

Mathematics Education when Formal Mathematics is Automated

Claes Johnson

KTH Royal Institute of Technology, Stockholm

claesjohnson@gmail.com

October 15, 2026

Abstract

Mathematics is to formulate an equation and to solve it. Formal mathematics is sometimes dismissed as empty, on the ground that a logical consequence adds nothing to the axioms that imply it — that is what makes it necessary, and it is Leibniz's own point that the opposite of a truth of reasoning is impossible. The dismissal is wrong, and computational mathematics shows why: the equation posits exactly as the axioms do, its solution is implicit but unavailable, and a chain of steps — Newton's method as readily as a proof — constructs what was not there before. Axioms correspond to equation, theorem to solution, proof to algorithm, and nobody says that a computed solution of a differential equation is empty because the equation already determined it.

So the division that matters is not proof against computation. It is *(i) formulate the equation* and *(ii) solve it*. Symbolic deduction and numerical computation are both step (ii), which is why neither is empty and why both are now machine work: systems based on large language models have begun to produce research-level proofs, and they write and run solvers. Step (i) is not machine work, and not because the technology is young — there is nothing to derive a formulation from, so no chain of steps produces it and no proof assistant can check it. Step (i) is moreover what human thinking is good at. The programme built on this division was run at Chalmers University of Technology from 2000 to 2007 as the mathematics course for chemistry students and was terminated in 2008; the claim of this paper is that AI pushes it back to the front line.

The professional product of the pure university mathematician is step (ii) in its symbolic form. Our claim is not that this work loses its value, but that it loses its scarcity, and with it the justification for an educational system organised from the top down around producing people who perform it by hand. Mathematics education is so organised, at every level: school arithmetic and school algebra are a filtered-down residue of a syllabus whose terminal purpose is the formal mathematician. If the terminal purpose is automated, the filtering has nothing left to deliver.

What remains for people is step (i), together with what is needed to judge the output of step (ii): formulating the model, controlling the error, and deciding whether a computed quantity carries information. That is a different curriculum, and it is not a curriculum that has to be invented. It has been written and published over three decades as *Applied Mathematics: Body and Soul* at university level and as *Leibniz World of Math* at school level: every subject presented in both its analytical and its computational form, the derivative as a difference quotient that is actually computed, the integral as a time-stepping algorithm that is actually run, and programming as the notation in which mathematics is written rather than as a subject beside it. We set out what changes at each level, and we state the objections, including the one that this programme made the same proposal before AI existed and did not win the argument.

“There are also two kinds of truths: truth of reasoning and truths of fact. Truths of reasoning are necessary and their opposite is impossible; those of fact are contingent and their opposite is possible.”

— Leibniz, *Monadology*, §33¹

1 Formulate and solve, and why solving is not empty

Mathematics is to formulate an equation and to solve it. That division was the organising structure of a mathematics education programme, *Applied Mathematics: Body and Soul* [7, 8], developed at Chalmers University of Technology and run from 2000 to 2007 as the mathematics course for students of chemistry, and terminated in 2008 when the present author moved to KTH.

The claim of this paper is that AI pushes that programme back to the front line, and for a reason that has nothing to do with its merits having been reassessed: step (ii) can now be carried out in automated form, while step (i) requires human thinking — and is, in fact, what human thinking is good at. A curriculum organised around the half that has been automated is in difficulty whatever anyone decides about it.

Everything below turns on the division, so we begin with the objection which would make step (ii) — the half now being automated — a matter of no consequence.

The objection. Formal mathematics is the drawing of logical consequences from a set of axioms. A consequence is already contained in the axioms — that is what makes it necessary, and it is Leibniz’s own point that the opposite of a truth of reasoning is impossible. So in a way nothing is ever added. The axioms are posited, the theorems are unpacked, and the unpacking cannot introduce content that the positing did not already carry. On this reading a proof is a tautology of record, the subject is empty, and automating it would be automating nothing.

Why it is not correct. The objection confuses what is implied with what is available. A chain of logical steps, or of computational steps, can bring out something that is in no useful sense present at the start: not an addition to the axioms, but a *construction* built on them as a foundation. The chain is where the content appears, because the chain is where the work is, and the length of the chain is not bounded by the length of the axioms. Nothing in the field axioms together with the definition of a polynomial makes a root of a particular quintic available; a construction makes it available.

