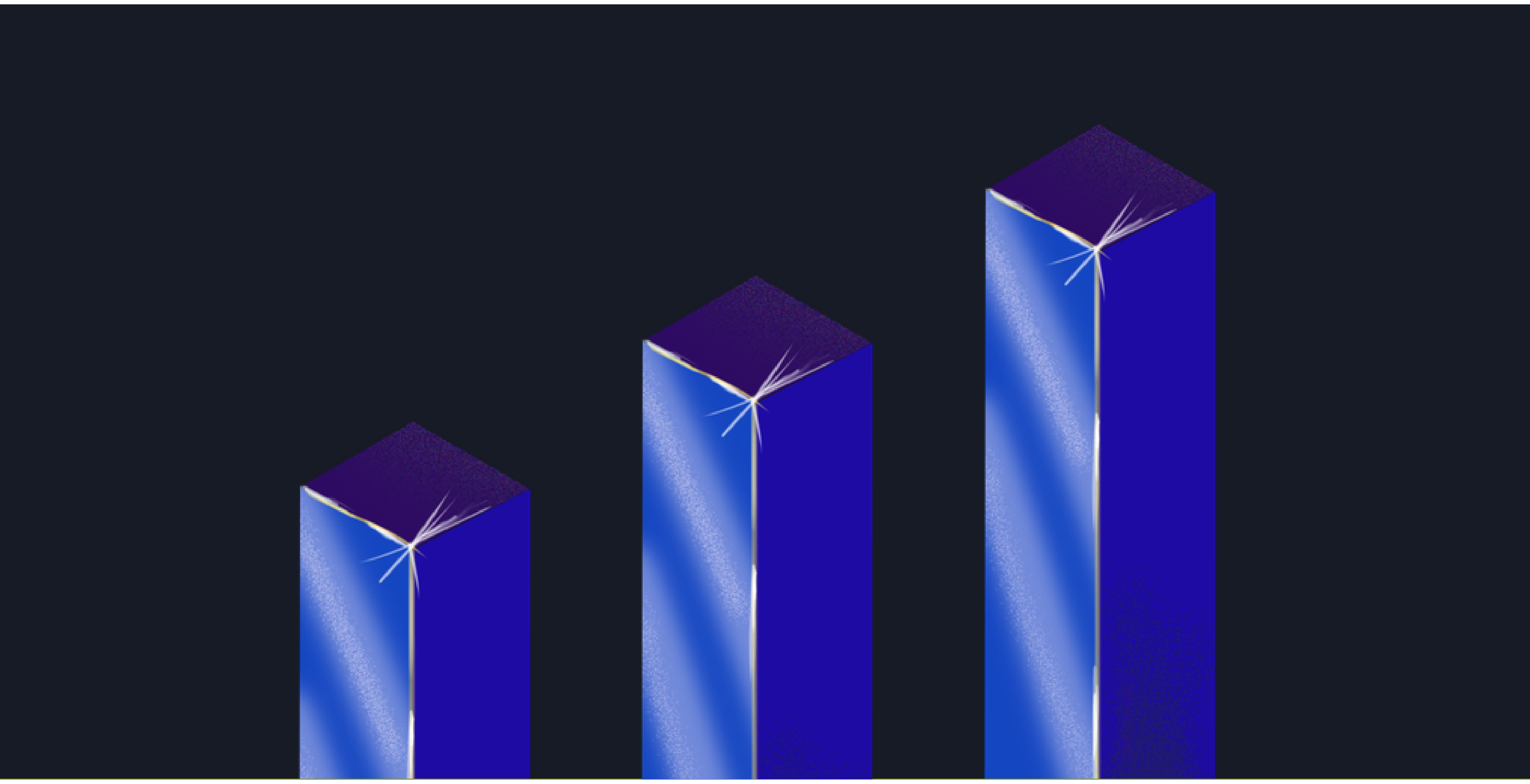
QUANSCIENT



Quanscient Webinar 30th April 2026 | PDF summary

The 3 foundational pillars of optimal MEMS design

See how cloud compute, agentic AI, and physics-aware models enable moving from specifications to the absolute best design

Contents

Overview

This webinar explored a fundamental change in engineering workflows: the shift from using simulation as a validation tool to using it as a mechanism for generating optimal designs.

Traditionally, engineers design a system and then use simulation to answer the question, “How will this design perform?” However, this approach inherently limits exploration because it evaluates only a small subset of the possible design space. The more relevant question, “What is the optimal design for my specifications?” has historically been difficult to answer because of computational and workflow limitations.

The session introduced a new paradigm enabled by Quanscient. This paradigm is built on three foundational pillars:

- Massive computational scale
- Programmatic control
- AI-driven optimization using physics-aware models

Together, these capabilities allow engineers to move beyond incremental iteration and instead generate optimal solutions directly from requirements.

