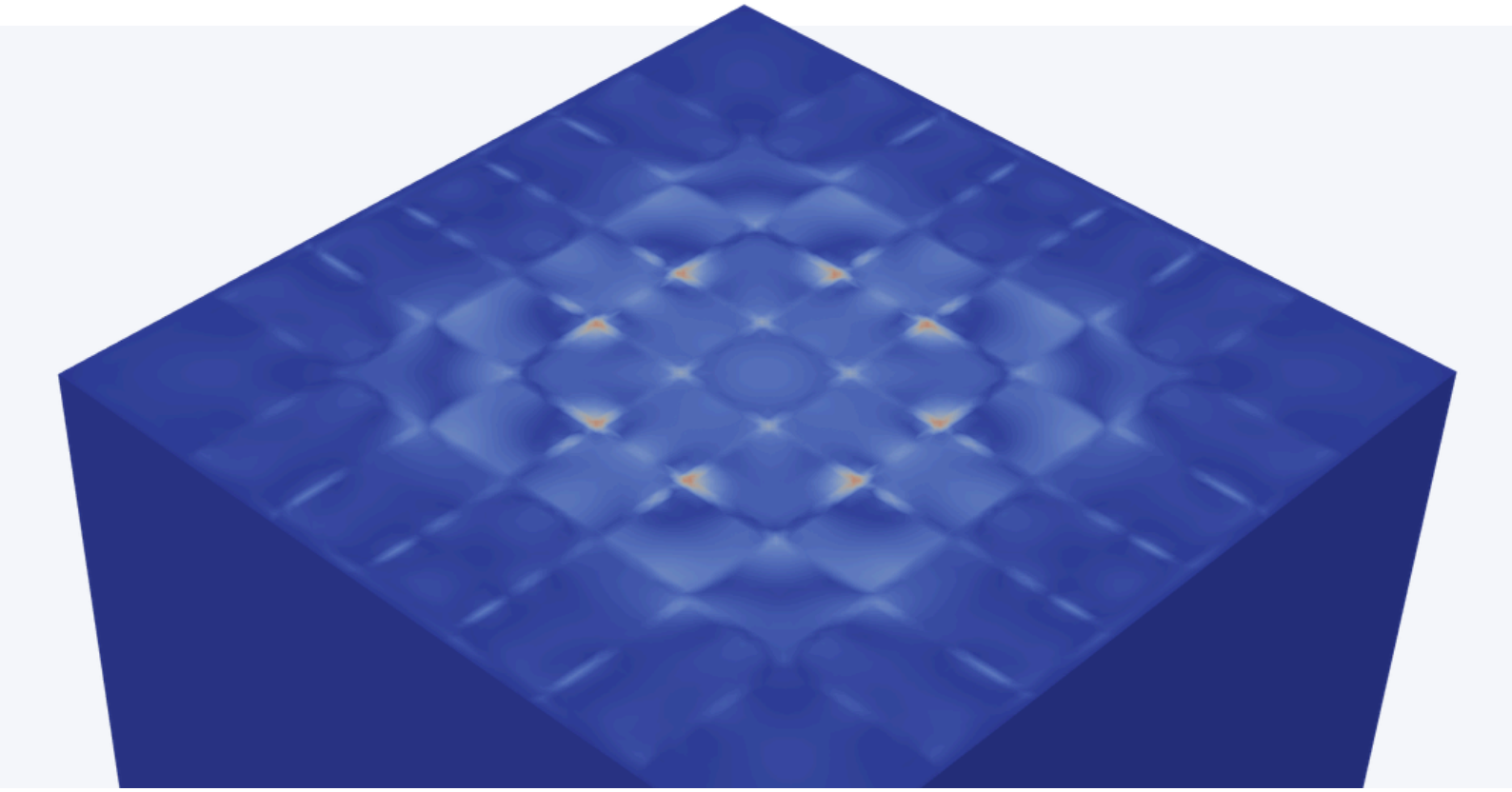
QUANSCIENT



Quanscient Webinar Oct 29th | PDF summary

Simulating ultrasonic piezocomposite arrays

Large arrays and AI-driven optimization — a live walkthrough in Quanscient Allsolve.

Contents

Overview 3

About the speakers 3

Introduction to Quanscient Allsolve 4

Why piezocomposite arrays are hard to simulate 4

The ladder of models 5

Step 1: The unit cell 6

Step 2: The large array 8

New feature: acoustic extrapolation 9

Design example: sub-dicing the matching layer 9

Live demo: the array model in Quanscient Allsolve 10

Step 3: AI-driven optimization 11

Questions and answers 13

Key takeaways 14

Get in touch 14

Overview

This webinar showed how Quanscient Allsolve takes the simulation of ultrasonic piezocomposite arrays all the way from a fast unit-cell model to a full, finite array model and on to AI-driven design optimization, using one platform, one set of physics and the same model carried through each step.

