AI is a tool

A working mathematician's experience and perspective

Martin Raum

July 28, 2026

Chalmers University of Technology
Gothenburg, Sweden

Starting point

My background

Pure mathematician: Analysis, alg. geometry, number theory, and combinatorics.

My background

Pure mathematician: Analysis, alg. geometry, number theory, and combinatorics.

Diplom at RWTH Aachen, PhD at MPI Bonn, postdoc at ETH Zurich, full professor at Chalmers in Gothenburg.

My background

Pure mathematician: Analysis, alg. geometry, number theory, and combinatorics.

Diplom at RWTH Aachen, PhD at MPI Bonn, postdoc at ETH Zurich, full professor at Chalmers in Gothenburg.

At Chalmers I teach high-performance computing.

Toolchain before AI

Working environment and toolchains refined over decades.

Toolchain before AI

Working environment and toolchains refined over decades.

Terminals controlled through a multiplexer (zellij): makes most graphical interfaces unnecessary.

Toolchain before AI

Working environment and toolchains refined over decades.

Terminals controlled through a multiplexer (zellij): makes most graphical interfaces unnecessary.

NeoVim for editing: Quick editing of half-good results allows earlier adoption of AI.

Toolchain before AI

Working environment and toolchains refined over decades.

Terminals controlled through a multiplexer (zellij): makes most graphical interfaces unnecessary.

NeoVim for editing: Quick editing of half-good results allows earlier adoption of AI.

Plain-text and markdown files for documents and notes: Markdown has become the default format for AI.

Toolchain before AI

Working environment and toolchains refined over decades.

Terminals controlled through a multiplexer (zellij): makes most graphical interfaces unnecessary.

NeoVim for editing: Quick editing of half-good results allows earlier adoption of AI.

Plain-text and markdown files for documents and notes: Markdown has become the default format for AI.

git for version control: make changes inspectable.

Requirements for AI

Preserve independence from a particular vendor: technological independence has always been a strong value.

Requirements for AI

Preserve independence from a particular vendor: technological independence has always been a strong value.

Keep data access and side effects within explicit permissions: system integrity and data protection may not be jeopardized.

Requirements for AI

Preserve independence from a particular vendor: technological independence has always been a strong value.

Keep data access and side effects within explicit permissions: system integrity and data protection may not be jeopardized.

Mathematical correctness: There is no excuse for making more mistakes than before AI.

AI's key challenge

Key challenge around AI is not a technical one.

AI's key challenge

Key challenge around AI is not a technical one.

Kontorovich

Veritasium on Bounded Gaps between Primes, Jun 14, 2026:

> *To me, this is a really interesting aspect of mathematics in particular [...] where we have a truly honest approach to success [...]. He sent in an argument, [...] they gave him the time that the argument was due.*

AI's key challenge

Key challenge around AI is not a technical one.

Kontorovich

Veritasium on Bounded Gaps between Primes, Jun 14, 2026:

> *To me, this is a really interesting aspect of mathematics in particular [...] where we have a truly honest approach to success [...]. He sent in an argument, [...] they gave him the time that the argument was due.*

An anti-AI movement is questioning this principle. Ideas are judged by their origin and the tools used.

AI strategy

Nonetheless, we focus on a technological narrative.

For a current task or project, I ask:

How can current AI help to

raise the quality of the outcome or speed up the process?

Highlight developments in 2026

Jan.: First use of agent harnesses to improve system infrastructure (Python).

Highlight developments in 2026

Jan.: First use of agent harnesses to improve system infrastructure (Python).

May: Sandboxed, multi-pass workflow for reviewing mathematical manuscripts.

Highlight developments in 2026

Jan.: First use of agent harnesses to improve system infrastructure (Python).

May: Sandboxed, multi-pass workflow for reviewing mathematical manuscripts.

May: First complete AI-composed research paper, followed by manual rewrite.

Highlight developments in 2026

Jan.: First use of agent harnesses to improve system infrastructure (Python).

May: Sandboxed, multi-pass workflow for reviewing mathematical manuscripts.

May: First complete AI-composed research paper, followed by manual rewrite.

July: Discovery and search for literature and topic exploration. Complete drafting of research plans.

Early adoption of AI

Tailor-made early tooling

Access almost exclusively through OpenRouter (API); towards the end, a Claude subscription.

Tailor-made early tooling

Access almost exclusively through OpenRouter (API); towards the end, a Claude subscription.