WATCH THE WEBINAR RECORDING →

About the speakers

Juha Riippi

CEO & Co-founder, Quanscient

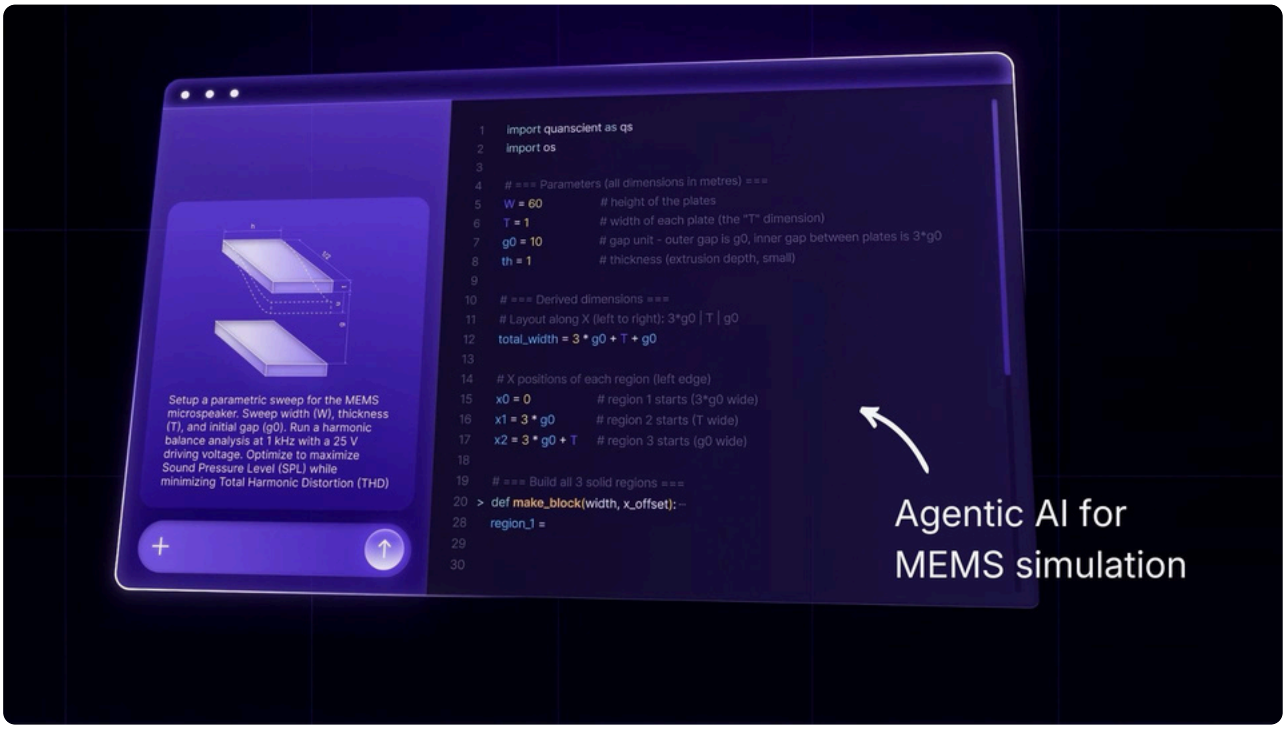
Juha has a long experience in software development and digitalization businesses. During that time he has been working in international sales, programming and leadership roles.

Dr. Andrew Tweedie

Co-CTO & Co-founder, Quanscient

With more than 20 years of experience, Dr. Tweedie has a passion for developing and leading cutting-edge technology in the engineering simulation space. He was the founder and CEO of Kogsys and co-founded OnScale, the first cloud engineering simulation platform.

What it takes to bring AI to hardware engineering?



While AI has rapidly transformed software development, dramatically increasing productivity and enabling even non-experts to build applications, hardware engineering has not experienced the same level of disruption.

Several structural challenges explain this gap. Many engineering tools still rely heavily on graphical user interfaces, which are inherently difficult for AI systems to interact with efficiently. In addition, workflows are often fragmented across multiple tools and teams, creating silos that limit scalability and automation.

As a result, simulation workflows remain largely unchanged. In many organizations, engineers still perform a limited number of simulations to validate a design. This approach leaves the vast majority of the design space unexplored, meaning that potentially superior or unconventional solutions may never be discovered.

To fully leverage AI, engineering workflows must evolve to support large-scale exploration, automation, and data-driven optimization.

The three foundational pillars

1. Massive computational scale

The first requirement for this change is access to large-scale computational resources. Cloud-based infrastructure enables engineers to run thousands, or even tens of thousands, of strongly coupled multiphysics simulations in parallel.

Unlike on-premises clusters, which are often constrained and require careful capacity planning, cloud computing provides elastic scalability. This is particularly important because simulation workloads are typically bursty. Engineers may need large amounts of computing power for short periods rather than constant usage.

With this level of scale, it becomes feasible to explore large portions of the design space rather than relying on a small number of design iterations.

2. Programmatic control

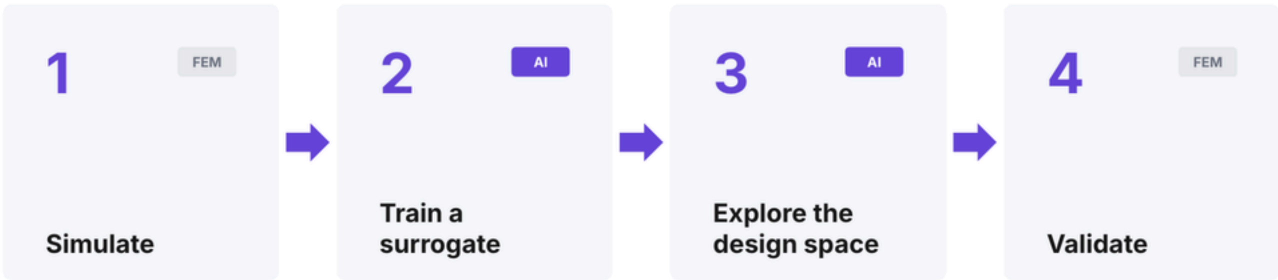
While computational scale is essential, it is not sufficient on its own. Without automation, running simulations at scale quickly becomes impractical. Programmatic control is therefore a foundational capability for modern simulation workflows, enabling simulations to be defined, executed, managed, and analyzed entirely through code rather than through manual interaction with graphical interfaces.