We began with why piezocomposite arrays are hard to simulate: their size (hundreds or thousands of pillars with fine features), fully coupled physics (elastic waves, acoustic waves and piezoelectric coupling) and broadband, transient behavior. We then walked up the modeling "ladder": a quarter-symmetry 1-3 piezocomposite unit cell (about 19,000 degrees of freedom, solved in 39 seconds on 3 CPUs) that is the right tool for stack design but cannot capture crosstalk, lateral modes, directivity or beam steering; and a finite 38×38-pillar, 19×19-element array with the identical stack (about 9.1 million degrees of freedom, solved in under an hour on 112 cores) that reveals exactly those effects.

Allsolve's new acoustic extrapolation feature, based on the Kirchhoff integral, was used to obtain on-axis pressure and far-field beam patterns without modeling a large water volume. A design comparison showed that sub-dicing the matching layer and filling the cuts with silicone rubber widens the −6 dB beam width from 53.8° to 65.7° at the cost of peak sensitivity (57.5 to 48.3), a trade-off that can now be evaluated before anything is manufactured. A live demo walked through the model setup, physics, simulation, extrapolation outputs and results in the Allsolve web UI.

Finally, an AI-driven optimization built on the Allsolve SDK explored six stack parameters of a two-matching-layer unit cell against two competing objectives (peak transmit sensitivity and −6 dB fractional bandwidth) with a band-center constraint, producing a Pareto front from which designers can choose; the design nearest the baseline raised fractional bandwidth from 69% to 84% while also increasing peak transmit sensitivity. The session closed with audience Q&A.

About the speakers



Jeff Dobson

Senior Application Engineer, Quanscient

Jeff Dobson is a Senior Application Engineer at Quanscient with a background in ultrasound simulation. He completed an engineering doctorate focused on simulating ultrasonic inspections of composite materials and has worked with many large companies in medical and NDT applications, much of it centered on simulating ultrasonic devices. At Quanscient he leads the work with ultrasound customers across diagnostic, ultrasonic flow and other ultrasound applications.



Jukka Knuutinen

Head of Marketing, Quanscient

Jukka Knuutinen is Head of Marketing at Quanscient. He hosted the webinar and moderated the audience Q&A.

Introduction to Quanscient Allsolve

The cloud-based multiphysics simulation platform Quanscient Allsolve was used for all simulations featured in the webinar.

About Quanscient Allsolve

- A cloud-based multiphysics simulation platform and the flagship product of Quanscient, a company founded in Finland in 2021
- Natively coupled physics in the solver and cloud-scale compute: hundreds or thousands of cores with domain decomposition for very large models
- Runs in the web browser; everything built in the UI generates an underlying Python simulation script that can be customized, and a Python SDK allows models to be built and run through APIs
- Includes a built-in AI agent for asking questions about a model, getting feedback and adding model features (covered in separate webinars)

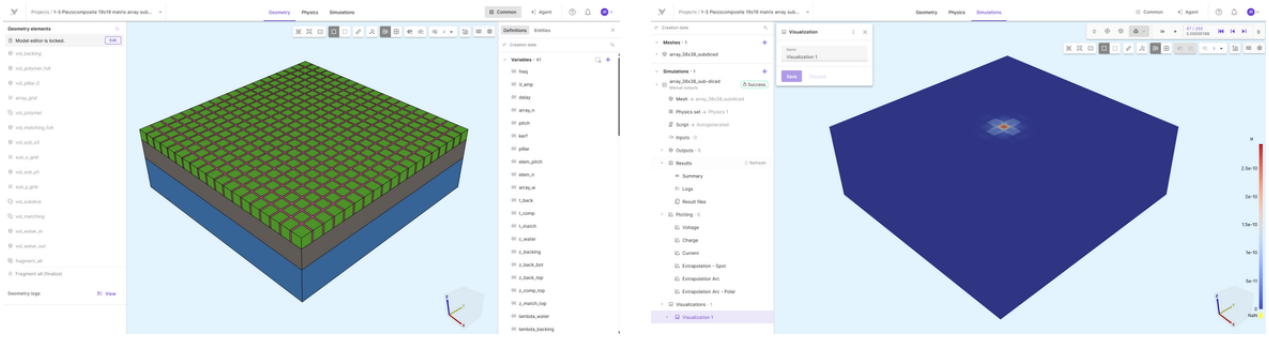


Figure 1 · Quanscient Allsolve runs in the web browser.

Why piezocomposite arrays are hard to simulate

Piezocomposite arrays still go into many of the imaging probes used across industries from medical to NDT. The physics of the stack is well understood; the difficulty has always been modeling the array, particularly large arrays, for three reasons:

- **Size:** realistic arrays have hundreds or even thousands of pillars, each with fine features, which quickly exceeds what a workstation can handle.
- **Coupled physics:** elastic waves in the solid, acoustic waves in the water and piezoelectric coupling between the electrical and mechanical sides, all on a complex geometry of pillars, kerfs, matching layers and backing.
- **Broadband, transient behavior:** the whole band matters, so transient simulation is typically needed, and beam patterns traditionally required modeling a large volume of water in front of the array.

As a result, the true behavior of an array has often only been discovered after it was built and measured on the bench, making corrections time-consuming and expensive.

