

Can AI Compute Marco's Favourite Eigenvalues?

The answer is yes (but subtle)!



Matthew to Marco

Hi Marco,

I hope that all is well! Quick question - what is your favourite eigenvalue?

Best, Matt

Marco to Matthew

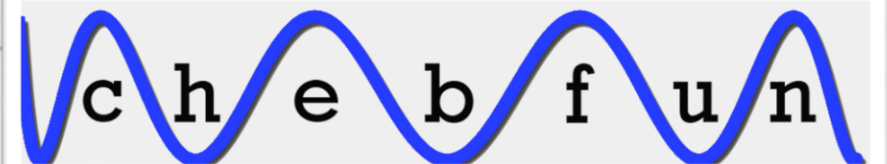
Hi Matt,

It is probably the ground state of hydrogen, so the bottom of the spectrum of the 1D Schroedinger operator with potential $V(r) = 2/r^2 - 1/r$. I believe its value is -0.0625 , or $-1/16$, and I seem to remember my first computed approximation was -0.0621 , before we had implemented the algorithm for doing better boundary conditions and meshing.

It was the first eigenvalue I ever computed numerically for a singular problem.

Many years later I was in Oxford visiting Nick Trefethen around the time he had just published Chebfun. He wanted to show me how good it was and asked me for a problem to test it. I gave him this problem, Chebfun promptly fell over, and Nick complained that it was not fair! Goodness, I had a lot to learn about social skills.

All the best, Marco



Matthew to Marco

That is a wonderful story! Do you also have a favourite non-normal version of this?

Marco to Matthew

I would choose one particular resonance pair from my 2014 many-author paper

<https://orca.cardiff.ac.uk/id/eprint/64130/8/OA-20142015-45.pdf>

It's the one at the bottom of page 15. It was an utter beast to compute. Never mind a 100 digit challenge, just getting 4 digits would have been nice!

**Can AI crack these
rigorously to ten digits?**

PROCEEDINGS
— OF —
THE ROYAL SOCIETY 

rspa.royalsocietypublishing.org

Research



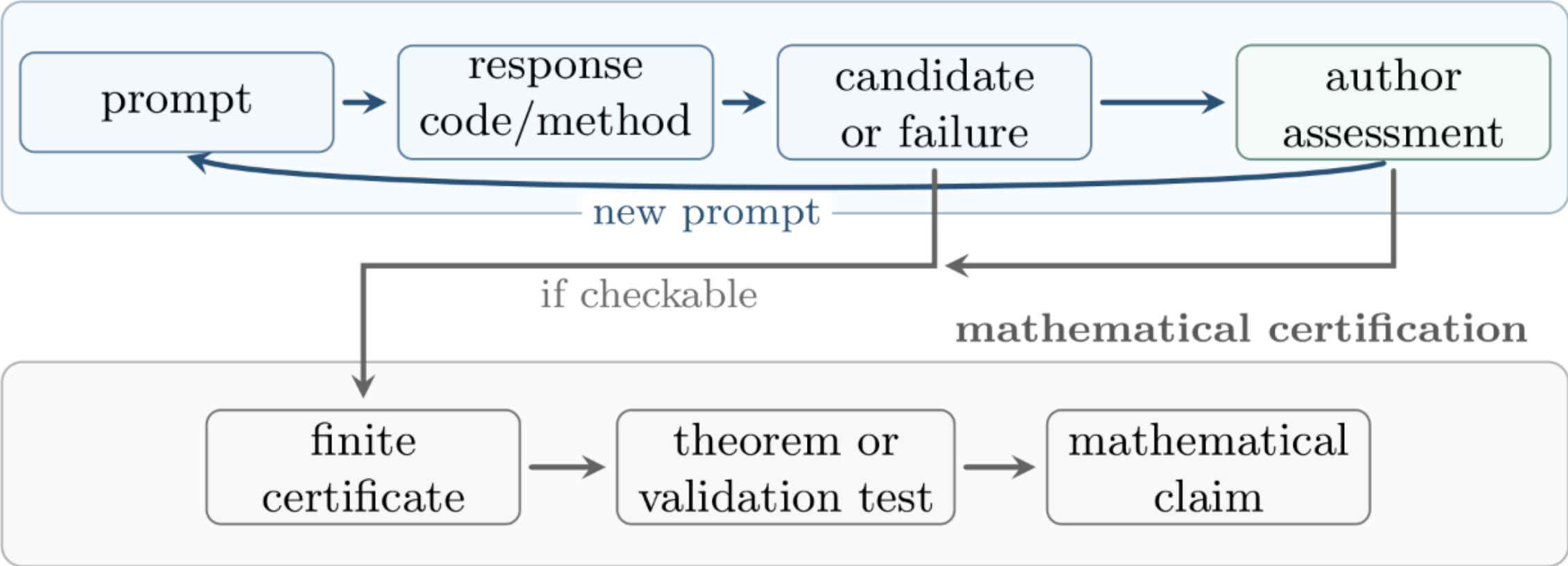
Guaranteed resonance
enclosures and exclosures for
atoms and molecules