That is not a rescue of formal mathematics by rhetoric. It is the ordinary situation in computational mathematics, where the same structure can be seen plainly and without controversy.

The equation is the axiom set. Consider solving $f(x) = 0$ by Newton’s method,

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

The equation plays exactly the role of the axioms: it posits, and it determines its solution without exhibiting it. The solution plays the role of the theorem: implicit from the outset,

¹Quoted at the head of [5].

unavailable until constructed. The iteration plays the role of the proof: a chain of steps, each one legitimate, terminating in an object that was not there before. The correspondence is worth setting down,

$$\text{axioms} \longleftrightarrow \text{equation}, \quad \text{theorem} \longleftrightarrow \text{solution}, \quad \text{proof} \longleftrightarrow \text{algorithm},$$

and in this light the emptiness objection refutes itself: nobody says that a computed solution of a differential equation is empty because the equation already determined it.

The computational case also shows something the formal case hides. What Newton's method returns is not the root but an approximation to it, together with a residual $f(x_k)$ measuring how far from the root it stands. The construction is therefore not even in principle a logical consequence of the equation. It is a different object, obtained in finitely many steps at finite precision, and what makes it mathematics rather than a guess is the bound that comes with it. So the last line of the correspondence is not quite an identity — a proof is certified by derivability and an algorithm by an error bound — and that difference is the subject of Section 2.

Why step (ii) stays in the curriculum although it is automated. Automating the execution does not remove the need to understand what the solution process is about. Newton's method is not an arbitrary recipe which happens to work. It is formulated for a reason, and the reason is one symbolic line: replace f near x_k by its tangent and solve the linear equation instead,

$$f(x_k) + f'(x_k)(x - x_k) = 0 \quad \implies \quad x = x_k - \frac{f(x_k)}{f'(x_k)}.$$

That line explains the iteration, and — this is the point — it also settles how the iteration will behave once launched. The term discarded is quadratic, so near a simple root the error is roughly squared at each step, which is the fast convergence observed. The construction divides by $f'(x_k)$, so it is in difficulty wherever the derivative is small. The tangent is a local picture only, so a remote starting point need not converge at all. None of this is learned by watching an automated solver succeed, and all of it is needed by anyone who has to judge the occasions when it fails.

So arguments of simple symbolic form carry a double value which is easy to miss in the present debate: they are at once *understandable and powerful*. Short enough to be followed by a student, and decisive for the behaviour of a computation the student will never execute by hand.

And there are only a few such arguments. The algorithms of this kind which drive stepwise solution processes are not a syllabus but a handful: linearise and iterate, as in Newton's method; iterate a contraction towards a fixed point; step forward in time by $dx = v dt$ and keep the error under control; split a coupled problem into parts and iterate on the coupling. Each rests on a short argument of the sort just given, and each argument both explains the construction and predicts its failure. This is a curricular fact of the first importance, because the part of the formal repertoire which genuinely illuminates computation turns out to be small — and it is not what the years now spent on symbolic technique are spent on.

Education is meant to bring illumination, not mystery. A solver handed to a student as a black box produces mystery; so does a symbolic repertoire drilled with no computation for

it to explain. The pairing removes both, and this is exactly the *Soul* of Body and Soul: the analytical argument which says why a computation is formulated as it is and how it will behave, the *Body* being the computation itself. Automating step (ii) raises the value of that argument rather than lowering it, because the argument is now the only thing standing between the student and an oracle.

So the real division is not proof against computation. It is

(i) formulate the equation and *(ii) solve the equation.*

Step (i) posits; step (ii) constructs what the positing implies. Symbolic deduction and numerical computation are both step (ii) — one carried out in symbols, the other in numbers — and the division was drawn this way long before there was a machine to do either half.

What has happened, in this language. AI does step (ii). It does it in the symbolic form, which is the news of the last two years, and it does it in the numerical form as well, writing and running solvers. Step (i) it does not do, and the reason is not a present limitation of the technology: *there is nothing to derive a formulation from*. An equation is answerable to the world, or to a mathematical purpose, and not to a rule, so no chain of steps produces it and no proof assistant can check it. Given the axioms, the machine; to give the axioms, a person.

That is the asymmetry the rest of this paper builds on, and it is the one that tells a curriculum what to do: teach (i), and teach what is needed to judge the output of (ii).