Through Python-based SDKs and APIs, engineers can automate the complete simulation lifecycle, from geometry generation and meshing to solver execution, optimization, and post-processing.

This allows organizations to move beyond isolated, manually executed simulations toward scalable and repeatable engineering workflows. Instead of running only a handful of design variations, teams can parameterize geometries and systematically explore large design spaces by executing tens of thousands or even hundreds of thousands of simulations in parallel using scalable computing infrastructure.

This programmatic approach enables a four-step workflow.

The three foundational pillars



Large-scale parametric simulation: Engineers first define parameterized models and run simulations across a broad design space.

Rather than evaluating a few manually selected configurations, entire ranges of geometric and physical parameters can be explored automatically.

This creates a high-quality dataset that captures the behavior of the design space in a structured and comprehensive manner.

Training surrogate models: The simulation data generated from these large-scale studies can then be used to train surrogate models.

These models approximate the behavior of high-fidelity numerical simulations while evaluating results in milliseconds instead of hours.

Because they are trained on simulation data that already spans the relevant design space, they provide a fast and reliable representation of system behavior.

Rapid design space exploration and inverse design: Once trained, surrogate models enable rapid exploration of the design space. Engineers can evaluate large numbers of candidate designs almost instantly and apply optimization algorithms to identify promising configurations.

More advanced workflows also become possible, including solving inverse problems in which the objective is not simply to evaluate a design, but to determine the optimal design that satisfies a specific set of performance requirements.

High-fidelity revalidation: The most promising candidates identified through surrogate modeling and optimization are then revalidated using deterministic, high-fidelity numerical simulations. This ensures that the final designs maintain the required level of physical accuracy before deployment or manufacturing.

The three foundational pillars

Programmatic control is also the key enabler for AI-driven engineering workflows. AI agents cannot efficiently operate through graphical user interfaces at an industrial scale. Instead, they require direct access to simulations through APIs and SDKs, allowing them to orchestrate workflows autonomously while keeping engineers in the loop for oversight and decision-making.

In this model, AI becomes an accelerator for engineering productivity, enabling experienced engineers to define, evaluate, and iterate on complex workflows significantly faster than traditional approaches.

Beyond automation and scalability, programmatic workflows also transform simulation models and scripts into long-term organizational assets. Simulation pipelines become reproducible, version-controlled, shareable, and centrally managed rather than fragmented across individual laptops, spreadsheets, or disconnected scripts. As a result, simulation workflows evolve into part of an organization's intellectual property and engineering intelligence.

Over time, the benefits compound. As more simulations are executed, organizations accumulate structured datasets describing their products and design spaces. These centralized datasets can then be leveraged for advanced analytics, optimization, and AI-driven insights, including identifying correlations between simulation outcomes and design decisions.

Whether deployed on premises or in the cloud, organizations maintain ownership of their data and models while gaining a significantly more scalable, maintainable, and intelligent engineering workflow.

3. MultiphysicsAI

The third pillar builds on the first two. With access to large datasets generated from simulations, engineers can train surrogate models that approximate system behavior. These models can make predictions in milliseconds, enabling rapid exploration of design alternatives.

This capability allows engineers to solve inverse problems. Instead of testing predefined designs, they can define performance targets and identify the designs that best meet those requirements.

In addition, these models can incorporate manufacturing variability, enabling predictions of real-world performance and yield. This represents a significant shift from traditional workflows, in which such analyses are often limited or performed late in the design process.

Streamlining hardware design with agentic simulation workflows

A central theme of the webinar was the potential of agentic AI to reshape engineering simulation workflows. Rather than requiring engineers to manually execute every step of the process, agentic workflows enable AI systems to automate many of the repetitive and time-intensive tasks involved in simulation and hardware development while engineers retain oversight and control.

These AI agents can support a broad range of activities, including:

- Model construction
- Analysis of existing models, including:
  - Mesh convergence
  - Design studies
  - Monte Carlo analysis
- Capturing results and preparing reports

A key emphasis throughout the discussion was that these workflows must remain transparent and inspectable. Engineers need to understand what the agent has done, verify that the outputs are valid, and confirm that the process aligns with robust engineering practices. This focus on inspectability and repeatability ensures that automation enhances confidence and reliability rather than obscuring critical decision-making processes.

The webinar presented a practical toolchain used internally:

- An AI-enabled IDE (Cursor) for interacting with agents
- A cloud-based multiphysics solver (Quanscient Allsolve) for running simulations
- A large language model (Claude Opus 4.6) acting as the agent
- A task management system (Linear) for tracking progress

This setup allows engineers to define tasks, monitor execution, and maintain a clear record of decisions and results. It also supports collaboration and reproducibility by centralizing information. Providing context is critical in this workflow. Agents rely on structured input, such as task descriptions, reusable “skills,” and predefined environments, to perform effectively.