Nov. 2024: Implement facileLLM, an AI interface for NeoVim, to avoid web GUI. Alternatives started to arise only mid 2025.

Tailor-made early tooling

Access almost exclusively through OpenRouter (API); towards the end, a Claude subscription.

Nov. 2024: Implement facileLLM, an AI interface for NeoVim, to avoid web GUI. Alternatives started to arise only mid 2025.

May 2025: Use and study deep-research agent architectures, both prompt design and knowledge representation.

Move to standard components

Aug. 2025: Attended AI workshop at MathFest; only then understood how far behind the community was.

Move to standard components

Aug. 2025: Attended AI workshop at MathFest; only then understood how far behind the community was.

Sep. 2025: First large course after students adopted AI; understood how little they knew about AI-supported learning.

Move to standard components

Aug. 2025: Attended AI workshop at MathFest; only then understood how far behind the community was.

Sep. 2025: First large course after students adopted AI; understood how little they knew about AI-supported learning.

Oct. 2025: facileLLM has fallen behind newer alternatives. Switch to CodeCompanion and Simon Willison's LLM-CLI. Integrate LinkUp MCP.

Move to standard components

Aug. 2025: Attended AI workshop at MathFest; only then understood how far behind the community was.

Sep. 2025: First large course after students adopted AI; understood how little they knew about AI-supported learning.

Oct. 2025: facileLLM has fallen behind newer alternatives. Switch to CodeCompanion and Simon Willison's LLM-CLI. Integrate LinkUp MCP.

Dec. 2025: Persistent prompt design for language learning role play.

Move to standard components

Aug. 2025: Attended AI workshop at MathFest; only then understood how far behind the community was.

Sep. 2025: First large course after students adopted AI; understood how little they knew about AI-supported learning.

Oct. 2025: facileLLM has fallen behind newer alternatives. Switch to CodeCompanion and Simon Willison's LLM-CLI. Integrate LinkUp MCP.

Dec. 2025: Persistent prompt design for language learning role play.

Jan. 5, 2026: First work with agent harnesses.

Pre-harness applications

Topic exploration and learning outside my profession. Variety of topics, but almost no mathematics.

Pre-harness applications

Topic exploration and learning outside my profession. Variety of topics, but almost no mathematics.

Very little programming: not least because facileLLM never had a suitable interface for this.

Pre-harness applications

Topic exploration and learning outside my profession. Variety of topics, but almost no mathematics.

Very little programming: not least because facileLLM never had a suitable interface for this.

Text work as a writing partner: E.g. 8h writing session with sustained feedback, iterating on content and polishing text.

Initial harness workflow

First use of harnesses

A harness lets a model inspect files and run tools inside a project, driven by agent loops and specialized system prompts.

Jan. 2026: OpenCode with a GitHub Copilot subscription (reacting to Anthropic's lock-in). Vendor independence had delayed harness adoption.

First use of harnesses

A harness lets a model inspect files and run tools inside a project, driven by agent loops and specialized system prompts.

Jan. 2026: OpenCode with a GitHub Copilot subscription (reacting to Anthropic's lock-in). Vendor independence had delayed harness adoption.

Jan. 5, 2026: System infrastructure (Python), with little mathematics involved.

Jan. 6 and 7, 2026: Custom arXiv summaries via embedding techniques.

First harness use for mathematics

Jan. to Feb. 2026: Complete rewrite of lecture notes for algebraic number theory.

First harness use for mathematics

Jan. to Feb. 2026: Complete rewrite of lecture notes for algebraic number theory.

Reliable generation, since material was standard; I could also recognize potential mistakes easily.

First harness use for mathematics

Jan. to Feb. 2026: Complete rewrite of lecture notes for algebraic number theory.

Reliable generation, since material was standard; I could also recognize potential mistakes easily.

Still using LLM-CLI as a home-brew subagent. No skills. LinkUp MCP to compensate for mediocre Copilot search.

Mathematical manuscript review

Review of mathematical writing among most time consuming parts of profession: By standards of the field, do everything reasonable to avoid mistakes.

Mathematical manuscript review

Review of mathematical writing among most time consuming parts of profession: By standards of the field, do everything reasonable to avoid mistakes.

Classical review is effective but expensive:

- leave a manuscript untouched for months,

Mathematical manuscript review

Review of mathematical writing among most time consuming parts of profession: By standards of the field, do everything reasonable to avoid mistakes.