One concession, which matters for the pure side. Pure mathematics has a step (i) of its own — choosing the axioms, framing the definitions, deciding which question is worth asking — and that is not automated either. We do not claim it is in danger. What is in danger is the scarcity of step (ii), which is the part the training is mostly spent on and the part the examinations mostly test.

2 Two kinds of truth, and which form of solving has been automated

Step (ii) comes in two forms which are not certified in the same way, and Leibniz's distinction, quoted above, is what separates them.

A truth of reasoning is established by derivation. Hypotheses are stated, rules are applied, and the conclusion follows; whether it follows is a decidable question, and a proof assistant decides it. Nothing outside the argument is consulted, and the verdict is binary: the proof is valid or it is not, and no quantity attaches to the answer.

A truth of fact is established otherwise. What does this flow do; what is the drag on this body; what is the binding energy of this molecule. The question has a definite answer, the answer is contingent — the world could have been otherwise — and it is reached by computing what a model predicts and comparing that with a measurement. Here the verdict is not binary but quantitative: a number with a stated accuracy, which can agree with an experiment or fail to.

The two are certified differently, and this is the whole of the matter. Derivability certifies the first and is checkable step by step. An error bound certifies the second, and the computation produces it alongside the answer. Both are constructions in the sense of Section 1 — both are step (ii) — but one is answerable only to its own hypotheses while the other is answerable to a measurement. Step (i) is certified by neither, which is precisely why it cannot be delegated: a formulation is answerable to the world, and there is nothing it could be derived from.

What machines have done. In the last two years, systems based on large language models have produced research-level mathematics of the first kind: analytical arguments, in some cases formalised in Lean, including substantial work on partial differential equations [1, 2]. One may argue about how much of it is new and how much recombined. The direction is not in doubt, and neither is the economics: derivation is symbol manipulation, symbol manipulation is what these systems do, and a proof assistant removes the need for a human to referee the output.

What the same systems have *not* done is reach the objects of the second kind. The record of their mathematical output contains almost no conventional numerical partial differential equations — no finite elements, no adaptive computation, no error control, no computational turbulence [3]. The reason is structural rather than incidental. Symbolic computation reaches objects that possess exploitable structure — a symmetry, an ansatz, a closed form — and reaches them exactly; it does not reach objects defined by the absence of such structure, of which turbulence is the standing example, and no increase in scale alters this [4].

So the automation is not uniform even within step (ii). It has fallen, first and hardest, on the symbolic form — which is precisely the form the university has organised itself to teach.

3 Whose mission this was

The professional product of the pure university mathematician is the theorem: a truth of reasoning, written for other mathematicians, certified by derivability. That is what the training produces, what the publication system circulates, and what appointment and promotion reward.

We are not claiming that this work ceases to be valuable, and we are not predicting that mathematics departments close. We are claiming something narrower and harder to escape: the work loses its *scarcity*. A skill retains its market when it is both useful and rare. The human computers of the 1940s — the people whose profession was arithmetic — were not refuted by the digital computer; arithmetic remained as necessary as before. The profession ended anyway. Nobody now considers it a loss to mathematics, because arithmetic was never the mathematics; it was the labour.

The question this raises for education is not whether mathematics survives. It is which part of the present curriculum was labour.

The filtering structure, which is why this reaches every level. Mathematics education is organised from the top down. What a mathematics department considers the subject determines the teacher-training syllabus; that determines the school curriculum; and that determines what a seven-year-old does with a pencil. The content thins as it descends but its shape is preserved: a repertoire of by-hand symbolic technique, examined by reproducing that technique, justified at each level by what the next level up requires.

This is why the automation of formal mathematics is a problem for school arithmetic and not only for the graduate seminar. The terminal purpose of the filtering — the person who produces truths of reasoning by hand — is the part now being automated. A pipeline keeps its shape for a while after its destination has gone, and the shape is what is visible in a Swedish syllabus in which the four arithmetic operations remain the *ledstjärna* for nine years, with *taluppfattning* consuming most of the time, and minimal progression across grades — a programme fairly described as “*ineffektivt och intressedödande*” [6].

That criticism was made before AI, on its own merits. What AI changes is that the defence has collapsed. One could previously answer that the by-hand repertoire, however joyless, was the entrance fee to a mathematical career. It is now the entrance fee to a competition with a machine that performs the repertoire faster, at any hour, for nothing.