PMUT simulation example

Overview

To illustrate these concepts, **Dr. Andrew Tweedie** presented an agentic workflow built around a piezoelectric micromachined ultrasonic transducer (PMUT), a device previously explored in several Allsolve webinars and widely used to demonstrate the platform's capabilities for PMUT array design.

The workflow began by creating a task in Linear directing an agent to:

- Create a simple PMUT model
- Extract and plot the relevant outputs
- Perform a mesh convergence study
- Track all progress in Linear
- Create a PowerPoint report

Acting much like a highly capable engineering assistant, the agent analyzed the requirements, developed an execution plan, and generated a complete simulation model using the SDK. Throughout the process, progress and task updates were tracked in Linear, creating a collaborative workflow similar to working with a colleague or intern on an engineering project.

Once the initial model was created, the engineer reviewed the generated setup before execution, reinforcing the importance of transparency and inspectability in agentic engineering workflows.

The agent was then instructed to perform a mesh convergence study automatically, iterating on the simulation while continuing to log progress and results.

Finally, the workflow concluded with the automatic generation of a PowerPoint presentation summarizing the model, results, and findings. This demonstrated how agentic systems can help automate not only simulation setup and analysis but also the often time-consuming task of preparing engineering deliverables for internal discussion and collaboration.

Setup

Providing the LLM with the right context is essential for achieving effective results. In Cursor, the "Skills" feature provides a practical way to supply that context.

An Allsolve SDK skill can be created to capture the knowledge and best practices required to work effectively with the SDK. Whenever an agent is asked to build or modify an Allsolve model, it can reference this skill automatically. As new workflows, edge cases, or issues arise, these skills can be updated and refined over time.

Another important step is setting up a local Python environment and installing the Allsolve SDK along with any supporting libraries that may be needed. Packages such as NumPy and Matplotlib are especially useful for post-processing simulation results and generating plots or visualizations.

Finally, Allsolve SDK credentials are typically stored in a local .env file to simplify authentication and keep credentials separate from source code, although alternative approaches may also be used depending on the workflow and environment.

PMUT simulation example

Problem description

The workflow began by creating a detailed Linear issue describing the PMUT cell simulation task. The issue captured key simulation parameters such as the target center frequency, stack-up description, lateral dimensions, boundary conditions, and drive specifications.

For relatively simple geometries, a text-based description was shown to be sufficient, although diagrams or CAD files could also be attached when available.

The issue further defined a set of concrete subtasks, including creating a unit cell model using the Allsolve SDK, running the simulation, generating a local Python notebook with output plots, performing a mesh convergence study, and preparing a final report for export to PowerPoint.

Once the task was documented in Linear, the agent was instructed to retrieve the issue, review the requirements carefully, generate a detailed execution plan, and execute the workflow.

As shown in the webinar, within a few minutes the agent was able to digest the information, organize the required steps, and prepare the implementation plan.

After review and approval in Cursor, the build process could then be triggered to automatically begin constructing the simulation model.

Resulting project

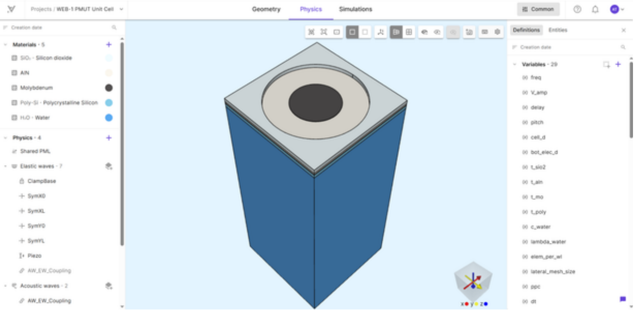
The webinar demonstrated how an AI agent can autonomously build a PMUT simulation model while still keeping engineers in control of the workflow.

The agent reviewed the project requirements, generated the geometry, physics setup, material definitions, and boundary conditions, and then paused for human inspection before continuing.

The resulting project was created in Allsolve's UI-driven environment, where all parameters such as excitation voltage, frequency, and dimensions were fully configurable.

Because the model was created in a transparent and inspectable way, engineers could review, modify, or share the project with colleagues before proceeding.

This approach combines automation with engineering oversight, ensuring that simulation models remain understandable and collaborative.



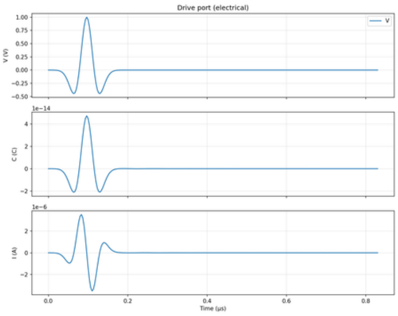
PMUT simulation example

Simulation results: time domain

After approval, the agent continued execution and generated the requested simulation reports. The time-domain outputs included both electrical and mechanical results.

Electrical outputs from the excitation port included:

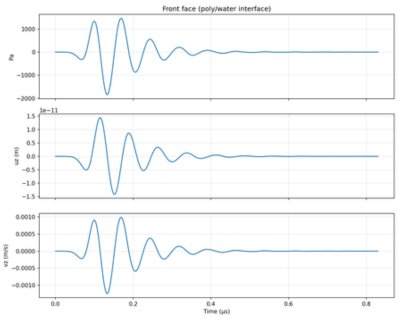
- Voltage
- Current
- Charge



Mechanical outputs focused on the PMUT front face and included:

- Pressure
- Velocity

This demonstrated how AI-driven workflows can automate not only simulation setup but also the extraction and reporting of key engineering metrics.



