

# Interactive Mathematics Beyond the Static Page

## Executive summary

The strongest resources for advanced mathematics are no longer just textbooks, CAS manuals, or code libraries in isolation. The best current systems combine mathematically serious content with one or more of the following: direct manipulation, browser-native visualization, executable notebooks, symbolic-numeric computation, or diagrammatic interfaces. In practice, the most mature ecosystems are concentrated in fields where local changes produce rich global structure: complex analysis, differential geometry, dynamical systems, low-dimensional topology, PDEs, and increasingly diagrammatic category theory. Official project pages and research papers show a clear pattern: the most effective tools are those that make invariants, singularities, parameter dependence, or rewrite structure visible while remaining close to real computation.

1

Several high-confidence conclusions stand out. First, complex analysis remains the exemplary success story: domain coloring and phase portraits turn four-real-dimensional function behavior into something legible on the domain itself, and resources such as [complex-analysis.com](http://complex-analysis.com), Needham's *Visual Complex Analysis*, and the recent Wolfram notebook-based complex-analysis materials still set the benchmark. Second, reactive notebooks and symbolic-numeric hybrids are now the dominant general-purpose delivery formats: Jupyter is still the most versatile open platform for research notebooks, Observable is best for browser-native interaction and visual polish, and Wolfram remains the most integrated symbolic-to-visual stack. Third, category theory and proof interaction have made unusually large gains because diagrammatic interfaces genuinely change what it feels like to "do" the subject. Fourth, operator theory and functional analysis remain the biggest under-served area: matrix pseudospectra are well supported, but genuinely interactive tools for semigroups, resolvents, continuous spectrum, Banach-space geometry, and operator-theoretic intuition remain sparse.

2

If one wants a practical buying decision without any budget constraint, the field currently divides into four best-in-class clusters. For maximal all-in-one power, use the Wolfram notebook ecosystem. For open, research-grade computational breadth, use Jupyter plus Python or Julia, with Julia especially strong in dynamics, PDEs, and interactive scientific visualization. For the cleanest browser-native interaction and publication, use Observable and WebGL/WebGPU pipelines. For expository animation rather than live exploration, use Manim. For advanced geometry, topology, and "mathematical making," the historical line from Geomview through Curved Spaces to contemporary XR and 3D-printing work is still the most imaginative part of the ecosystem.

3

## Why interactivity matters

At advanced level, interactivity matters because many mathematical objects are not just hard to compute; they are hard to see. Static exposition is excellent for logical structure, but weak at transmitting deformation, sensitivity, local-to-global propagation, and the behavior of families. The strongest interactive resources close this gap by making it possible to manipulate a parameter, drag a diagram, inspect a singularity, recompute a proof state, or watch a discretization converge. [complex-analysis.com](http://complex-analysis.com) explicitly

frames the computer as a laboratory for geometric reasoning; Jupyter foregrounds executable documents with rich outputs; Observable emphasizes reactive notebooks for visualization and experimentation; and ProofWidgets/Alectryon show that even proof states and formal derivations can now be rendered as manipulable interfaces rather than static transcripts. <sup>4</sup>

This matters especially in subjects where the representation itself is part of the insight. In complex analysis, zeros, poles, and branch structure become visible through phase plots and domain coloring. In dynamical systems, the difference between asymptotic stability and transient growth is often not visible from eigenvalues alone. In topology, a link editor or a triangulation interface changes what can be tested in real time. In applied category theory and higher rewriting, direct manipulation of string diagrams removes a large cognitive tax imposed by purely symbolic syntax. All of these are cases where interactivity is not cosmetic; it changes the accessible *form* of the mathematics. <sup>5</sup>