4 What is left for people, stated concretely

By Section 1 the question has a sharp form. Step (ii) is going; what in step (i), and in the judgement of step (ii)’s output, requires a person who is answerable for the result? The honest form is not “what can AI not do”, since that boundary moves.

The research loop on the side of truths of fact runs

hypothesis $\rightarrow$ computation $\rightarrow$ error and stability analysis $\rightarrow$ interpretation $\rightarrow$ new computation,

and AI can already operate large parts of it: write the solver, run the parameter sweep, refine the mesh, plot the result [3]. We concede this fully, and it is the strongest objection to any reform that merely replaces proof with programming: the body of the subject is as automatable as its soul, because both are step (ii).

Three things in that loop are nevertheless not delegable, and they are curricular content.

Formulating the model. Which differential equation, on which domain, with which boundary conditions, representing which physical question. This is a judgement about correspondence between mathematics and world, and it is answerable to the world, not to a rule. It cannot be certified by derivability, because there is nothing to derive it from.

Controlling the error. A computed output is worth exactly what is known about its accuracy. A finite element computation returns a bound on how wrong its output may be, computed from the residual weighted by an adjoint solution; this is the mechanism by which a computation becomes communicable to someone who did not perform it. The problem AI now faces — an output too long to check and too consequential to trust blindly — is the problem numerical analysis met fifty years ago and answered in this way [4]. A student who cannot estimate an error cannot tell a simulation from a picture.

Judging whether the output means anything. In turbulent flow the detailed solution is not unique and does not determine anything; mean-value outputs such as drag and lift are reproducibly computable and do. Knowing which quantity in a computation carries information and which does not is the central skill of the subject, and it is not symbol manipulation at all.

These three are what a curriculum built for truths of fact would teach. The present curriculum teaches almost none of them, and devotes years to the repertoire that has just been automated.

5 The reform is already written

The proposal we are making is not new, which we regard as its principal evidence rather than as an embarrassment, and we return to the objection in Section 7.

University level: Body and Soul. Developing analysis and computation together, rather than treating computation as a service added afterwards, has been pursued as a programme for three decades under the name *Applied Mathematics: Body and Soul* [7, 8], with Volumes I–IV published by Springer between 2003 and 2009. The name is the thesis: the body of a mathematical subject is its computational content and the soul its analytical content, and a subject that has lost either is crippled. Volumes I–III carry this through calculus and linear algebra, each subject presented in both forms.

It was not only written but taught. The programme ran at Chalmers University of Technology from 2000 to 2007 as the mathematics course for students of chemistry — seven years of undergraduate delivery, which is a longer trial than most reform proposals receive — and was discontinued in 2008 when the present author moved to KTH. We record the termination because it is evidence of the right kind: what ended the programme was not a finding about the students but the departure of the person teaching it, and a curriculum that survives only while its author is in the building has not yet been institutionally adopted. That is the position this paper argues AI has now changed. Volume IV takes the further step of putting the computational form forward as the mathematical model of physics itself [9]; Volumes V and VI [10, 11] carry the same pairing from macroscopic flow to atomic structure, with statistical aggregation replaced by deterministic field equations at both ends.

School level: Leibniz World of Math. The corresponding school programme, published online in three levels — Pre, School and Pro [5] — replaces symbolic calculus by constructive, digital calculus. Its content is the opposite of a thinning: the elementary level is claimed to cover more than the Swedish gymnasium syllabus, and the secondary level more than a typical undergraduate one [6]. Three of its principles are worth stating because they are concrete and testable, not slogans.

Programming is part of mathematics, not a subject beside it. Code is the notation in which a mathematical construction is written down. Among the first things a child does is program a calculator for the four operations — “*elever själv programmerar en räknedosa för de fyra räknesätten under tidiga skolår*” — which both explains the operations and frees the nine years that manual drill now consumes [6].

Calculus is time-stepping. The core of the subject is

$$dx = v dt, \quad dv = a dt,$$

a construction that can be computed, watched, and refined, and from which the derivative and the integral arise as what the construction converges to. The existence proof and the algorithm

are the same object. A pupil who has stepped this scheme has met the content of calculus; one who has memorised a table of antiderivatives has met its labour.

The same things are done at every age, as in music. A six-year-old and a graduate student work on the same material at different complexity, exactly as both play the violin. This removes the filtering structure of Section 3 at a stroke, because nothing is being saved up for a later stage.