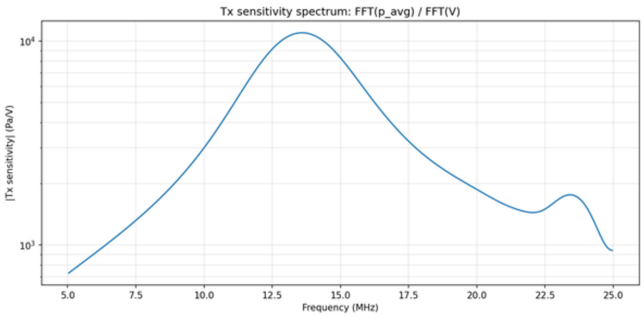
Simulation results: frequency domain

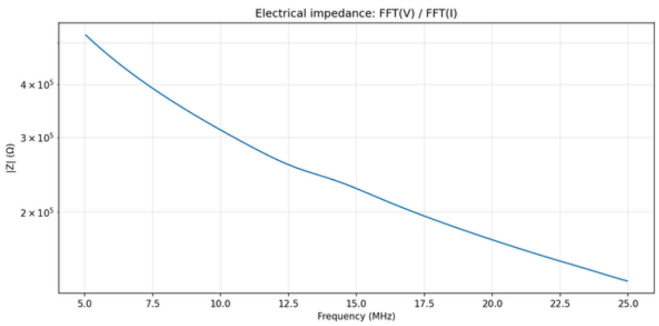
The workflow also included automated post-processing in the frequency domain. The agent calculated:

- Transmit sensitivity spectrum
- Complex electrical impedance spectrum

Results were automatically distributed through integrated tools such as Linear and local development environments.

The webinar highlighted how notifications from the Linear app provided immediate updates on simulation progress, demonstrating how engineering workflows could be integrated into broader collaborative and automated systems.





PMUT simulation example

SDK code

```python
def main():
    client = allsolve.Client(dotenv_file=".env")
    project = client.create_project(
        name="WEB-1 PMUT Unit Cell",
        description="PMUT cell transient piezo-acoustic simulation (60 um pitch, 12 MHz)",
    )
    project.pml_num_layers = 6
    project.save()

    create_variables(project)
    build_geometry(project)
    regions = create_regions(project)
    assign_materials(project, regions)
    configure_physics(project, regions)
    mesh, status = run_mesh(project, regions)
    save_state(project, mesh, status, client.host)


if __name__ == "__main__":
    main()
```

Allsolve SDK project creation syntax from pmut_model.py

Although the model was built through the graphical interface, every project also generates underlying Python SDK code. This code provides a structured, high-level representation of the simulation workflow, including:

- Variable creation
- Geometry generation
- Physics setup

Having simulations represented in code offers several advantages:

- Programmatic execution of simulations
- Integration into design studies and optimization workflows
- Version control through Git repositories
- Easier debugging and testing using AI agents

Dr. Andrew Tweedie explained that agents could later be used to improve or validate simulations by automatically generating additional test cases, helping engineers increase confidence in simulation robustness and accuracy.

Mesh convergence study

The webinar showcased an automated mesh convergence study performed by the agent. The simulation was executed in parallel using four different mesh sizes while tracking the PMUT center frequency.

As the mesh was refined, the center frequency converged toward a stable value. A fitted convergence curve was used to estimate the final converged solution without requiring extremely fine meshes.

This approach allows engineers to estimate the expected simulation accuracy while balancing runtime and precision requirements. It also enables the use of coarse meshes during the early stages of design exploration, with the flexibility to increase accuracy later in the design process as needed.

The agent was instructed to perform the following tasks:

- Sweep lateral_mesh_size over [10, 7.5, 5.0, 2.5] um
- Monitor peak Tx sensitivity frequency (fc).
- Plot fc and % error vs mesh size.

Results

- Fully converged fc predicted to be 12.91 MHz
- The 1% error threshold was exceeded at 4 µm

PMUT simulation example

Exploring the problem space

Dr. Andrew Tweedie discussed how AI-driven simulation workflows support broader design exploration tasks. Instead of evaluating only a single design, engineers can use large simulation datasets to explore many design alternatives.

Allsolve-generated datasets were used to train AI surrogate models capable of rapidly predicting simulation outcomes. These surrogate models significantly accelerate exploration of engineering design spaces and help solve inverse design problems, such as:

- Finding the optimal PMUT design
- Identifying trade-offs between performance objectives
- Generating multiple viable design options

With combination of simulation, AI, and automation, Quanscient MultiphysicsAI enables much faster iteration than traditional engineering workflows.