There is a methodological payoff as well. Many of the best resources are really instruments for experimental mathematics: one specifies equations symbolically, varies parameters, inspects geometry numerically, and then loops back to theorem-proof reasoning. Dedalus, FEniCSx, DynamicalSystems.jl, BifurcationKit.jl, and QuantEcon are explicitly built around this experiment-inspect-refine workflow. In this sense, interactivity matters not only for pedagogy, but for research practice itself. <sup>6</sup>

```
flowchart LR
  A[Static theorem-proof exposition] --> B[Executable notebooks]
  A --> C[Interactive geometry and visualization]
  B --> D[Reactive notebooks]
  B --> E[Symbolic-numeric hybrids]
  C --> E
  C --> F[Diagrammatic interfaces]
  E --> G[Research-grade explorable systems]
  F --> G
```

The main caution is equally important: visual and interactive systems can mislead when numerical discretization artifacts, rendering choices, or hidden defaults are confused with mathematics. The best resources counter this by exposing source code, providing references to papers, or keeping symbolic and numerical layers close together. That is a major reason mature ecosystems such as Wolfram notebooks, Jupyter, JuliaDynamics, Catlab, and the Lean widget/Alectryon family are more valuable than “pretty demo” sites that cannot be inspected or extended. <sup>7</sup>

## Paradigms of explorable mathematics

The current ecosystem is best understood not by subject first, but by *interaction paradigm*. The same paradigm often reappears across several branches of mathematics. <sup>8</sup>

Paradigm	What it does best	Canonical examples	Main mathematical payoff	Main risk	Core sources
Domain coloring / phase portraits	Encode complex-valued behavior directly on the domain	complex-analysis.com; Wegert-style phase portraits	Makes zeros, poles, branch behavior, argument structure, and conformality visually legible	Easy to over-read color artifacts or ignore scale issues	9
Reactive notebooks	Recompute all dependent views when one cell changes	Observable notebooks	Ideal for parameterized visualizations, dataflow thinking, interactive exposition	Browser-centric; weaker symbolic stack than CAS	10
Executable research notebooks	Mix code, text, equations, figures, widgets	Jupyter, JupyterLab	Best general-purpose format for reproducible computational mathematics	Notebook sprawl; reactivity less native than Observable	11
Symbolic-numeric hybrids	Keep formulas, solvers, and visuals in one environment	Mathematica/Wolfram, Dedalus, FEniCSx tutorials	Strongest bridge from algebraic specification to computation and visualization	Often steeper stack complexity or commercial lock-in	12
Diagrammatic reasoning	Represent composition and rewriting graphically	Catlab, Quantomatic, Globular, homotopy.io	Compresses categorical syntax into compositional geometry	Tooling fragmented; maturity varies sharply	13
Geometry / topology viewers	Navigate spaces, triangulations, and surfaces directly	Geomview, 3D-XplorMath, Curved Spaces, SnapPy	Excellent for manifolds, triangulations, non-Euclidean intuition	Legacy software and platform friction	14
Expository animation	Produce precise mathematical motion and staged explanations	Manim	Superb for polished explanation and lecture visuals	Usually less exploratory for end users than notebooks	15

Paradigm	What it does best	Canonical examples	Main mathematical payoff	Main risk	Core sources
Proof-aware documents	Render formal proof state as an explorable object	ProofWidgets, Alectryon	Makes theorem-prover interaction legible and publishable	Still concentrated in proof-assistant communities	16

## Curated catalog by subfield

Maturity labels below mean **established**, **mature**, **research-active**, **experimental**, or **legacy but valuable**. “Interactivity type” distinguishes browser applets, notebooks, GUI/CLI systems, shader/GPU systems, and proof/diagram interfaces.

**Complex analysis.** This is still the model case for advanced mathematical visualization because the core representation problem is so severe: a function on  $\mathbb{C}$  naturally lives in four real dimensions, and domain coloring or phase portraits partially solve that problem at the level of the domain itself. The current best stack combines one browser-native visual handbook, one classic geometric book, and one computational notebook ecosystem. 17

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience
complex-analysis.com	<a href="#">site</a>	Interactive visual book with applets on domain coloring, Riemann surfaces, conformal maps, and analytic landscapes	Mature; browser applets	Best all-around visual handbook; explicitly treats computation as a laboratory for reasoning	Not a complete standard-course replacement; some applets are desktop-oriented	Advanced undergrad through beginning graduate
<i>Visual Complex Analysis</i> by Tristan Needham	<a href="#">publisher</a>	Geometric print handbook for intuition	Established; non-executable, highly visual	Probably the best geometry-first complement to a rigorous course	Not interactive; less computational than modern notebook ecosystems	Advanced undergrad, instructors, self-learners