The ladder of models

In practice, designers use a ladder of models, each rung answering a question the one below cannot:

01

Unit cell

A single pillar (typically a quarter, with symmetry on each side) representing an infinite array. The right tool for fundamental stack design: impedance, surface displacement and coupling.

02

Cross sections

Elevation and azimuth strips (quarter or half element, or 2D) that start to give beam patterns, array impedance and crosstalk between elements.

03

Full 3D section or full array

Gives the closest prediction of imaging performance before the device is built.

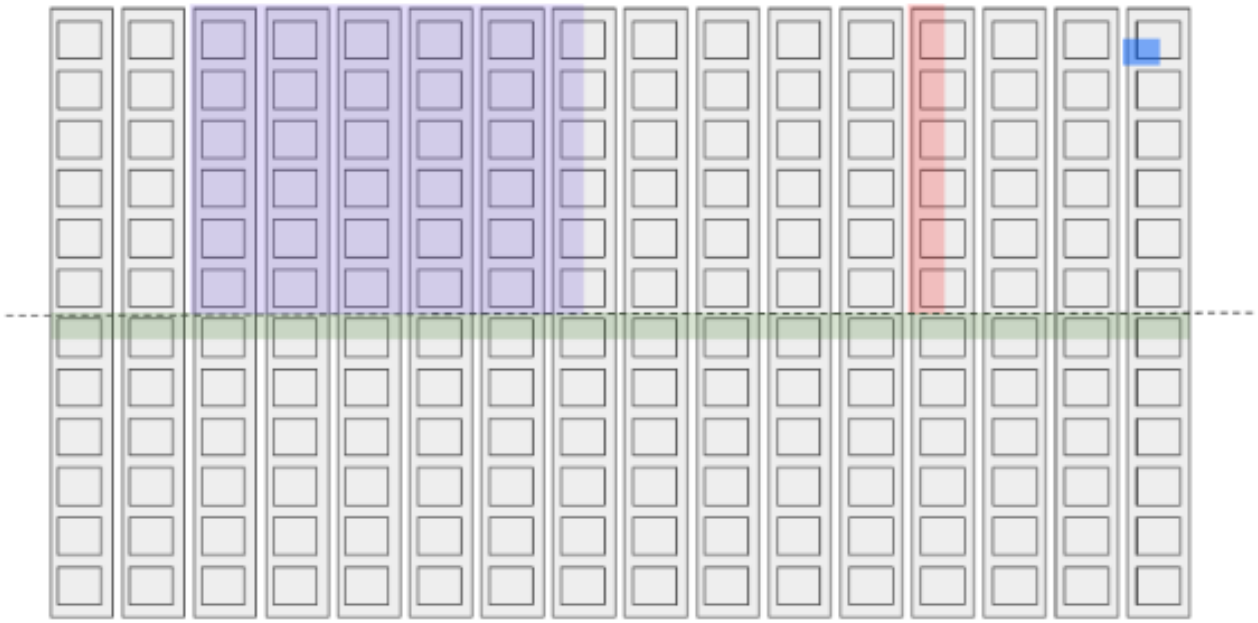


Figure 2 · The ladder of models, from unit cell to full array.

Each rung has traditionally cost more compute and setup effort. The key message of the webinar is that, with Allsolve, climbing the ladder no longer means changing tools, rebuilding the model or waiting days for results: the same model, with one set of physics, is carried from stack design (unit cell) to large arrays (crosstalk and lateral modes) to AI-guided design.

Unit cell

Stack design, impedance and transmit sensitivity.

Large arrays

Crosstalk, lateral modes and directivity.

AI-guided design

Explore the design space against your KPIs.

Step 1: The unit cell

Model setup

The unit cell is a quarter-symmetry 3D model of a 1-3 piezocomposite pillar with symmetry on all lateral faces, representing an infinite array of identical pillars. The stack, deliberately kept identical throughout the webinar so that any differences come from the array and not from the materials, consists of a backing layer, the composite with PZT-5H piezoelectric material, a hard-set matching layer and one wavelength of water load (1.48 mm at the 1 MHz operating frequency).

- **Physics:** elastic waves in the solid parts, acoustic waves in the water, electrostatics with piezoelectric coupling in the PZT, and the acoustic-structure interface where solid meets water.
- **Boundary conditions:** PMLs at the bottom of the backing and the top of the water to prevent reflections; symmetry on the lateral faces.
- **Electrical drive:** ground electrode and a broadband drive on the top and bottom of the pillar; transient analysis covering the whole band in a single run.