And the professional level follows the same four steps. At the Pro level the method is stated plainly: formulate the model as a differential equation; replace derivatives by difference quotients on a mesh to obtain an algebraic system; express that system in code; run it. FEniCS, which generates compiled solver code directly from a variational form including the adjoint problem required for error control [12], is the demonstration that the last two steps are themselves automatable — which is the point. The automation of the labour is what leaves room for the mathematics.

6 What changes, level by level

School. Arithmetic is kept as understanding and dropped as drill: the child programs the operations rather than performing them for nine years. Function means algorithm, not formula. Geometry is constructed and drawn by code. The derivative appears as a difference quotient that is evaluated, and the differential equation as a scheme that is stepped. The time released is not nothing: it is most of the present syllabus.

Upper secondary. Modelling becomes the subject: mass–spring systems, population models, wave propagation, written as differential equations, discretised, coded and run, with convergence under refinement checked by the pupil rather than asserted by the teacher. Error is introduced here, as a quantity to be estimated, because this is the age at which a simulation first becomes persuasive enough to be dangerous.

Undergraduate. Calculus and linear algebra in both forms, as in Body and Soul: the integral as the solution of $du/dx = f$ constructed by a scheme whose convergence is proved, so that the analysis and the computation are one course and not two. Stability, residuals and a posteriori error estimation move from a specialised later course into the first one. The by-hand symbolic repertoire shrinks to what is needed to see structure, and that need is real, permanent and much smaller than the present examination: the handful of arguments of Section 1 which explain why each algorithm is formulated as it is and when it will fail.

Graduate and research. The computational-mathematician loop of Section 4, with AI operating it and the student responsible for the parts that are answerable to the world. The thesis question becomes: what did you compute, how wrong might it be, and what does it mean.

Assessment, which is forced and not optional. An examination that a language model passes outright measures nothing about the candidate. This is now true of most of the written

mathematics examinations in use, and no change in invigilation repairs it, because the content and not the procedure is what has been automated. Assessment has to move to what remains: formulate a model, implement it, show convergence, bound the error, state what the result establishes and what it does not. That is harder to mark and impossible to automate, and it is also, not coincidentally, the only part of the examination that resembles the work.

7 Objections

“This is self-serving: you proposed this before AI, and you are now claiming vindication.” Correct on the facts, and it cuts both ways. The programme was not designed for this moment, which is some evidence that it was not designed to exploit it; and it was put to the profession once already, on its own merits — taught for seven years at Chalmers and then discontinued — and did not win the argument. We do not claim the argument was won; we claim the conditions under which it was lost have gone. The case against the reform was that the by-hand symbolic repertoire remained the foundation of a mathematical career, and that computation was an application to be added later by engineers. The first half of that is what has been automated. If the reform is adopted now it will not be because the argument improved.

“AI will automate the computational side too, so your curriculum is next.” We concede this and have built it into the proposal — it is the content of Section 1, where proving and computing turn out to be the same step. AI already writes solvers and runs sweeps better than a student will, and we expect that to continue. What it does not supply is the warrant — the model’s correspondence to the world and the bound on the computed output — and a warrant is exactly what must be produced by someone who can be held answerable. A curriculum that taught programming instead of proof, without error control and without modelling judgement, would deserve the objection entirely.

“You are proposing to produce people who can run a solver and cannot tell when it lies.” This is the real failure mode, and it is an argument for more analysis rather than less. Stability, residual, convergence rate and adjoint sensitivity are analytical content; they are the soul half of the name, and they are what distinguishes the programme from a coding course. A reform that dropped them would reproduce the failure in the opposite direction from the present one.

“Formal proof teaches rigour, and nothing else does.” It teaches one kind of rigour, and an important one, which is why the symbolic repertoire shrinks in Section 6 rather than vanishing. But an error bound is also rigour, and of a kind the formal curriculum never teaches: it is answerable to a measurement, whereas a derivation is answerable only to its own hypotheses. A student who has only ever met the second kind has no defence against a model that is internally impeccable and wrong about the world. And the symbolic arguments we keep are kept for illumination rather than for ritual: they are what tells a student why an algorithm works and where it breaks, which is the one question an automated solver cannot be asked.

“The claim that an elementary course can cover more than the gymnasium syllabus is not credible.” It is a strong claim, it is made explicitly [6], and it is the kind of claim that should be settled by trial rather than by argument — which is the next paragraph.

8 What would settle it

A proposal about education is worth what its failure conditions are worth, so we state them.