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience
<i>Visual Complex Functions</i> by Elias Wegert	<a href="#">publisher</a>	Systematic phase-portrait treatment	Established; visual monograph	Most systematic book-level treatment of phase portraits	More specialized than a first-course resource	Graduate students, researchers, visualizers
Wolfram Demonstrations + eTextbook + Wolfram U	<a href="#">Demonstrations</a> · <a href="#">eTextbook</a> · <a href="#">course</a>	Computational complex analysis with live notebooks, demonstrations, and exercises	Research-active; notebook + browser demos	Strongest symbolic-computational stack; easy transition from theory to executable examples	Commercial center of gravity; best experience often inside Wolfram stack	Students who want computation tightly integrated with theory
Domain coloring literature	<a href="#">IEEE overview</a> <a href="#">via PubMed</a>	Technical design principles for domain-coloring systems	Established; research literature	Best source for understanding what visual encodings actually mean	Not a teaching environment by itself	Builders of visual tools; instructors

**Differential geometry and manifolds.** The strongest resources here are split between legacy visualization systems, computational geometry-processing stacks, and a small number of unusually good immersive tools for non-Euclidean geometry. The most successful systems let the user *move in the space* or *compute on the geometry*, rather than merely look at a static rendering. <sup>23</sup>

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	Sources
Geomview	<a href="#">site</a>	Interactive viewing/manipulation of geometric objects, including non-Euclidean settings	Legacy but valuable; desktop GUI	Historically foundational; still conceptually important for geometry viewing and external program integration	Aging interface and workflow	Geometers, historians of math visualization, advanced hobbyists	<sup>24</sup>

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	Sources
3D-XplorMath	<a href="#">site</a>	Galleries of curves, surfaces, polyhedra, PDEs, fractals, with tunable parameters	Mature; desktop visual lab	Beautifully curated “experimental lab” model; unusually broad exhibit range	Desktop-first and somewhat dated operationally	Instructors, students, visual thinkers	25
Curved Spaces	<a href="#">site</a>	Flight-simulator exploration of multi-connected 3-manifolds	Mature; immersive GUI	One of the rare tools that gives geometric/topological intuition through navigation; explicitly useful up to research contexts	Focused niche; not a general DG toolkit	Geometry/topology students and researchers	26
Keenan Crane’s Discrete Differential Geometry course	<a href="#">course</a>	Computational entry point to curvature, geodesics, and surface geometry	Mature; course + code	Superb bridge between classical DG and geometry processing	Requires programming; less turnkey than applet-style tools	Advanced undergrads, grad students, computational geometers	27
geometry-central / libigl tutorial	<a href="#">geometry-central</a> · <a href="#">libigl</a> <a href="#">tutorial</a>	Research-grade geometry processing and mesh computation	Mature; code-centric	Best for users who want algorithms, data structures, and extensibility rather than canned demos	Less pedagogically guided for pure-geometry intuition	Researchers and technically strong students	28

**Dynamical systems and chaos.** This is the subfield where interactive computation is closest to a necessity rather than a luxury. Phase portraits, bifurcation diagrams, Lyapunov exponents, attractors, and continuation are natively exploratory objects. The best current tools are either Julia-based open research ecosystems or long-lived continuation packages with strong GUIs. <sup>29</sup>