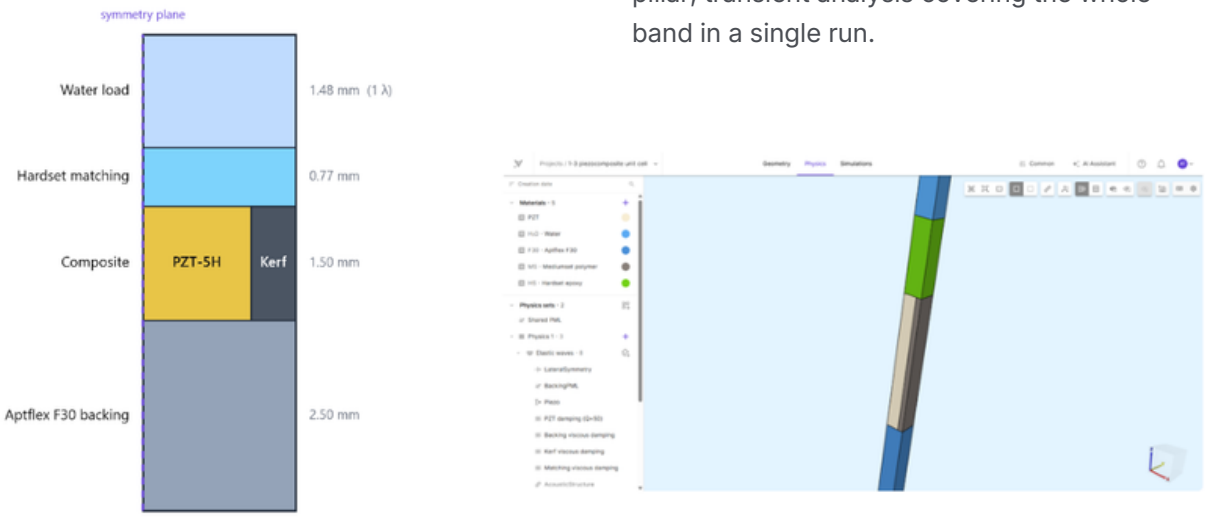


Figure 3 · Unit-cell geometry (left) and simulation setup (right).

Results

The field animation shows the pulse generated in the pillar, the displacement through the stack and the acoustic pressure in the water load: very clean thickness-mode behavior with no lateral complications. Raw time signals include voltage, charge and current from the drive port and the average pressure or velocity on the front face; the ring-down of the main pulse already reveals a lot about bandwidth and damping.

Taking the FFT of the time signals gives the transmit sensitivity (front-face average response per volt of drive across the band) and the electrical impedance, from which resonance, anti-resonance and coupling can be read. The same run also yields center frequency, −6 dB fractional bandwidth and transmit level, i.e. all the stack KPIs from a single run.

Step 1: The unit cell

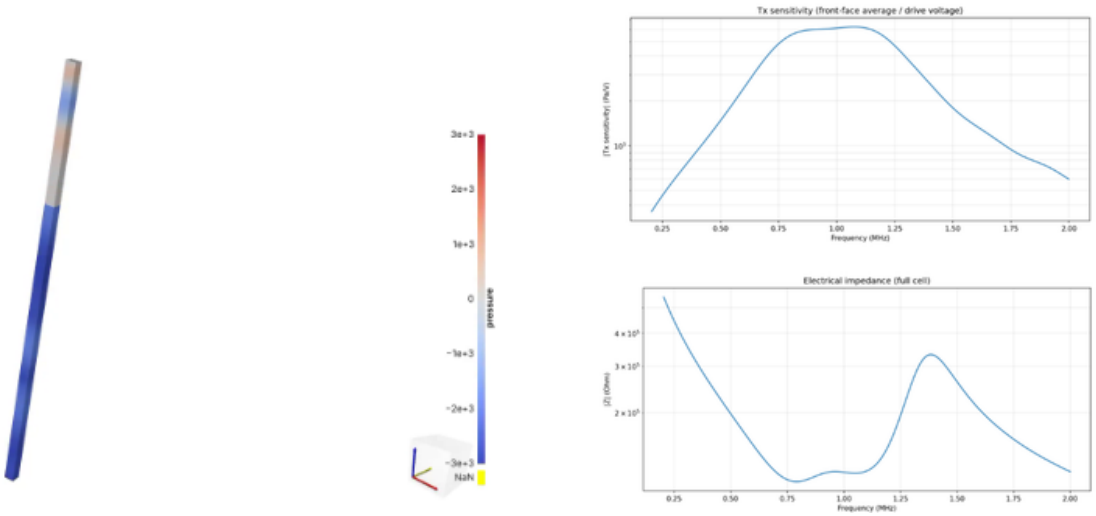


Figure 4 · Unit-cell field animation frame (left) and spectra (right).

The unit cell is fast: just under 19,000 degrees of freedom, solved in 39 seconds on 3 CPUs. That makes it the right tool for stack trade-offs such as backing material and thickness, ceramic volume fraction and matching-layer thickness, and an ideal model for AI optimization, where many simulations are needed.

18.8k
Degrees of freedom

39s
Run time on 3 CPUs

Where the unit cell stops being enough

The unit cell assumes every pillar has identical neighbors in every direction, forever. A real array is finite: it has edges, energy travels sideways between elements (crosstalk), the diced aperture has lateral modes, each element and the array have a directivity and beam pattern, and elements can be driven selectively with delays or apodization to steer or focus the beam.

None of this exists in an infinite periodic model, so seeing how an array actually behaves requires modeling the array itself.

No finite edges — Element-to-element crosstalk is invisible.

No real aperture — No lateral modes across a diced aperture, no element directivity or beam pattern.