The pareto front

The webinar introduced the concept of Pareto front in PMUT design optimization. Two competing objectives were considered:

- Bandwidth
- Device sensitivity

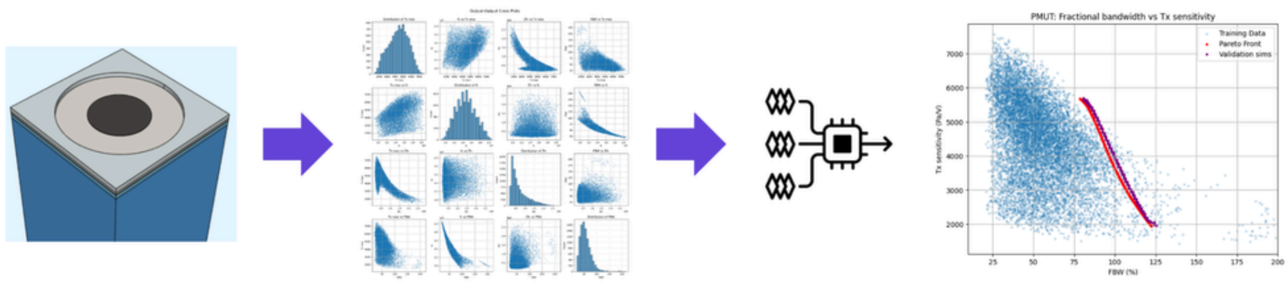
The AI workflow generated a large simulation dataset, trained a surrogate model, and identified Pareto-optimal designs in which improving one objective would reduce the other.
The resulting Pareto front allowed engineers to choose between:

- Wide-bandwidth designs
- High-sensitivity designs
- Balanced intermediate solutions

Compared to the original PMUT design, the optimized solutions achieved:

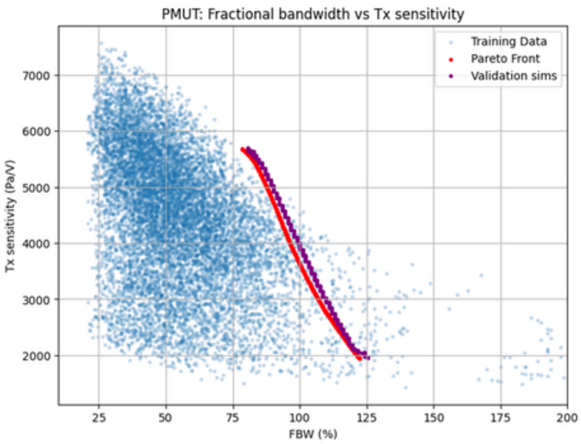
- A 35% improvement in bandwidth
- Increased sensitivity
- Accurate matching of the target center frequency

This example highlighted the practical value of agentic AI workflows for automated engineering optimization.

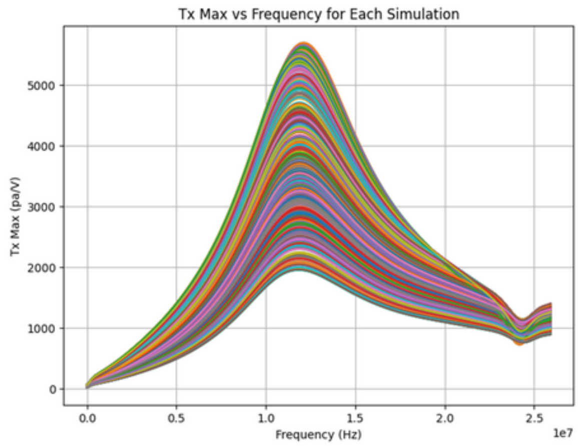


Fully parameterised model   10k simulation dataset   AI surrogate training   Problem space exploration

PMUT simulation example



PMUT unit cell Pareto front for fc=12MHz



Transmit sensitivity of Pareto front designs

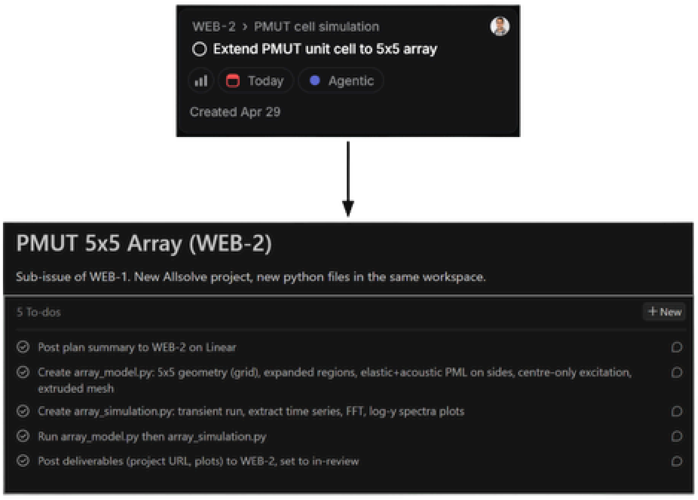
Building on existing projects

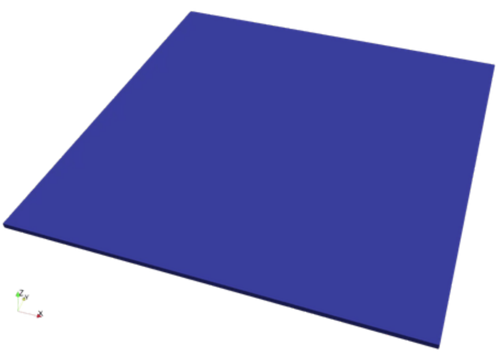
How existing simulation projects can serve as context for future AI tasks is demonstrated. In the example, the original PMUT unit cell model was extended into a 5×5 array model through a new agent task.

