on knowledge exchange in the mathematical sciences. engineering and physical sciences research council and the knowledge transfer network. *https://epsrc.ukri.org/newsevents/news/mathsciencereview/*, 2018.

[9] J. Borwein and K. Devlin. The computer as crucible: An introduction to experimental mathematics. *The Australian Mathematical Society*, page 208, 2009.

[10] J. M. Borwein and P. B. Borwein. Strange series and high precision fraud. *The American mathematical monthly*, 99(7):622–640, 1992.

[11] Paul L. Boynton. What constitutes good teaching? *Peabody Journal of Education*, 28(2):67–73, 1950. ISSN 0161956X. URL `http://www.jstor.org/stable/1489824`.

[12] R.J. Bradford, J.H. Davenport, and C.J. Sangwin. A Comparison of Equality in Computer Algebra and Correctness in Mathematical Pedagogy. In J. Carette *et al.*, editor, *Proceedings Intelligent Computer Mathematics*, pages 75–89, 2009.

[13] Meredith Broussard. *Artificial unintelligence: How computers misunderstand the world*. MIT Press, 2018.

[14] Bruno Buchberger. Should students learn integration rules? *ACM Sigsam Bulletin*, 24(1):10–17, 1990.

[15] Neil J. Calkin, Eunice Y.S. Chan, and Robert M. Corless. *Computational Discovery on Jupyter*. SIAM, 2023 (in progress). URL `https://computational-discovery-on-jupyter.github.io/Computational-Discovery-on-Jupyter/`.

[16] A.C. Camargos Couto, M. Moreno Maza, D. Linder, D.J. Jeffrey, and Corless R.M. Comprehensive LU Factors of Polynomial Matrices. *MACIS 2019*, pages 80–88, 2020.

[17] David Carlson, Charles R. Johnson, David Lay, and A. Duane Porter. Gems of exposition in elementary linear algebra. *The College Mathematics Journal*, 23(4):299–303, September 1992. doi: 10.1080/07468342.1992.11973473.

[18] David Carlson, Charles R Johnson, David C Lay, and A Duane Porter. The linear algebra curriculum study group recommendations for the first course in linear algebra. *The College Mathematics Journal*, 24(1):41–46, 1993.

[19] Eunice Y. S. Chan and Robert M Corless. A random walk through experimental mathematics. In *Jonathan M. Borwein Commemorative Conference*, pages 203–226. Springer, 2017.

[20] R. Chapman and F. Schanda. Are We There Yet? 20 Years of Industrial Theorem Proving with SPARK. In *Proceedings of Interactive Theorem Proving*, pages 17–26, 2014.

[21] R. M. Corless and J. E. Jankowski. Variations on a theme of Euler. *SIAM Review*, 58(4):775–792, 2016.

[22] R. M. Corless and D. J. Jeffrey. Scientific computing: One part of

the revolution. *Journal of Symbolic Computation*, 23(5):485–495, 1997.

[23] R. M. Corless, C. Essex, and P. J. Sullivan. *First year engineering mathematics using supercalculators*. SciTex, The University of Western Ontario, 2 edition, 1995.

[24] Robert M Corless. Six, lies, and calculators. *The American mathematical monthly*, 100(4):344–350, 1993.

[25] Robert M Corless. Computer-mediated thinking. *Proceedings of Technology in Mathematics Education*, 2004. `https://github.com/rcorless/rcorless.github.io/blob/main/CMTpaper.pdf`.

[26] Robert M Corless and David J Jeffrey. The Turing factorization of a rectangular matrix. *ACM SIGSAM Bulletin*, 31(3):20–30, 1997.

[27] Robert M Corless, David J Jeffrey, and David R Stoutemyer. Integrals of functions containing parameters. *The Mathematical Gazette*, 104(561):412–426, 2020.

[28] Robert M. Corless, David J. Jeffrey, and Azar Shakoori. Teaching linear algebra in a mechanized mathematical environment, 2023. Accepted to CICM 2023, Cambridge.

[29] James H Davenport, David Wilson, Ivan Graham, Gregory Sankaran, Alastair Spence, Jack Blake, and Stef Kynaston. Interdisciplinary teaching of computing to mathematics students: Programming and discrete mathematics. *MSOR Connections*, 14(1):1–8, 2014.

[30] James H. Davenport, Alan Hayes, Rachid Hourizi, and Tom Crick. Innovative pedagogical practices in the craft of computing. In *2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, pages 115–119. IEEE, 2016.

[31] J.H. Davenport. Methodologies of Symbolic Computation. In *Proceedings AISC 2018*, pages 19–33, 2018.

[32] F. Fischer, K. Böttinger, H. Xiao, C. Stransky, Y. Acar, M. Backes, and S. Fahl. Stack Overflow Considered Harmful? The Impact of Copy&Paste on Android Application Security. *38th IEEE Symposium on Security and Privacy (SP)*, pages 121–136, 2017.

[33] Michael Frame and Benoit B. Mandelbrot. *Fractals, graphics, and mathematics education*. Number 58. Cambridge University Press, 2002.

[34] Scott Freeman, Sarah L Eddy, Miles McDonough, Michelle K Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23):8410–8415, 2014.

[35] Laureano Gonzalez-Vega. Using linear algebra to introduce computer algebra, numerical analysis, data structures and algorithms (and to teach linear algebra, too). *International Journal of Computer Algebra in Mathematics Education*, 6(3):209, 1999. ISSN 1362-7368. URL `https://www.learntechlib.org/p/92417`.