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience
DynamicalSystems.jl	<a href="#">docs</a>	General nonlinear dynamics and time-series analysis	Mature; code + notebooks + visualization	Exceptionally broad coverage, strong docs, active development, research-grade extensibility	Requires Julia and some coding maturity	Advanced students through researchers
InteractiveDynamics.jl	<a href="#">docs</a>	GUI applications layered on DynamicalSystems.jl	Research-active; GUIs via Makie	Good example of “interactive scientific software” done right; easy to modify	Coupled to Julia/Makie ecosystem	Users already in Julia or willing to learn it
BifurcationKit.jl	<a href="#">docs</a>	Continuation and bifurcation analysis, including large-scale problems	Research-active; code-centric	Strong for continuation, stability, periodic orbits, GPUs, and large-scale systems	Less beginner-friendly than classic GUI tools	Researchers advanced dynamical-systems students
XPPAUT	<a href="#">overview</a>	Interactive ODE/delay/stochastic simulations with AUTO integration	Established; GUI + scripting	Extremely useful for quick phase-plane and bifurcation exploration; still common in practice	Legacy UX and ecosystem	Applied dynamical systems, mathematical biology, teaching
MatCont	<a href="#">project</a>	Numerical continuation/bifurcation in MATLAB	Established; GUI + MATLAB	Widely used, strong continuation workflow, interactive GUI	MATLAB dependency; narrower openness than Julia stack	Researchers who prefer MATLAB or established workflows

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience
ChaosBook.org	<a href="#">site</a>	Free online chaos text with videos and notebook-linked course materials	Established; hypertext + videos + notebooks	Still one of the most intellectually ambitious online chaos resources	Less software-like than Julia/MatCont stacks	Serious self-learners and graduate readers
Fractal Lab	<a href="#">site</a>	Real-time WebGL fractal exploration	Experimental but impressive; shader/GPU browser interface	Immediate intuition for Julia/Mandelbrot parameter dependence	Narrow topic focus; GPU/browser variability	Visual thinkers, chaos enthusiasts, teaching demos

**Topology and knot theory.** Here the mature tools are concentrated in low-dimensional topology, knot manipulation, triangulations, and TDA. There are fewer broad “interactive handbooks” than in complex analysis, but the flagship packages are very strong. <sup>37</sup>

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	Sources
SnapPy	<a href="#">docs</a>	Hyperbolic 3-manifolds, link editor, Dirichlet domains, cusp neighborhoods	Mature; GUI + CLI + Python	Best all-around computational environment for hyperbolic 3-manifolds and link-to-geometry workflows	Requires serious background to use deeply	Graduate students and researchers in low-dimensional topology	<sup>38</sup>
Regina	<a href="#">site</a>	Triangulations, normal surfaces, angle structures, knots/links	Mature; GUI + Python + C++	Excellent for combinatorial 3- and 4-manifold work; broad algorithmic support	Interface is more utilitarian than visual-artful	Low-dimensional topologists and computational researchers	<sup>39</sup>

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	Sources
KnotPlot	<a href="#">site</a>	Interactive knot and link visualization/manipulation	Established; desktop GUI	Still uniquely strong for geometric and visual knot manipulation	Less formal than Regina/SnapPy in theorem-oriented workflows	Knot theorists, artists, educators	40
giotto-tda	<a href="#">docs</a>	Persistent homology and TDA workflows	Mature; Python library	Strong bridge from topology to data analysis; integrates with modern ML workflows	More pipeline-oriented than visually geometric	TDA users, applied topologists, data scientists	41

**Category theory and diagrammatic mathematics.** This is one of the most exciting current fronts. The key shift is from symbolic term manipulation to diagrammatic composition and rewriting. These tools are not yet uniform in maturity, but the progress is real. <sup>13</sup>

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	Sources
Catlab.jl	<a href="#">docs</a>	Applied/computational category theory, wiring diagrams, typed symbolic systems	Research-active; code + notebook visualization	Best computational applied-category environment today; directly useful outside pure category theory	Not a proof assistant; requires Julia	Applied category theorists, compositional modelers	42
Quantomatic	<a href="#">project</a>	Diagrammatic reasoning and graph rewriting for monoidal theories	Legacy / maintenance-light; GUI + rewrite engine	Historically very important; still conceptually influential	No longer actively maintained	Researchers studying diagrammatic-reasoning tooling or ZX-style workflows	43