Classical review is effective but expensive:

- leave a manuscript untouched for months,
- return with enough distance to notice mistakes,

Mathematical manuscript review

Review of mathematical writing among most time consuming parts of profession: By standards of the field, do everything reasonable to avoid mistakes.

Classical review is effective but expensive:

- leave a manuscript untouched for months,
- return with enough distance to notice mistakes,
- pay the cognitive cost of reconstructing the whole topic.

Mathematical manuscript review

Review of mathematical writing among most time consuming parts of profession: By standards of the field, do everything reasonable to avoid mistakes.

Classical review is effective but expensive:

- leave a manuscript untouched for months,
- return with enough distance to notice mistakes,
- pay the cognitive cost of reconstructing the whole topic.

AI offers a way to make review stricter and more continuous. Also synergizes well with teaching, which requires feedback generation.

Initial review prompts

Mar. 2026: In the harness:

- expose manuscript as one (well-structured) tex file,

Initial review prompts

Mar. 2026: In the harness:

- expose manuscript as one (well-structured) tex file,
- request review and numerical verification and tests,

Initial review prompts

Mar. 2026: In the harness:

- expose manuscript as one (well-structured) tex file,
- request review and numerical verification and tests,
- instruct to work paragraph by paragraph,

Initial review prompts

Mar. 2026: In the harness:

- expose manuscript as one (well-structured) tex file,
- request review and numerical verification and tests,
- instruct to work paragraph by paragraph,
- write review artifacts and summary to markdown files (similar to ADRs in programming).

Initial review prompts

Mar. 2026: In the harness:

- expose manuscript as one (well-structured) tex file,
- request review and numerical verification and tests,
- instruct to work paragraph by paragraph,
- write review artifacts and summary to markdown files (similar to ADRs in programming).

Review takes about 2 hours. Work through feedback manually item by item.

Formatting and clean code as a foundation

End of Mar. 2026: Observation that better formatted code yields better results; context window becomes an issue (GitHub Copilot offers a very short window).

Formatting and clean code as a foundation

End of Mar. 2026: Observation that better formatted code yields better results; context window becomes an issue (GitHub Copilot offers a very short window).

- Split sections into separate tex files;
- Introduce latexindent into toolchain;
- File map and LaTeX style guide in AGENTS.md.

Formatting and clean code as a foundation

End of Mar. 2026: Observation that better formatted code yields better results; context window becomes an issue (GitHub Copilot offers a very short window).

- Split sections into separate tex files;
- Introduce latexindent into toolchain;
- File map and LaTeX style guide in AGENTS.md.

Break until end of April to focus on a math project.

Iterated review workflow

Review process outgrows AGENTS

Review conducted via a separately stored prompt.

Review process outgrows AGENTS

Review conducted via a separately stored prompt.

AGENTS.md contains a LaTeX style guide.

Review process outgrows AGENTS

Review conducted via a separately stored prompt.

AGENTS.md contains a LaTeX style guide.

This hampers performance during verification in Julia.

Review process outgrows AGENTS

Review conducted via a separately stored prompt.

AGENTS.md contains a LaTeX style guide.

This hampers performance during verification in Julia.

End of Apr. 2026: Note and document ingestion reveals further problems with context management and single model performance.

Review process outgrows AGENTS

Review conducted via a separately stored prompt.

AGENTS.md contains a LaTeX style guide.

This hampers performance during verification in Julia.

End of Apr. 2026: Note and document ingestion reveals further problems with context management and single model performance.

The simplicity and convenience of AGENTS.md no longer outweighed the negative consequences.

Components of the refactored review process

Separate components:

1. Summary: Manuscript and section summary and purpose;

Components of the refactored review process

Separate components:

1. Summary: Manuscript and section summary and purpose;
2. Project map: files, build commands, and local constraints;

Components of the refactored review process

Separate components:

1. Summary: Manuscript and section summary and purpose;
2. Project map: files, build commands, and local constraints;
3. Agent: reviewer role with explicit inputs and outputs;

Components of the refactored review process

Separate components:

1. Summary: Manuscript and section summary and purpose;
2. Project map: files, build commands, and local constraints;
3. Agent: reviewer role with explicit inputs and outputs;
4. Skill: reusable LaTeX style guide;

Components of the refactored review process

Separate components:

1. Summary: Manuscript and section summary and purpose;
2. Project map: files, build commands, and local constraints;
3. Agent: reviewer role with explicit inputs and outputs;
4. Skill: reusable LaTeX style guide;
5. Artifacts: persistent review notes.