Sabine Bögli¹, B. Malcolm Brown², Marco Marletta³,
Christiane Tretter^{4,5} and Markus Wagenhofer⁶

¹Mathematisches Institut, Universität Bern, Alpeneggstrasse 22,
3012 Bern, Switzerland

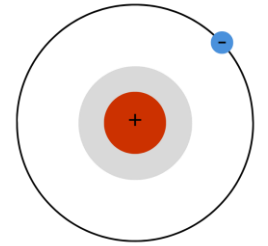
²School of Computer Science, Cardiff University, 5 The Parade,

AI-assisted exploration



First problem was “too easy”

Hydrogen Atom



$$-u''(r) + \left(\frac{2}{r^2} - \frac{1}{r} \right) u(r) = \lambda u(r), \quad u(r) = O(r^2) \quad (r \downarrow 0)$$

Johann Jakob Balmer



Johannes Rydberg



Niels Bohr



Erwin Schrödinger



$$\lambda_n = -\kappa^2 = -\frac{1}{4(n+1)^2}, \quad u_n(r) = r^2 e^{-r/(2(n+1))} L_{n-1}^{(3)}(r/(n+1)), \quad n = 1, 2, \dots$$

AI found this exactly with a good ansatz (not from memory)

Prompt 1: modifying Marletta's eigenvalue

Below, I will paste an email exchange with Marco Marletta. He describes his favourite eigenvalue. Let's change the potential to $\frac{2}{r^2} - \frac{1}{r(1+r/200)^2}$. I want you to investigate whether this has a negative eigenvalue. If so, I want you to come up with a way to compute its smallest eigenvalue. You will need to rigorously prove error bounds on the eigenvalue and fully describe your computational procedure. You will find Python installed in the current directory. You can probably build a rational computation verification in Python, or you may need to look at interval arithmetic implementations.

AI response, worked for 10m 54s

Yes. The modified potential still has a negative ground-state eigenvalue. I produced a rigorous certificate in `EIGENVALUE.md` and the verifier script `verify_temple.py`. Certified enclosure:

$$-0.05319676154302054294 \leq \lambda_1 \leq -0.05319676154233948813.$$

The proof uses a five-term hydrogenic Ritz vector, computes its Rayleigh quotient and residual with rational interval arithmetic, then applies Temple's inequality. The spectral gap needed for Temple comes from comparing the new potential with Marco's Coulomb potential, whose second $l = 1$ level is $-1/36$. I also ran the verifier successfully with the local Python.