The engineer provided:

- Updated boundary condition requirements
- Voltage excitation placement instructions

The agent generated a plan and quickly produced the expanded PMUT array model. This showcased how reusable project context can significantly accelerate the development of more complex simulations.

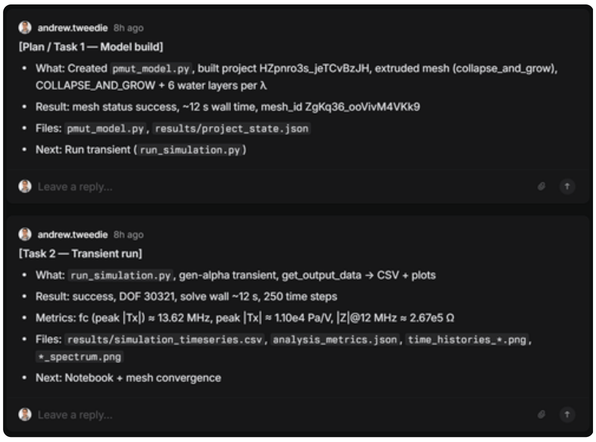




PMUT simulation example

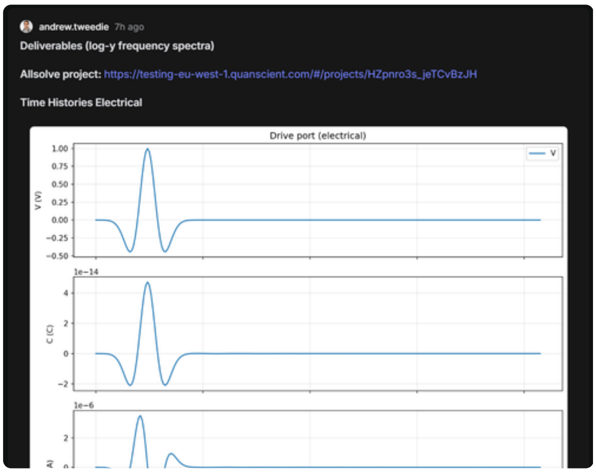
Reporting

Dr. Tweedie emphasized the importance of transparency and traceability in AI-assisted engineering. The agent automatically posted concise progress updates and results summaries to Linear issues throughout the workflow.



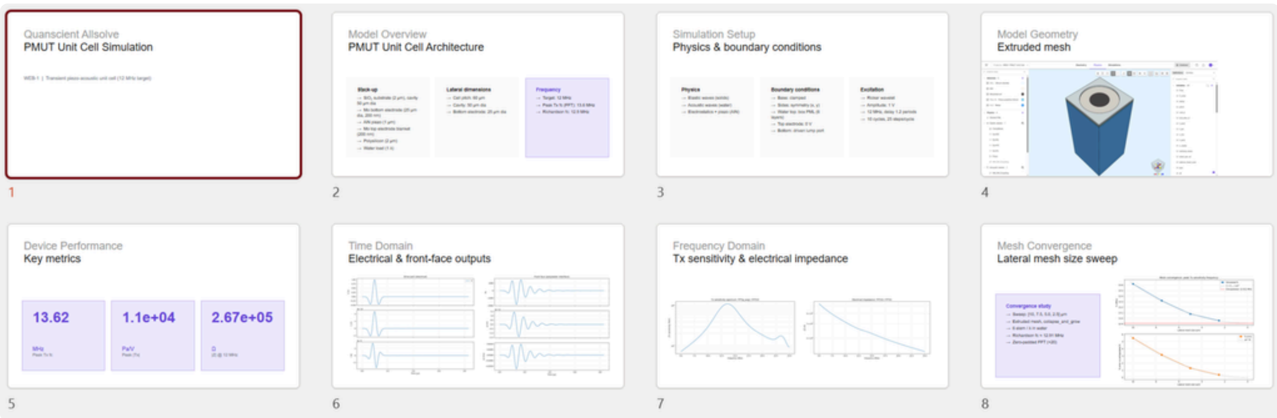
These updates were intentionally designed to remain short and focused so engineers could easily monitor progress without being overwhelmed by excessive information.

This reporting capability enables engineers to:
- Review intermediate results
- Detect potential issues early
- Request additional tests or validation
- Maintain confidence in simulation quality



Slides

The workflow also included automated generation of PowerPoint slides summarizing simulation activities and results. Although the presenters acknowledged that the slides were basic in appearance, they highlighted the practical value of automatically producing presentation materials for internal communication and reporting purposes.



Agentic simulation

Missing information

A common concern regarding AI agents is addressed: handling incomplete information.

When insufficient details are provided, the agent identifies missing requirements and asks clarifying questions.

For example, when the user omitted details about the top ground electrode, the agent automatically requested clarification in a structured multiple-choice format.

This interaction was compared to collaborating with a colleague through email or messaging platforms.