No selective drive — A single driven element, delays and apodization cannot be represented.

Step 2: The large array

Model setup

The large array model is a realistic design-study section rather than an academic extreme: 38×38 pillars grouped into 19×19 elements, each element being 2×2 pillars. It uses exactly the same stack as the unit cell. The only change that matters is that the lateral symmetry boundaries are replaced by absorbing PMLs at the array edges, making it a genuinely finite array.

The central element (the central 2×2 pillars) is driven on purpose to excite crosstalk and lateral modes; this is also the building block for later beam steering, where elements are controlled independently.

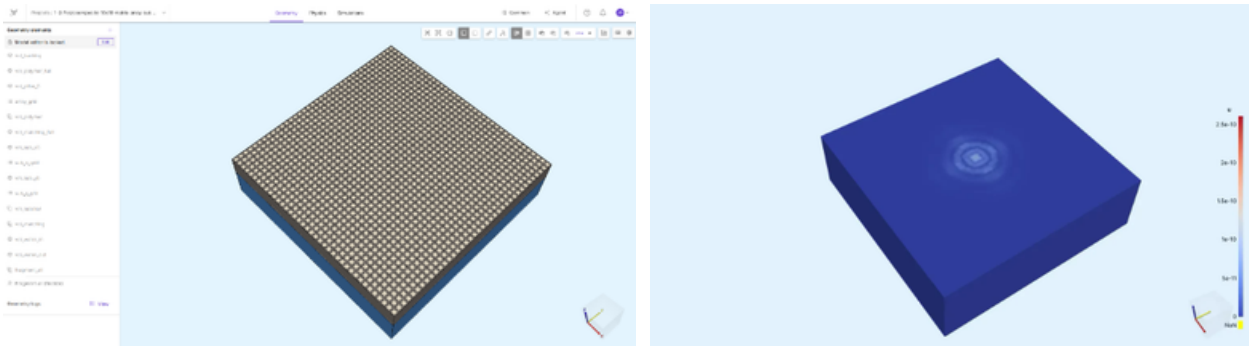


Figure 5 · Large array geometry (left) and surface displacement with the centre element driven (right).

Results

The surface displacement field shows energy spreading from the driven element into its neighbors as rings through the matching layer and composite: the crosstalk and lateral behavior that the unit cell cannot show. The spectra, transmit sensitivity (pressure in the water per volt of drive) and electrical impedance of the driven central element, now reflect an element sitting in a finite, coupled aperture, with the response shaped by its neighbors as well as by the stack.

The model has about 9.1 million degrees of freedom and solved in under an hour on 112 cores in Allsolve, with the same physics as the unit cell simply scaled up. At this run time, design studies on full array models are practical before anything is manufactured.

~9.1M	56 min
Degrees of freedom	Wall-clock time on 112 cores

New feature: acoustic extrapolation

Allsolve's UI now includes acoustic extrapolation, which uses the Kirchhoff integral to compute the pressure anywhere outside the modeled domain. This gives on-axis pressure and beam patterns at any distance without simulating wave propagation through a large volume of water. It drops directly into an existing model, and the far-field results come out alongside everything else.

For the large array, the extrapolated results included the on-axis pressure pulse and ring-down at a point in the water and the far-field beam pattern, exactly what is needed when thinking about array steering.

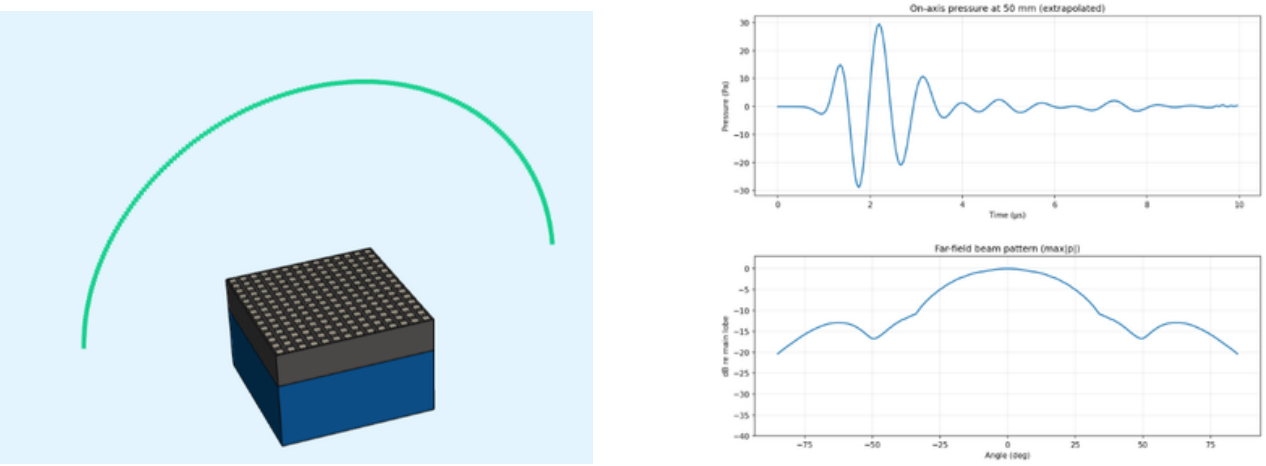


Figure 6 · Extrapolated on-axis pressure (left) and far-field beam pattern (right).