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	Sources
Globular	<a href="#">site</a>	Web-based proof assistant for higher-dimensional rewriting	Research-active; browser proof interface	High-impact concept: point-and-click higher-categorical proof construction with type checking	Specialized and narrower community	Higher category theory, rewriting, categorical logic	44
homotopy.io	<a href="#">site</a>	Browser proof assistant for finitely presented globular $n$ -categories	Experimental but serious; browser + WebAssembly	Direct geometric manipulation of proof objects; exports diagrams; unusually forward-looking	Still a young ecosystem	Higher category theorists, formalization researchers	45

**Spectral theory, operator theory, and functional analysis.** This is the most uneven area in the survey. There are excellent resources for matrix pseudospectra and transient growth, decent function-approximation/operator-computation libraries, and lots of PDE-adjacent work. What is missing is a broadly usable interactive layer for semigroups, resolvent geometry, spectral measures, continuous spectrum, and Banach-space phenomena. <sup>46</sup>

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	So
Pseudospectra Gateway	<a href="#">site</a>	Educational gateway to non-normality and pseudospectra	Established; web exposition	Best concise introduction to why spectra alone can fail for non-normal problems	More educational portal than live computational environment	Anyone entering non-normal spectral analysis	4
EigTool	<a href="#">site</a>	MATLAB GUI for pseudospectra, eigenvalues, field of values	Mature; GUI + MATLAB	Still the canonical interactive tool for matrix pseudospectra	Matrix/discretization centered; MATLAB dependency	Numerical linear algebra, hydrodynamic stability, control	4

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	So
Chebfun examples	<a href="#">examples</a>	Numerical experiments for operators, transient growth, resolvent norms	Mature; executable examples	Excellent for operator-flavored numerical intuition in function spaces	Not a dedicated spectral GUI; MATLAB-based	Analysts who want continuous analogues of matrix intuition	4
ApproxFun.jl	<a href="#">docs</a>	Function approximation and operator representations in Julia	Research-active; code-centric	Strong for operator representations at the function-space level	Sparse pedagogical visualization layer	Analysts and numerical approximation researchers	5
Trefethen–Embree pseudospectra literature	<a href="#">paper</a>	Conceptual anchor for non-normal operator intuition	Established; literature, not software	Still the central conceptual reference for why pseudospectra matter	Not interactive on its own	Operator theorists, numerical analysts	5

**PDEs and mathematical physics.** In this area, the most useful resources are symbolic-to-solver frameworks with strong notebook or visualization layers. Advanced intuition often comes from moving fluently among symbolic specification, discretization, and interactive post-processing. <sup>52</sup>

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	Sources
Dedalus	<a href="#">site</a>	Spectral PDE framework for IVPs/ BVPs/ eigenvalue problems	Mature; Python + notebooks	Symbolic equation specification with spectral methods; especially strong for fluid dynamics	Not a low-entry visual-first system	PDE researchers, computational physicists	53

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	Sources
FEniCSx tutorial + PyVista demo	<a href="#">tutorial</a> · <a href="#">PyVista demo</a>	Finite-element PDE solving with notebook-style examples and interactive visualization	Mature; Jupyter + interactive 3D viz	Strongest open FEM teaching/ research path in the survey	More engineering/ setup overhead than notebook-native systems	Applied mathematicians, mechanics, PDE students	54
QuTiP	<a href="#">guide</a>	Quantum dynamics and Bloch-sphere visualization	Mature; Python library + visual tooling	Very good for state-space intuition in low-dimensional quantum systems	Visualization is narrower than full PDE/field settings	Quantum mechanics students and researchers	55
Einstein Toolkit + kuibit	<a href="#">toolkit</a> · <a href="#">visualization</a>	Community platform for numerical relativity with visualization support	Mature; HPC software + post-processing tools	Serious research infrastructure; includes official visualization tooling	Heavyweight and specialized	Relativity and gravitational-physics researchers	56

**Probability and stochastic processes.** This area has fewer unified advanced “visual handbooks” than dynamics or complex analysis, but it has good notebook ecosystems and several strong simulation-based resources. The field’s current sweet spot is Markov chains, SDEs, and Brownian motion. 57