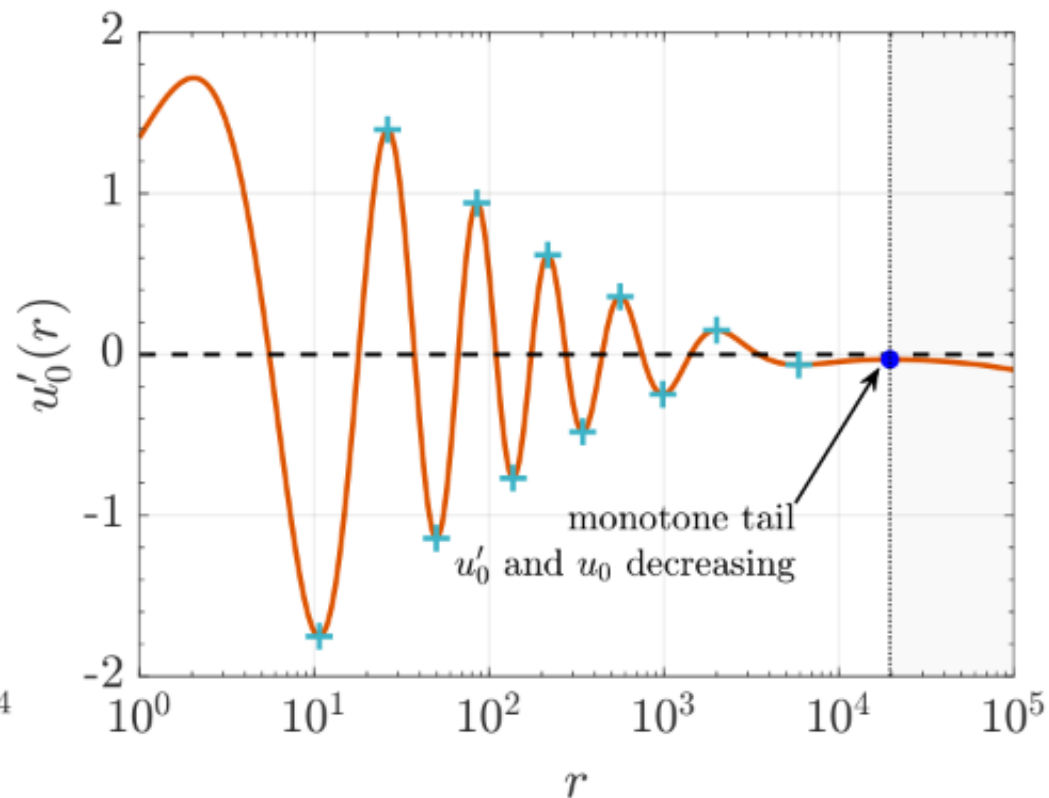
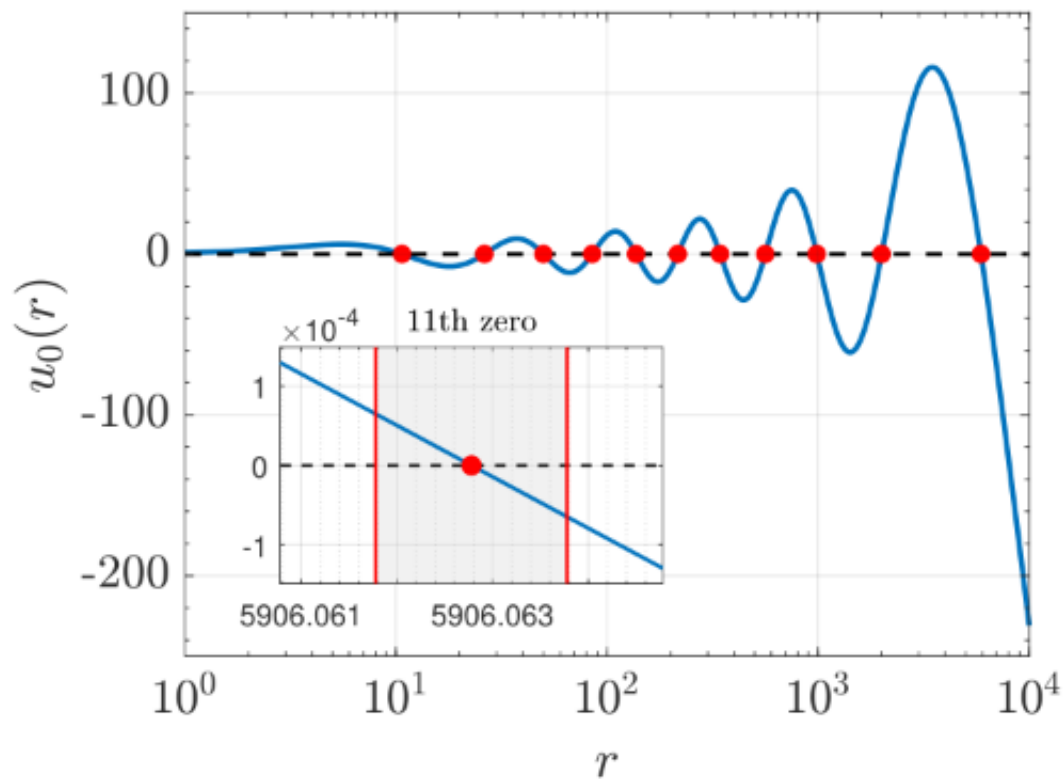
Prompt 2: finite or infinite negative spectrum

Wonderful! Are there infinitely many negative eigenvalues or finitely many? If there are finitely many, say N , compute the first $\min(10, N)$ negative eigenvalues to 10 digits. If there are infinitely many, compute the smallest 10 to 10 digits. Again, everything has to be rigorously justified.