Design example: sub-dicing the matching layer

The baseline design used a continuous epoxy sheet as the matching layer across the whole array. It is easy to manufacture, but it acts as a stiff plate that mechanically couples each element to its neighbors, increasing crosstalk. The alternative design sub-dices the matching layer along every element boundary and fills the cuts with a soft material, silicone rubber; the composite, backing and center-element drive are unchanged.

The simulation shows that the change is not free:

- **Wider beam:** the −6 dB beam width increases from 53.8° to 65.7°, which matters for phased arrays because element directivity limits how far the beam can be steered before too much signal is lost.
- **Lower sensitivity:** peak sensitivity drops from 57.5 to 48.3.

Whether the trade is worth it depends entirely on the application. The point is that the designer can ask the question and see the whole trade-off before building anything, which is far quicker and cheaper than going through manufacturing.

Design example: sub-dicing the matching layer

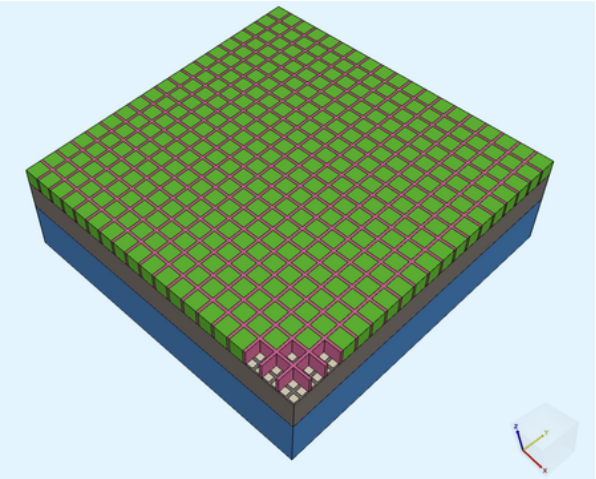


Figure 7 · Sub-diced matching layer geometry. Note, some matching layer regions have been hidden to show the PZT pillars below

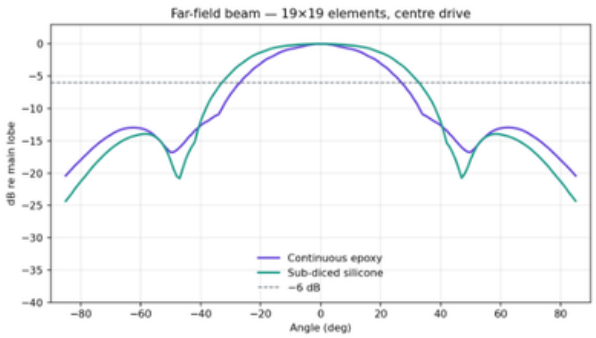


Figure 8 · Element beam pattern, baseline vs sub-diced.

Metric	Continuous epoxy	Sub-diced silicone
Center frequency	1.14 MHz	1.02 MHz
−6 dB beam width	53.8°	65.7°
Peak Tx at 50 mm	57.5 Pa/V	48.3 Pa/V
|Z| at f c	32.5 kΩ	28.6 kΩ
−6 dB fractional bandwidth	61%	50%

Same device and aperture. About 1.5 dB of transmit sensitivity buys a wider, cleaner element beam, but center frequency and bandwidth also drop.

Live demo: the array model in Quanscient Allsolve

Allsolve runs in the web browser. The demo walked through the array model tab by tab:

- **Geometry:** built from primitive shapes or imported CAD, with patterning to create the array without building each pillar individually. Variables (frequency, number of elements, all dimensions) make the model fully parameterized, so sweeps over dimensions or material properties can be set up quickly. Functions allow custom drive signals, overrides define sweeps, and regions specify volumes and surfaces to which materials, loads and boundary conditions are applied, so they follow geometry changes automatically.
- **Physics:** material definitions (PZT with elasticity matrix and piezoelectric properties, plus the other array materials), then elastic waves, acoustic waves and electrostatics with PMLs, damping, piezoelectric and acoustic-structure coupling, and electrodes (drive on the central four pillars, ground on top, passive electrodes on the other elements). Each element can be given individual electrodes to read every element's electrical signal or to drive elements individually for beam steering.

Live demo: the array model in Quanscient Allsolve

- **Simulations:** meshes and simulations are created by selecting a mesh and a physics set; several physics sets with different boundary conditions or loads can coexist in one model. Everything done in the UI generates a Python simulation script in which sections can be disabled and custom code inserted, with two-way coupling so UI changes do not overwrite the customizations.
- **Outputs:** voltage, charge and current from the drive electrode, the displacement field, and extrapolation outputs. Extrapolation points can be defined as an arc (taking the maximum pressure at each green point to build the beam profile), single spots for time traces, lines, or a 2D grid of points for a field profile.
- **Results:** the run used 112 cores. Voltage, charge, current, the extrapolated maximum pressure along the arc (also as a polar plot) and the displacement field can all be viewed in the UI, downloaded as CSV for post-processing elsewhere, and turned into animations. Plots can be watched while the simulation is still running on the cloud.