[36] R. W. Hamming. *Methods of mathematics applied to calculus, probability, and statistics*. Courier Corporation, 2012.

[37] J. Handelsman, D. Ebert-May, R. Beichner, P. Bruns, A. Chang,

R. DeHaan, J. Gentile, S. Lauffer, J. Stewart, S. M. Tilghman, et al. Scientific teaching. *Science*, 304(5670):521–522, 2004.

[38] Stephen Hegedus, Colette Laborde, Corey Brady, Sara Dalton, Hans-Stefan Siller, Michal Tabach, Jana Trgalova, and Luis Moreno-Armella. *Uses of technology in upper secondary mathematics education.* Springer Nature, 2017.

[39] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, Philadelphia, 2nd edition, 2002.

[40] David J Jeffrey and Robert M Corless. Linear algebra in Maple®. In Leslie Hogben, editor, *Handbook of Linear Algebra*, pages 89–1. Chapman and Hall/CRC, 2nd edition, 2013.

[41] William M. Kahan. Handheld calculator evaluates integrals. *Hewlett-Packard Journal*, 31(8):23–32, 1980.

[42] William M. Kahan. Mathematics written in sand. In *Proc. Joint Statistical Mtg. of the American Statistical Association*, pages 12–26, 1983. `http://people.eecs.berkeley.edu/~wkahan/MathSand.pdf`.

[43] Zoltán Kovács, Tomás Recio, Philippe R Richard, and M Pilar Vélez. GeoGebra automated reasoning tools: A tutorial with examples. In *Proceedings of the 13th International Conference on Technology in Mathematics Teaching*, pages 400–404, 2017.

[44] Zoltán Kovács, Tomás Recio, and M. Pilar Vélez. Automated reasoning tools with GeoGebra: What are they? what are they good for? In *Mathematics Education in the Age of Artificial Intelligence*, pages 23–44. Springer International Publishing, 2022. doi: 10.1007/978-3-030-86909-0_2. URL https://doi.org/10.1007/978-3-030-86909-0_2.

[45] David C Lay, Steven R Lay, and Judi McDonald. *Linear algebra and its applications*. Pearson Education, 2016.

[46] Ao Li and Robert M Corless. Revisiting Gilbert Strang's "A chaotic search for *i*". *ACM Communications in Computer Algebra*, 53(1): 1–22, 2019.

[47] Rose Luckin, Wayne Holmes, Mark Griffiths, and Laurie B Forcier. *Intelligence unleashed: An argument for AI in education*. Pearson Education, 2016. ISBN 9780992424886.

[48] Benoit B. Mandelbrot and Michael Frame. Some reasons for the effectiveness of fractals in mathematics education. *Fractals, Graphics, & Mathematics Education*, pages 3–9, 2002.

[49] Cleve Moler. Roots—of polynomials, that is, 1991. URL *https://www.mathworks.com/company/newsletters/articles/roots-of-polynomials-that-is.html*.

[50] John Monaghan, Luc Trouche, and Jonathan M Borwein. *Tools and mathematics*. Springer, 2016.

[51] S. Papert. *Mindstorms*. Basic Books, 2 edition, 1993.

[52] John Paxton. Live programming as a lecture technique. *Journal of Computing Sciences in Colleges*, 18(2):51–56, 2002.

[53] Christopher Rackauckas and Qing Nie. Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of open research software*, 5(1), 2017.

[54] Peter A. Rosati, Robert M. Corless, G. Christopher Essex, and Paul J. Sullivan. An evaluation of the HP28S calculator in calculus. *Australian J. Engineering Education*, 3(1):79–88, 1992.

[55] Marc J Rubin. The effectiveness of live-coding to teach introductory programming. In *Proceeding of the 44th ACM technical symposium on Computer science education*, pages 651–656, 2013.

[56] Elaine Seymour and Nancy M Hewitt. *Talking about leaving*. Westview Press, Boulder, CO, 1997.

[57] J.R. Slagle. A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus. *Journal of the ACM*, 10:507–520, 1963.

[58] Richard C Smith and Edwin F Taylor. Teaching physics on line. *American Journal of Physics*, 63(12):1090–1096, 1995.

[59] G. Strang. Too much calculus. `http://www-math.mit.edu/~gs/papers/essay.pdf`, 2001.

[60] Gilbert Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, Wellesley, MA, 1986.

[61] Diane Hobenshield Tepylo and Lisa Floyd. Learning math through coding. 2016. *https://researchideas.ca/mc/learning-math-through-coding/*.

[62] Lloyd N Trefethen and David Bau. *Numerical linear algebra*, volume 181. Siam, 2022.

[63] Charles F. Van Loan and K.-Y. Daisy Fan. *Insight Through Computing - A MATLAB Introduction to Computational Science and Engineering*. SIAM, 2010. ISBN 978-0-89871-691-7.

[64] Greg Wilson. Software carpentry: getting scientists to write better code by making them more productive. *Computing in Science & Engineering*, 8(6):66–69, 2006.

[65] Greg Wilson, Dhavide A Aruliah, C Titus Brown, Neil P Chue Hong, Matt Davis, Richard T Guy, Steven HD Haddock, Kathryn D Huff, Ian M Mitchell, Mark D Plumbley, et al. Best practices for scientific computing. *PLoS Biol*, 12(1):e1001745, 2014.