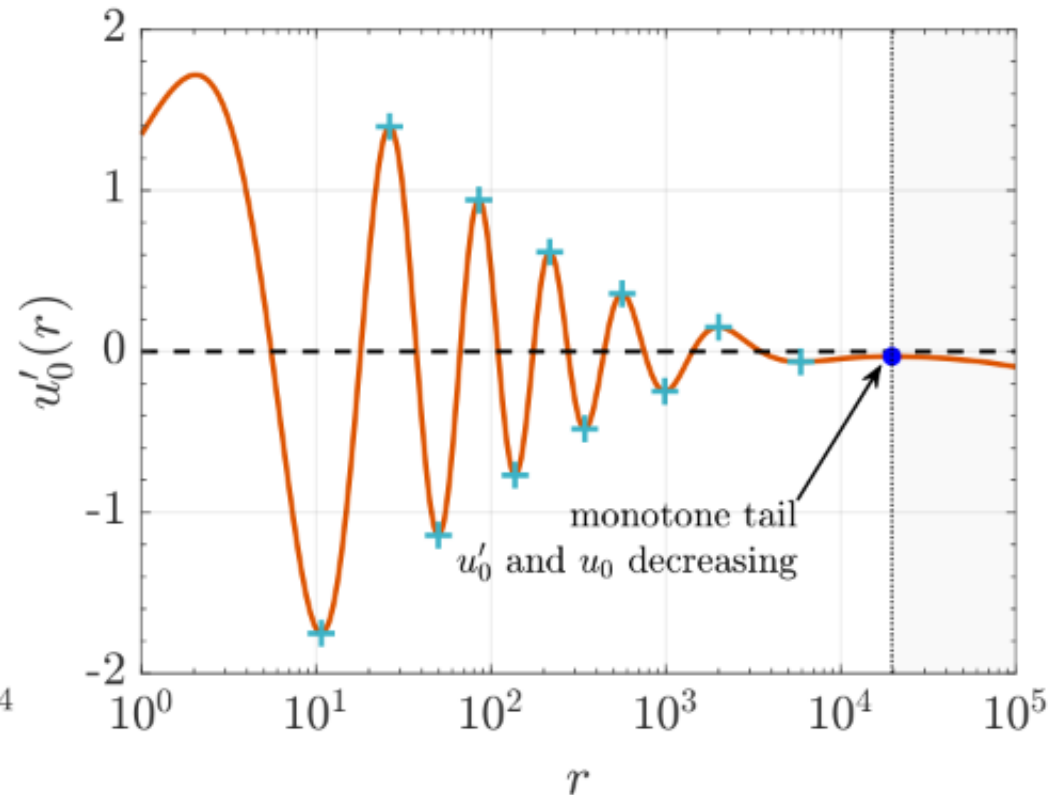
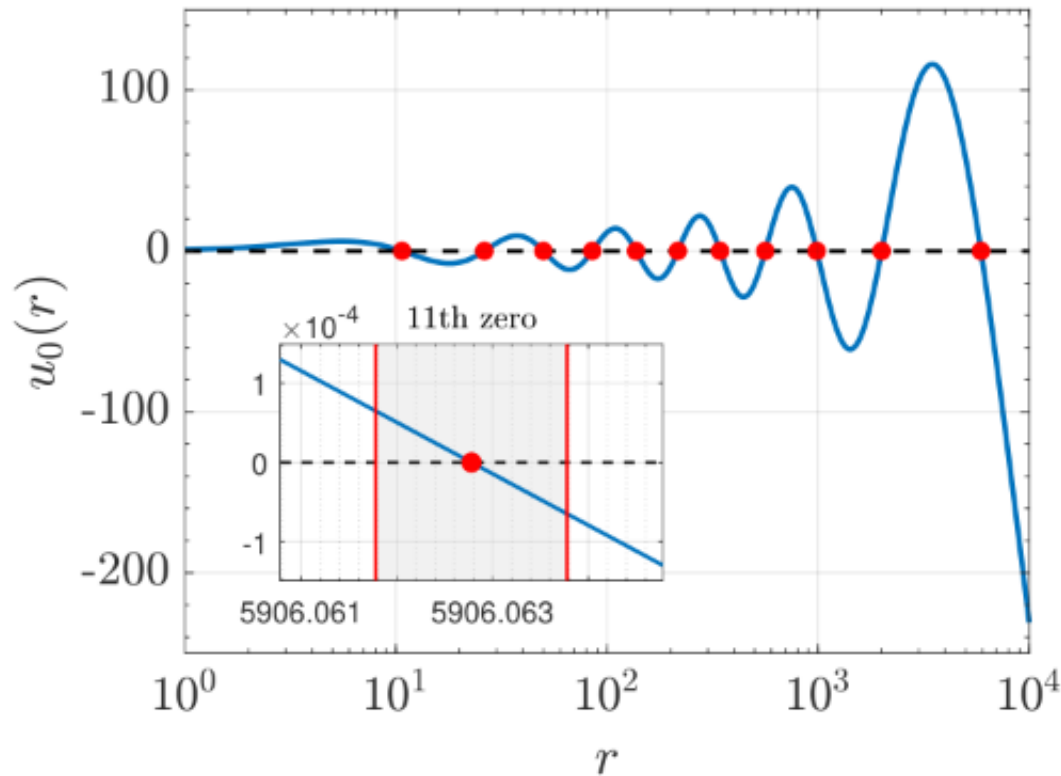
AI response, worked for 21m 56s