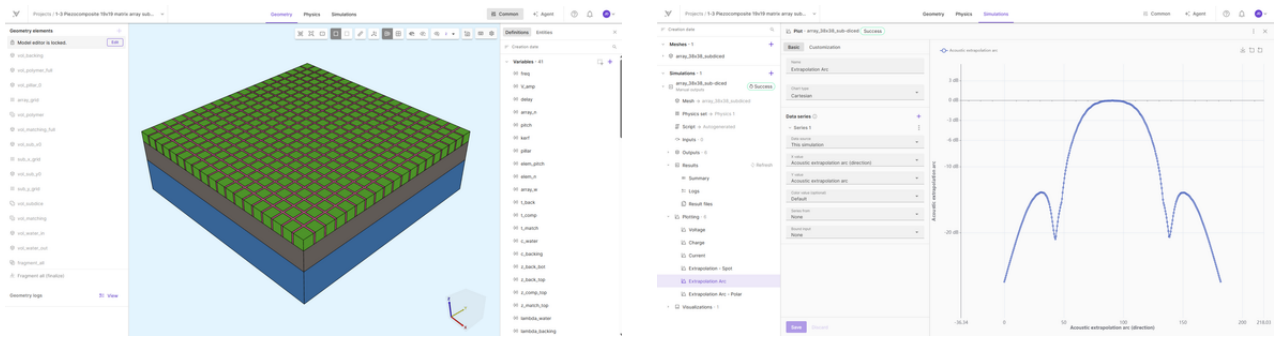


Figure 9 · The array model in the Allsolve web UI: geometry (left) and results (right).

Step 3: AI-driven optimization

Principles

Any optimizer is only as good as the numbers it is optimizing: if the KPIs coming out of the model are wrong, the AI will be very efficient at finding the wrong design. The approach therefore has two layers: an AI search layer that proposes designs, balances competing objectives and decides where to spend compute, and a physics layer in Allsolve that scores each candidate. The right physics model is chosen for the KPI: fast, cheap unit cells where they are sufficient, and array models when the KPIs involve effects that unit cells cannot capture.

The layers are connected by the Allsolve SDK, which lets Python build and run models through APIs, so optimization loops can create designs, submit runs and pull back KPIs without opening the UI. Anything created through the SDK is still an Allsolve project that can be opened in the browser, so the models built by the AI can be reviewed and checked, which matters because AI can get things wrong.

Step 3: AI-driven optimization

Example: optimizing the unit-cell stack

The unit cell from earlier was extended with a second matching layer, giving six inputs for the optimizer: composite thickness, ceramic volume fraction, and the thickness and acoustic impedance of each of the two matching layers, each within specified ranges (which can be tailored to manufacturing tolerances or other restrictions). Two objectives were set: maximize peak transmit sensitivity and maximize −6 dB fractional bandwidth, which generally pull against each other. A constraint fixed the band center at just over 1 MHz with a ±8% tolerance; without it, the optimizer would find high-sensitivity designs whose frequency had doubled or halved.

Input	Range
Composite thickness	0.95–1.55 mm
Ceramic volume fraction	0.22–0.52
Inner matching thickness	0.25–0.70 mm
Inner matching impedance	3–6 MRayl
Outer matching thickness	0.25–0.70 mm
Outer matching impedance	1.5–3.5 MRayl

Objectives: maximize peak transmit sensitivity and −6 dB fractional bandwidth; band center held at about 1.05 MHz ±8%.

The optimizer generates candidates, builds and runs the Allsolve models through the API, retrieves the results, proposes the next candidates and converges on final designs. The resulting plot of sensitivity against fractional bandwidth shows every simulation as a gray dot, the original baseline as a red star, and the Pareto front as a purple line: the set of designs where bandwidth cannot be improved without giving up sensitivity, or vice versa. Points beyond the front are designs that failed the frequency constraint. Designers can then choose where on the front they want to sit.

Compared with the baseline, the design on the front nearest to it increased the −6 dB fractional bandwidth from 69% to 84% while also raising peak transmit sensitivity, from the same stack with only slight changes to layer thicknesses and impedances; the band center shifts slightly but stays within the gate. The natural next step is to test such a design in the array model, or to run the same optimization directly on the array model, which is feasible with cloud resources.