The programme would be *supported* if pupils taught this way can, at an earlier age than the present syllabus reaches, set up and run a differential-equation model and say how accurate it is; if the released drill time is demonstrably recovered as content rather than lost; and if these pupils are not worse at the symbolic manipulation that remains, having met it in a constructive form.

It would be *refuted*, or confined, if the symbolic repertoire turns out to be the mechanism by which mathematical structure is perceived at all, so that removing the drill removes the capacity — a possibility we cannot exclude on present evidence; or if pupils taught constructively acquire fluency with code and no ability to judge an output, which would mean the soul half had not in fact been transmitted; or if the assessment problem proves insoluble, since a curriculum that cannot be examined will not be adopted whatever its merits.

And what will not settle it. Comparative performance on existing examinations. Those examinations test the automated repertoire, so the present curriculum will win them, and the result will mean nothing. This is the methodological trap in every evaluation of a mathematics reform, and it should be named in advance.

9 Conclusion

Mathematics is to formulate an equation and to solve it. Solving is a construction from what has been posited — which is why it is not empty, the old objection notwithstanding, and why it is also exactly the kind of thing a machine does. Formulating is answerable to the world rather than to a rule, so there is nothing for a machine to derive it from. That is the division, and it was drawn before there was a machine to fall on either side of it.

Leibniz’s two kinds of truth then say which form of solving has gone first. Truths of reasoning are constructed in symbols and certified by derivability, and that is the form now being automated at research level. Truths of fact are constructed in numbers and certified by a bound the computation itself produces, compared in the end against a measurement — and they are the harder half to reach, because the objects that matter there are defined by the absence of the structure symbolic methods exploit.

And step (i) is not merely what is left over. It is what human thinking is good at: deciding what question to ask, what to idealise away, and which equation would answer it. That is the one part of the subject whose automation nobody has demonstrated and the part a curriculum has been least willing to examine.

Mathematics education is organised, at every level, around producing people who carry out step (ii) in its symbolic form by hand. That is the part a machine now does. The part that is

left is the part the present curriculum barely teaches: formulating the model, computing the consequence, bounding the error, and judging what it means.

The name of the older programme was always its thesis: the body of a mathematical subject is its computational content, the soul its analytical content, and a subject that has lost either is crippled. For three decades the argument for pairing them was that each is better with the other. The argument now is stronger and less comfortable. One half has been taken over by machines. A curriculum that teaches only that half is teaching the labour and not the mathematics, and it will be found out quickly, by its own students.

A programme that ran for seven years and was discontinued in 2008 is therefore back on the front line, not because it has been vindicated by argument, but because the division of labour it was built on has become the actual division of labour.

References

- [1] OpenAI, *Finite time blowup for Navier–Stokes*, preprint, 2026. <https://cdn.openai.com/pdf/32d9f210-8b73-45e0-91bc-82a30aef8a9a/navier-stokes.pdf>
- [2] OpenAI, *Finite time blowup for the Euler equation*, preprint, 2026.
- [3] C. Johnson, *AI as Computational Mathematician Rather Than Theorem Prover*, 2026.
- [4] C. Johnson, *AI versus Numerical Analysis: Two Forms of Computation*, 2026.
- [5] C. Johnson, *Leibniz World of Math* (Pre — School — Pro), <https://mathsimulationtechnology.wordpress.com>.
- [6] C. Johnson, *Leibniz vs Ny Kursplan*, <https://mathsimulationtechnology.wordpress.com/digimat-vs-ny-laroplan/>.
- [7] The Body and Soul project, www.bodysoulmath.org.
- [8] K. Eriksson, D. Estep and C. Johnson, *Applied Mathematics: Body and Soul*, vols. 1–3, Springer, 2003–2004.
- [9] J. Hoffman and C. Johnson, *Computational Turbulent Incompressible Flow*, Applied Mathematics: Body and Soul, vol. 4, Springer, 2007.
- [10] C. Johnson, *Computational Thermodynamics*, Applied Mathematics: Body and Soul, vol. 5, draft.
- [11] C. Johnson, *Real Quantum Mechanics*, Applied Mathematics: Body and Soul, vol. 6, draft, <https://claes542.github.io/RealMolecule/realquantum3.pdf>.
- [12] A. Logg, K.-A. Mardal and G. N. Wells (eds.), *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, Springer, 2012.