It is finite: there are exactly 11 negative eigenvalues. I wrote the justification in `SPECTRUM.md`, with the shooting generator `spectrum_shooting.py`. The first ten, ordered from most negative upward, are: -0.0531967615 , -0.0193674097 , -0.0082420874 , -0.0036685443 , -0.0016120468 , -0.0006697369 , -0.0002515677 , -0.0000804935 , -0.0000198995 , and -0.0000031084 . The proof structure is: the potential is nonnegative for $r \geq 9800 + 4000\sqrt{6}$, so there can only be finitely many negative eigenvalues; the zero-energy regular solution has exactly 11 zeros, giving $N = 11$ by Sturm oscillation; Dirichlet–Neumann bracketing then encloses the first ten half-line eigenvalues in intervals of radius 5×10^{-12} .

$$-u_0''(r) + \left(\frac{2}{r^2} - \frac{1}{r(1+r/200)^2} \right) u_0(r) = 0, \quad u_0(r) \sim r^2 \quad (r \downarrow 0)$$



$$-u_0''(r) + \left(\frac{2}{r^2} - \frac{1}{r(1 + r/200)^2} \right) u_0(r) = 0, \quad u_0(r) \sim r^2 \quad (r \downarrow 0)$$



WARNING 1: AI said it was verified but hadn't used interval arithmetic.

WARNING 2: It missed the analytic form $u_0(r) = 6 \times 200^{3/2} \sqrt{r} P_\nu^{-3} \left(\frac{200 - r}{200 + r} \right)$

- Truncate $(0, \infty)$ to $(0, L)$, compare to Dirichlet & Neumann problem:

$$\lambda_j(H_L^N) \leq \lambda_j(H) \leq \lambda_j(H_L^D), \quad j = 1, \dots, 11$$

- Frobenius expansion on $(0, 10^{-2})$ then verified Taylor expansions...

j	midpoint of certified interval	radius	value to 10 decimal places
1	$-5.3196761542354 \times 10^{-2}$	5×10^{-12}	-0.0531967615
2	$-1.9367409728912 \times 10^{-2}$	5×10^{-12}	-0.0193674097
3	$-8.2420874282006 \times 10^{-3}$	5×10^{-12}	-0.0082420874
4	$-3.6685443298679 \times 10^{-3}$	5×10^{-12}	-0.0036685443
5	$-1.6120468156717 \times 10^{-3}$	5×10^{-12}	-0.0016120468
6	$-6.6973685120550 \times 10^{-4}$	5×10^{-12}	-0.0006697369
7	$-2.5156770074481 \times 10^{-4}$	5×10^{-12}	-0.0002515677
8	$-8.0493482670423 \times 10^{-5}$	5×10^{-12}	-0.0000804935
9	$-1.9899482902437 \times 10^{-5}$	5×10^{-12}	-0.0000198995
10	$-3.1084439935000 \times 10^{-6}$	5×10^{-12}	-0.0000031084
11	$-1.6239698497433 \times 10^{-7}$	5×10^{-12}	-0.0000001624



Total AI time: 38 minutes 19 seconds.

Total human time to understand, check, correct, and polish in paper: 3 days

Prompt 4: from Marco's desired four digits to ten

Wonderful, now let's do a different problem. Marco sent me the message pasted below. I want us to push the boat out. Figure out a way to compute this resonance to 10 digits of accuracy (not just the 4 digits that Marco would like). It has to be fully rigorous and accurate. We may need to come up with new ideas to do this.