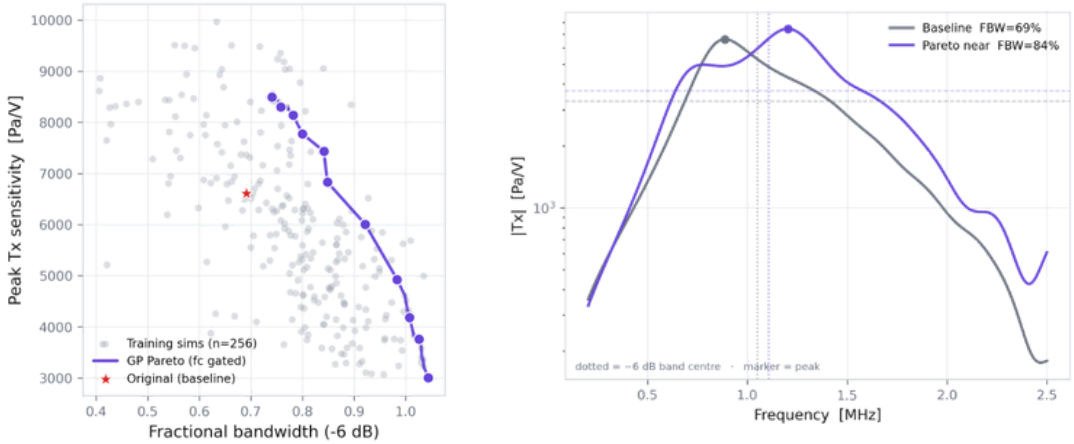


Figure 10 · Pareto front of 22 designs (left) and baseline vs selected Pareto design (right).

Questions and answers

Why 38 pillars, and how do you ensure the number of elements captures the physics while staying efficient?
The 38×38 pillars give a 19×19-element array. The size was not chosen for a particular physical reason; it was picked to give a computationally challenging model of about 10 million degrees of freedom that other software might struggle to run. It is by no means the top end of what Allsolve can handle.

Does extrapolation work for inhomogeneous media?
At the moment, extrapolation beyond the simulated domain assumes a homogeneous medium: a single material for which density and velocity are specified.

Can pulse simulations through media tens of wavelengths long be run efficiently in a single run?
Yes, this can be done with Allsolve.

How is the silicone fill in the matching layer defined: as a solid using velocities, or in pressure acoustics?
It is defined as an elastic material. Elastic materials in Allsolve can be defined by density together with longitudinal and shear velocities, or by Poisson's ratio and Young's modulus.

Which software was this?
Quanscient Allsolve, Quanscient's FE simulation software.

How large can simulations be with Quanscient Allsolve?
This model was about 9 million degrees of freedom; arrays in the 100 million degrees-of-freedom range have been run. With cloud compute, hundreds or thousands of cores can be used with domain decomposition. A colleague recently ran a purely acoustic simulation of one billion degrees of freedom on 3,500 cores, improved from around 12 minutes to under 3 minutes.

Would co-simulation with TX and RX electrical interfaces be possible?
Yes. The demo used a single transmit electrode, but multiple electrodes can transmit and receive separately, and circuits can be attached to them. More details on the specific case would be needed.

Does AI replace the physics in Allsolve?
No. In what was demonstrated, the AI optimizes the design (here the unit cell) by varying input parameters and using the KPIs returned by the physics simulation, sensitivity and bandwidth, to suggest new designs that are then built and rerun in Allsolve in a loop. The physics is always computed by the solver.

Key takeaways

- **Unit-cell results can miss what matters in a real array.** The unit cell remains the right tool for stack trade-offs: at about 19,000 degrees of freedom it solves in 39 seconds on 3 CPUs. But as an infinite periodic model it cannot show crosstalk, lateral modes, element directivity or beam steering.
- **One model, carried up the ladder.** Climbing from unit cell to array to optimization no longer means changing tools, rebuilding the model or waiting days for results; the array's real behavior can be predicted before it is built.
- **Trade-offs evaluated in simulation.** Sub-dicing the matching layer and filling the cuts with silicone rubber widened the −6 dB beam width from 53.8° to 65.7° while reducing peak sensitivity from 57.5 to 48.3; the decision can now be made before manufacturing.
- **Modeling the full, finite array is now practical.** A 19×19-element array with about 9.1 million degrees of freedom, built with the same stack and physics as the unit cell, solved in under an hour on 112 cores in Quanscient Allsolve.
- **Acoustic extrapolation.** Based on the Kirchhoff integral, it delivers on-axis pressure and far-field beam patterns at any distance without modeling a large water volume, and drops directly into an existing model.
- **AI-guided design builds on the physics, not instead of it.** Using the Allsolve SDK, an optimizer explored six stack parameters against two competing objectives under a band-center constraint and produced a Pareto front; the design nearest the baseline raised fractional bandwidth from 69% to 84% while also increasing sensitivity. With Allsolve running array models at 100 million degrees of freedom and acoustic models at a billion degrees of freedom on 3,500 cores in under 3 minutes, the same approach is feasible on large array models.

Get in touch

Everything shown in the webinar was built in Quanscient Allsolve, and the models are available. More webinars, including those on the Allsolve AI agent, can be found on the Quanscient website.

If what you saw resonated with your work, whether array-scale simulation or AI-led engineering, and you would like to discuss how it could apply to your specific use case, please get in touch with us.

Watch the webinar recording →



Jeff Dobson
Senior Application Engineer
info@quanscient.com



Jukka Knuutinen
Head of Marketing
jukka.knuutinen@quanscient.com

QUANSCIENT

quanscient.com

info@quanscient.com

linkedin.com/company/quanscient