Tips and tricks

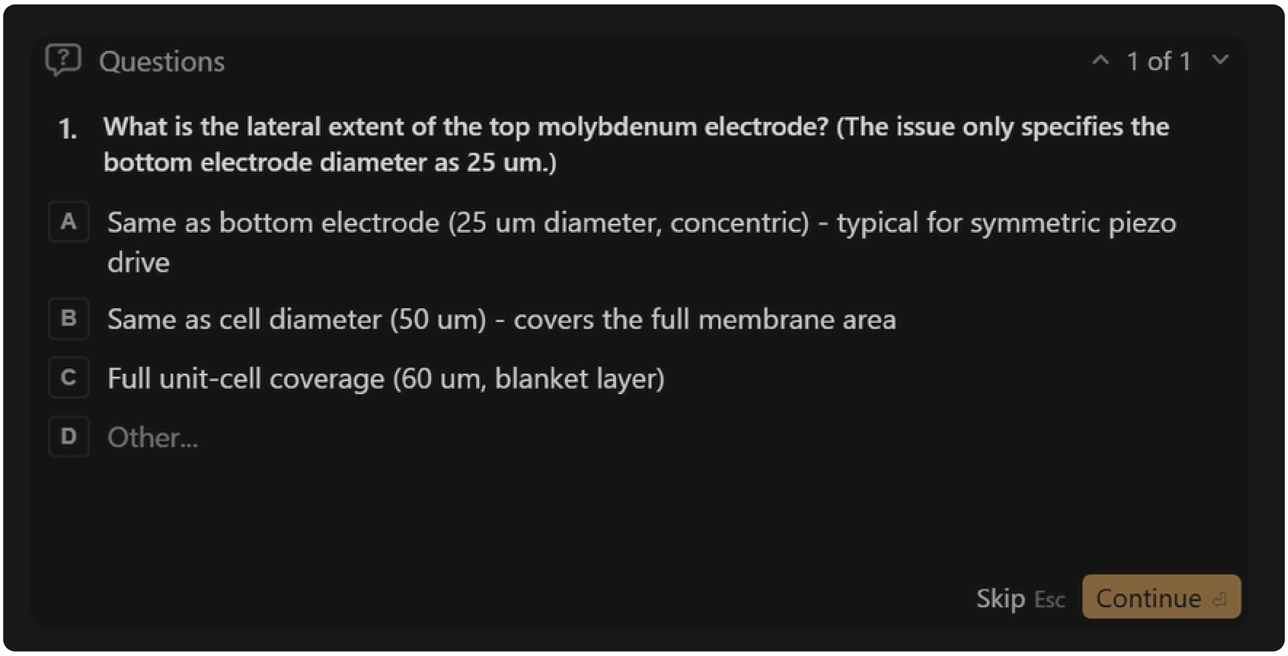
Dr. Andrew Tweedie recommended treating AI agents like highly capable engineering interns. While agents can solve complex technical problems, they still require:

- Clear instructions
- Explicit engineering requirements
- Context about organizational standards and workflows

Two forms of context were emphasized:

1. Task-specific context provided directly in prompts
2. General organizational context stored in reusable instruction files or "skills"

This structured approach helps ensure consistent and reliable outputs across projects.



Insights from the Q&A

The discussion also addressed practical questions about adoption and scalability:

Cost considerations

Running large numbers of simulations can be surprisingly affordable for smaller models, sometimes comparable to everyday expenses, while larger models scale with complexity.

Flexibility

The workflow is not tied to a specific software stack and can be integrated into different environments.

Accessibility

Engineers do not need advanced programming skills to get started. GUI-generated scripts and AI assistance make the transition to programmatic workflows easier.

Future workflows

Engineering work is already evolving, with AI significantly increasing productivity. The expectation is that engineers will be able to operate at much higher levels of efficiency, focusing on problem-solving rather than manual tasks.

Conclusion

The webinar highlighted how the convergence of cloud computing, programmatic control, multiphysics simulation, and agentic AI is driving a fundamental transformation in hardware engineering.

By shifting from traditional validation-focused workflows to AI-assisted generative exploration, engineers can evaluate vast design spaces, identify optimal solutions, and incorporate real-world variability much earlier in the development process.

The presenters emphasized that SDK-driven, agentic workflows automate repetitive low-level tasks while improving reproducibility, documentation, transparency, and inspectability.

This allows engineers to spend less time on manual setup and iteration, and more time making higher-level engineering decisions and extracting meaningful insights.

Rather than viewing AI-driven automation as an experimental add-on, the webinar positioned these workflows as an emerging standard for modern engineering practice at Quanscient.

As these technologies continue to evolve, they are expected to accelerate innovation, enhance simulation-driven decision-making, and enable more efficient and scalable product development across the hardware engineering industry.

Key takeaways

- Engineering workflows are shifting from using simulation mainly for validation toward using AI-assisted simulation to directly identify optimal designs from performance requirements.
- Agentic AI workflows can automate repetitive engineering tasks such as model building, mesh convergence studies, reporting, and optimization, while engineers retain oversight and control.
- Three foundational pillars enable this shift: large-scale cloud computing, programmatic simulation workflows through SDKs and APIs, and AI-driven optimization using physics-aware surrogate models.
- The PMUT example demonstrated how combining simulation, cloud compute, and AI-driven optimization can significantly accelerate design exploration and improve device performance, including a 35% bandwidth improvement.
- Programmatic workflows allow engineers to automate thousands of multiphysics simulations, generate large structured datasets, and train AI models that can explore design spaces in milliseconds instead of hours.

Get in touch

Learn more and request a demo at
quanscient.com →

Juha Riippi

CEO & Co-founder
juha.riippi@quanscient.com

Dr. Andrew Tweedie

Co-CTO & Co-founder
andrew.tweedie@quanscient.com

QUANSCIENT

quanscient.com

info@quanscient.com

linkedin.com/company/quanscient