AI response, worked for 1h 15m 12s

I've identified the target and obtained high-precision candidate values. The pair is the Abramov pair $\hat{\lambda}_2^-, \hat{\lambda}_2^+$ in Table 6, at the bottom of page 15 of Bögli–Brown–Marletta–Tretter–Wagenhofer. Using the paper's stretched scaling $(R, \theta, X) = (10, 0.76, 10)$, I get:

$$(4.1) \quad \hat{\lambda}_2^- \approx 0.9180018173874698 - 17.00068824973865 i,$$

$$(4.2) \quad \hat{\lambda}_2^+ \approx 0.9178768454495876 - 17.00064483695551 i.$$

These are high-precision candidates, not yet a rigorous certificate. The route I think will get us the rigorous 10 digits is: work in the z -plane, shoot backwards from $X = 10$ with the outgoing condition $y'(X) = -\sqrt{-z}$, form the Dirichlet and Neumann determinants, propagate the ODE and variational equation with interval Taylor arithmetic, apply an interval Newton/Krawczyk test on small boxes in z , use the paper's truncation bound $\alpha_{X,\theta,R} \leq 1.77 \times 10^{-17}$, so that the tail error is far below the requested ten-digit scale. The next step is to turn the Taylor shooting into an interval certificate.

Truncate domain, scale the variable:

$$-y''(x) + q_{\theta,R}(x)y(x) = z y(x), \quad 0 \leq x \leq X$$

$$q_{\theta,R}(x) = R^2 e^{2i\theta} (R^2 e^{2i\theta} x^2 - 1.6) \exp\left(-\frac{R^2 e^{2i\theta} x^2}{10}\right)$$

$$X = 10 \quad \theta = 0.76 \text{ rad} \quad R = 10 \quad z = R^2 e^{2i\theta} \lambda$$

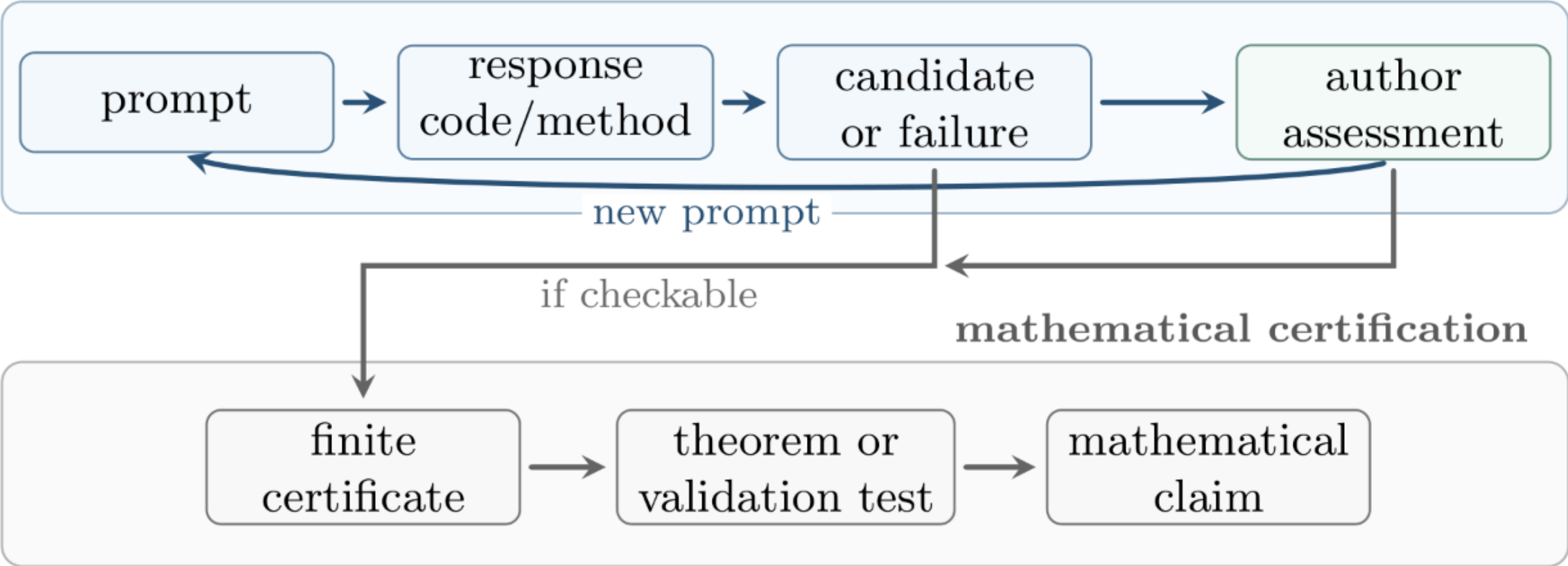
$$y'(X) + k(z)y(X) = 0 \quad k(z) = \sqrt{-z}$$