Components of the refactored review process

Separate components:

1. Summary: Manuscript and section summary and purpose;
2. Project map: files, build commands, and local constraints;
3. Agent: reviewer role with explicit inputs and outputs;
4. Skill: reusable LaTeX style guide;
5. Artifacts: persistent review notes.

This opens the door to extensions and refinement.

Review pipeline

Mid May 2026: Refinement of reviewer agent into subagents with associated skills.

Review pipeline

Mid May 2026: Refinement of reviewer agent into subagents with associated skills.

- Index the manuscript;
- Run independent section reviews;
- Synthesize findings; Adjudicate issues;
- Perform final adversarial review.

Review pipeline

Mid May 2026: Refinement of reviewer agent into subagents with associated skills.

- Index the manuscript;
- Run independent section reviews;
- Synthesize findings; Adjudicate issues;
- Perform final adversarial review.

Separate reviews of architecture, mathematics, citations, verification, and language.

Review pipeline

Mid May 2026: Refinement of reviewer agent into subagents with associated skills.

- Index the manuscript;
- Run independent section reviews;
- Synthesize findings; Adjudicate issues;
- Perform final adversarial review.

Separate reviews of architecture, mathematics, citations, verification, and language.

Separation is part of context management, reveals omissions, and allows faster execution.

Scoping towards manual review

The review restricts scope:

- the orchestrator reads no manuscript sections;
- each reviewer reads exactly one section;
- every issue carries a file, line, anchor, and quotation.

Scoping towards manual review

The review restricts scope:

- the orchestrator reads no manuscript sections;
- each reviewer reads exactly one section;
- every issue carries a file, line, anchor, and quotation.

The manuscript stays read-only; findings go to separate reports.

Scoping towards manual review

The review restricts scope:

- the orchestrator reads no manuscript sections;
- each reviewer reads exactly one section;
- every issue carries a file, line, anchor, and quotation.

The manuscript stays read-only; findings go to separate reports.

This approach costs a lot of tokens and does not work well with caching.

Scoping towards manual review

The review restricts scope:

- the orchestrator reads no manuscript sections;
- each reviewer reads exactly one section;
- every issue carries a file, line, anchor, and quotation.

The manuscript stays read-only; findings go to separate reports.

This approach costs a lot of tokens and does not work well with caching. Once more detailed reviews are performed this is less relatively less relevant.

Example: Claim-level inspection

Within section scope, the mathematics reviewer examines:

- claims, hypotheses, and normalizations;

Example: Claim-level inspection

Within section scope, the mathematics reviewer examines:

- claims, hypotheses, and normalizations;
- gaps, signs, quantifiers, and edge cases;

Example: Claim-level inspection

Within section scope, the mathematics reviewer examines:

- claims, hypotheses, and normalizations;
- gaps, signs, quantifiers, and edge cases;
- opportunities for numerical or algebraic tests.

Example: Verification evidence

A review item should lead to one of four outcomes:

- a reproducible computation,

Example: Verification evidence

A review item should lead to one of four outcomes:

- a reproducible computation,
- a precise source comparison,

Example: Verification evidence

A review item should lead to one of four outcomes:

- a reproducible computation,
- a precise source comparison,
- a formalization (not yet integrated into toolchain), or

Example: Verification evidence

A review item should lead to one of four outcomes:

- a reproducible computation,
- a precise source comparison,
- a formalization (not yet integrated into toolchain), or
- an explicit statement that the concern remains unresolved.

Example: Verification evidence

A review item should lead to one of four outcomes:

- a reproducible computation,
- a precise source comparison,
- a formalization (not yet integrated into toolchain), or
- an explicit statement that the concern remains unresolved.

The report records the evidence behind each conclusion.

Human post-review tasks

The system produces detailed criticism based on which I decide:

- which objections are mathematically sound ($\approx 20\%$ false positive);

Human post-review tasks

The system produces detailed criticism based on which I decide:

- which objections are mathematically sound ($\approx 20\%$ false positive);
- which explanations require more detail (AI more verbose than mathematical convention);

Human post-review tasks

The system produces detailed criticism based on which I decide:

- which objections are mathematically sound ($\approx 20\%$ false positive);
- which explanations require more detail (AI more verbose than mathematical convention);
- which ideas deserve emphasis (AI serves as a brainstorming tool).