Challenges: Non-normal, shooting from $x = 10$ to $x = 0$ very unstable, need to correct for the truncation from $(0, \infty)$ to $(0, X)$

AI failed but corrected itself to find the right method!

route	obstruction	lesson
amplitude shooting	enclosures grew rapidly in the unstable solution component	raw amplitudes are poor variables for long validated propagation
variation-of-constants and amplitude Lohner forms	free growth and some wrapping were reduced, but one-way amplification remained	better local coordinates did not cure the long composition
midpoint or Wronskian matching	the enclosure of the matching determinant remained too wide	this characteristic formulation was still too ill-conditioned
Riccati/projective shooting	amplitude growth disappeared, but a chart denominator could no longer be certified to exclude zero	solution lines are the appropriate variables, but the long one-way composition remained unsuitable
centered projective Lohner forms	local projective remainders initially stayed small but later widened	the node lines should be treated as simultaneous unknowns
more precision	roundoff shrank while the principal widths and failure point persisted	the tested formulation was not roundoff limited
global projective multiple shooting	short validated maps coupled by projective matching equations	a componentwise Krawczyk inclusion succeeded

AI-assisted exploration



The successful method

- Matching projective coordinates:

$$\begin{pmatrix} y(x_{j+1}) \\ y'(x_{j+1}) \end{pmatrix} = T_j(z) \begin{pmatrix} y(x_j) \\ y'(x_j) \end{pmatrix}$$

$$\begin{pmatrix} a_j(z) & b_j(z) \\ c_j(z) & d_j(z) \end{pmatrix} = P_{j+1}^{-1} T_j(z) P_j$$

$$w_{j+1} = f_j(z, w_j) = \frac{c_j(z) + d_j(z)w_j}{a_j(z) + b_j(z)w_j}$$

$$x_j = jh, \quad h = 0.0025, \quad j = 0, \dots, 4000$$

$$P_j = \frac{1}{(|Y_j|^2 + |Y_j'|^2)^{1/2}} \begin{pmatrix} Y_j & -\overline{Y_j'} \\ Y_j' & \overline{Y_j} \end{pmatrix}$$

$$F : \mathbb{C}^{4001} \times \mathbb{C} \rightarrow \mathbb{C}^{4002} \quad W = (w_0, \dots, w_{4000})$$

$$F_0(W, z) = w_0 - w_{\text{left}}(z),$$

$$F_{j+1}(W, z) = w_{j+1} - f_j(z, w_j), \quad j = 0, \dots, 3999,$$

$$F_{4001}(W, z) = w_{4000} - w_{\text{right}}(z).$$

- Finite-interval $(0, X)$ Krawczyk-type inclusion:

$$\mathcal{B} = \{(w_0, \dots, w_{4000}, z) : |w_j| \leq r_j, \quad j = 0, \dots, 4000, \quad |z - z_0| \leq 10^{-12}\}$$

$$(W_0, z_0) - \Upsilon F(W_0, z_0) + (I - \Upsilon F'(\mathcal{B}))(W_0, z_0) \subset \text{int } \mathcal{B}$$

WARNING 3: AI did not correctly treat the tail: truncation from $(0, \infty)$ to $(0, X)$

This came later...

- Dealing with the tail (X, ∞) :

$$\alpha_{X,\theta,R} = \int_X^\infty |q_{\theta,R}(x)| dx < 1 \quad \text{and} \quad \varepsilon = \alpha_{X,\theta,R}/(1 - \alpha_{X,\theta,R}),$$

accounts for
tail uncertainty

$$y(X) = 1 + \eta_0, \quad y'(X) = -k(z)(1 + \eta_1), \quad |\eta_0|, |\eta_1| \leq \varepsilon$$

$$(W_0, z_0) - \Upsilon F(W_0, z_0) - \Upsilon E + (I - \Upsilon F'(\mathcal{B}))(\mathcal{B} - (W_0, z_0)) \subset \text{int } \mathcal{B}$$

CERTIFICATE 4.1 (Resonance pair). *Let*

$$c_D = 0.9180018173874697859784363 - 17.0006882497386479271609736 i,$$

$$c_N = 0.9178768454495876175818796 - 17.0006448369555127699716473 i.$$

Then $|\hat{\lambda}_2^- - c_D| < 6.50 \times 10^{-19}$ *and* $|\hat{\lambda}_2^+ - c_N| < 6.50 \times 10^{-19}$. *In particular,*

$$\hat{\lambda}_2^- = 0.9180018174 - 17.0006882497 i, \quad \hat{\lambda}_2^+ = 0.9178768454 - 17.0006448370 i$$

to ten decimal places.