Human post-review tasks

The system produces detailed criticism based on which I decide:

- which objections are mathematically sound ($\approx 20\%$ false positive);
- which explanations require more detail (AI more verbose than mathematical convention);
- which ideas deserve emphasis (AI serves as a brainstorming tool).

Almost no experience with AI-based resolution of reports.

Harness infrastructure

Multi-harness setup

End May, 2026: GitHub announced its rug pull; switch to DeepSeek V4 to limit costs.

Multi-harness setup

End May, 2026: GitHub announced its rug pull; switch to DeepSeek V4 to limit costs.

In reaction to opaque AI-generated results, work on a full description of AI-composed research.

Multi-harness setup

End May, 2026: GitHub announced its rug pull; switch to DeepSeek V4 to limit costs.

In reaction to opaque AI-generated results, work on a full description of AI-composed research.

One co-author did not want to have Chinese models mentioned in the paper.

Multi-harness setup

End May, 2026: GitHub announced its rug pull; switch to DeepSeek V4 to limit costs.

In reaction to opaque AI-generated results, work on a full description of AI-composed research.

One co-author did not want to have Chinese models mentioned in the paper.

Bootstrap scripts for Codex, ClaudeCode, and OpenCode to evade vendor lock-in under subscriptions.

Multi-harness setup

End May, 2026: GitHub announced its rug pull; switch to DeepSeek V4 to limit costs.

In reaction to opaque AI-generated results, work on a full description of AI-composed research.

One co-author did not want to have Chinese models mentioned in the paper.

Bootstrap scripts for Codex, ClaudeCode, and OpenCode to evade vendor lock-in under subscriptions.

Drawback: This removes the possibility to optimize agents and skills for individual models and harnesses.

Harness isolation

In default setups, harnesses can read and write all user data.

- System integrity: Danger of "classical" rm −rf /.
- Context management: Harnesses escape intended working directories to see shadowed information.
- Exfiltration: ClaudeCode known to send system information via steganography.

Harness isolation

In default setups, harnesses can read and write all user data.

- System integrity: Danger of "classical" rm −rf /.
- Context management: Harnesses escape intended working directories to see shadowed information.
- Exfiltration: ClaudeCode known to send system information via steganography.

Sandbox via bubblewrap:

- working directory only available user data;
- selected system data available for read.

Exposing material to AI

Folder infrastructure classifies information; For instance:

Exposing material to AI

Folder infrastructure classifies information; For instance:

- “llm” contains LLM investigations and artifacts.

Exposing material to AI

Folder infrastructure classifies information; For instance:

- “llm” contains LLM investigations and artifacts.
- “external_material” contains external material.
- “communication” contains project-related correspondence.

Exposing material to AI

Folder infrastructure classifies information; For instance:

- "llm" contains LLM investigations and artifacts.
- "external_material" contains external material.
- "communication" contains project-related correspondence.
- "handwritten" contains handwritten notes and annotated notes and papers.

Exposing material to AI

Folder infrastructure classifies information; For instance:

- "llm" contains LLM investigations and artifacts.
- "external_material" contains external material.
- "communication" contains project-related correspondence.
- "handwritten" contains handwritten notes and annotated notes and papers.
- "transcriptions" contains transcriptions of external material, handwritten notes, and annotated work.

Exposing material to AI

Folder infrastructure classifies information; For instance:

- "llm" contains LLM investigations and artifacts.
- "external_material" contains external material.
- "communication" contains project-related correspondence.
- "handwritten" contains handwritten notes and annotated notes and papers.
- "transcriptions" contains transcriptions of external material, handwritten notes, and annotated work.
- "cite" contains Markdown summaries of cited statements.
- "code" contains test and verification code.

Parallel agent work

Early July, 2026: Git worktrees for parallel agents on same project. This requires isolation.

Parallel agent work

Early July, 2026: Git worktrees for parallel agents on same project. This requires isolation.

Interaction with folder infrastructure not fully solved.

Summary

Key components of successful AI work

- Know your process: describe it to AI.
- Infrastructure and system engineering.
- Knowledge management: exposure and shadowing.
- Reserve ideas, emphasis, and final judgment for the mathematician.