Total AI time: 8 hours 29 minutes 8 seconds.

Total human time to understand, check, correct, and polish in paper: 5 days

Summary

- AI made answers cheap; certificates remained hard.
- Self-adjoint stability gave Temple, Sturm, and bracketing proofs.
- Resonances required global projective multiple shooting.
- Convincing AI routes failed; one “final” certificate failed audit.
- Cheap claims make verification and provenance central.

Summary

**THESE PROBLEMS ARE HARD!
UNTIL RECENTLY WOULD HAVE MADE
UP A CHUNK OF A GOOD PHD THESIS**

- AI made answers cheap; certificates remained hard.
- Self-adjoint stability gave Temple, Sturm, and bracketing proofs.
- Resonances required global projective multiple shooting.
- Convincing AI routes failed; one “final” certificate failed audit.
- Cheap claims make verification and provenance central.

Summary

**THESE PROBLEMS ARE HARD!
UNTIL RECENTLY WOULD HAVE MADE
UP A CHUNK OF A GOOD PHD THESIS**

- AI made answers cheap; certificates remained hard.
- Self-adjoint stability gave Temple, Sturm, and bracketing proofs.
- Resonances required global projective multiple shooting.
- Convincing AI routes failed; one “final” certificate failed audit.
- Cheap claims make verification and provenance central.

9 prompts

Total AI time: 9 hours 7 minutes 27 seconds

Total human time to understand, check, correct, and polish in paper: 8 days

Social warnings

- Some mathematics will become routine suddenly.
- Visible effort now an even poorer measure contribution.
- Danger that students skip apprenticeship where judgment forms.
- Weak arguments can arrive polished, (wrongly?) cited, and coded.
- Inequality will grow between critical users and passive consumers. Resources become critical.
- Numerical analysis must state what counts as evidence.

We must set standards before convenience sets them.

Mathematics > Numerical Analysis

[Submitted on 22 Jun 2026]

Ten Digits on a Train: AI-Assisted Verification of Two Eigenvalue Problems

Matthew J. Colbrook

Accurate numerical eigenvalues are often difficult to certify, especially in singular or non-normal settings. This article reports a human--AI collaboration on two such computations. For a singular self-adjoint Schrödinger operator, a verified zero count and Dirichlet--Neumann bracketing certify the complete negative spectrum to ten decimal places. For a delicate non-normal atom--molecule benchmark, a previously unresolved resonance pair is separated, with each member enclosed to ten digits. The second result is achieved not by increasing the precision of one-way shooting, but by reformulating the problem as a global matching system for projective solution lines. The infinite tail is encoded as uncertainty in the terminal projective data, and a componentwise, tail-robust Krawczyk--Brouwer inclusion supplies the certificate. This gives a reusable architecture for analytic boundary-value systems with ill-conditioned propagation and uncertain asymptotic data. The collaboration also exposes the strengths and limits of AI assistance. AI rapidly produced accurate candidates and plausible proof strategies, but several failed, including one apparently complete tail argument that omitted the componentwise check required by a nonuniform polydisc. Validated computation is a stringent test of AI-assisted mathematics: the output is not merely a number, but a number with a proof. These examples show why the proof object matters, and why human mathematical judgment remained decisive. More broadly, as AI makes code, exposition, and plausible numerical claims inexpensive, standards for verification, attribution, peer review, and training must adapt. The implications are unsettling; the opportunity is extraordinary.

The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with 'main', '1 Branch', and '0 Tags'. A search bar and 'Add file' button are visible. Below this is a table of files:

File Name	Action	Commit Date
resonances	Add files via upload	yesterday
self-adjoint_all_evals	Add files via upload	yesterday
self-adjoint_smallest_eval	Add files via upload	yesterday
CITATION.cff	Add files via upload	yesterday
LICENSE	Initial commit	yesterday
README.md	Add files via upload	yesterday
verify_resonance_certificates.py	Add files via upload	yesterday
verify_selfadjoint_certificates.py	Add files via upload	yesterday

Below the table, there's a 'README' section with a 'BSD-3-Clause license' icon. The title is 'Codes for "Ten Digits on a Train"'. The text explains that this directory accompanies the article and contains code and certificate files. It lists the contributors as Matthew J. Colbrook and provides a link to the paper. A list of languages is shown, with Python at 100%.





A wave is coming.

**The opportunity is enormous,
if our standards rise with it.**

**Thank you, Marco
— and happy 60th.**