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	S
QuantEcon Markov chain lectures	<a href="#">site</a> · <a href="#">CTMC lectures</a>	Notebook-based intuition for finite and continuous-time Markov chains	Mature; notebook lectures	Clear bridge from rigorous stochastic theory to executable examples	Emphasis comes from economics/ quantitative modeling	Advanced undergrads, graduate students	

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience
SdePy	<a href="#">docs</a>	Simulation and numerical integration for Itô SDEs	Established; Python package	Clean entry to path simulation and statistics of realized paths	Less of a visual pedagogy layer	Stochastic calculus students, computational finance
DifferentialEquations.jl SDE tutorials	<a href="#">docs</a>	High-performance SDE solving in Julia	Mature; code-centric	Excellent solver ecosystem; naturally connects to the larger SciML stack	Less turnkey as a teaching-first handbook	Researchers and technically strong students
IMAGINARY's Colorful Brownian Motion	<a href="#">site</a>	Browser demo for Brownian-motion intuition	Research-adjacent outreach; browser simulation	Rare genuinely interactive probability visualization with nontrivial mathematical content	Simpler than a full advanced course	Students, instructors, outreach with serious content
Wolfram Brownian-motion demonstrations	<a href="#">example</a>	Interactive stochastic-process demos	Established; browser demo / notebook	Good for quick visual phenomena and classroom use	Better as complements than as a coherent curriculum	Instructors and exploratory learners

**Algebraic and tropical geometry.** The high-end interactive story here is still more fragmented than in geometry or dynamics. The strongest resources are CAS-centered and often become interactive only when paired with browser visualization, tropical/polyhedral backends, or 3D model export paths. <sup>63</sup>

Resource	Link	Purpose	Maturity / interactivity	Strengths	Limitations	Best audience	So
Macaulay2 + Visualize	<a href="#">Macaulay2</a> · <a href="#">Visualize</a>	Commutative algebra/ algebraic geometry with browser-based interactive visualization for graphs, posets, simplicial complexes, ideals	Mature; CAS + browser interaction	One of the clearest examples of algebraic computation becoming genuinely interactive in-browser	Visualization coverage is selective, not universal	Algebraic geometers, combinatorialists, instructors	6
polymake tropical application	<a href="#">docs</a>	Tropical hypersurfaces and tropical polytopes	Mature; code-centric, polyhedral-focused	Strongest general computational tropical-geometry environment in the survey	More computational than pedagogically visual	Tropical and polyhedral geometers	6
SageMath tropical geometry	<a href="#">docs</a>	Tropical varieties and tropical semirings in open-source CAS stack	Research-active; notebook/ CAS	Good open-source bridge to experimentation in tropical geometry	Visualization not as central as computation	Students and researchers wanting an open stack	6
Singular + surfex/ surfex.lib	<a href="#">Singular</a> · <a href="#">surfex</a> · <a href="#">docs</a>	Polynomial computations with algebraic-surface visualization support	Established; CAS + external visualization	Still important in singularity/ algebraic-surface workflows	Interface less modern and less integrated than notebook-first systems	Algebraic geometers, singularity theorists	6

## Computational ecosystems compared

The table below compares the major *delivery stacks* rather than individual libraries. The qualitative ratings in the last columns are synthesis judgments based on the cited official documentation and the surrounding project ecosystems. <sup>68</sup>

Ecosystem	Symbolic vs numeric	Visualization power	Extensibility	Research suitability	Licensing	Learning curve	Typical spot
Mathematica / Wolfram notebooks	Symbolic-first, but broad numeric and UI stack	Very high: dynamic graphics, notebook interfaces, built-in interactivity	High inside one integrated ecosystem	Very high, especially for rapid symbolic-visual prototyping	Proprietary / commercial notebook platform with viewing/deployment options	Medium	Fastest time from symbol to idea to polished interactive object
Julia ecosystem	Numeric-first with growing symbolic layers; excellent scientific computing	Very high with Makie and browser/GL backends	Very high; open package culture and strong performance	Very high in dynamics, PDEs, optimization, scientific visualization	MIT for Julia language; open-package ecosystem	Medium to high	Research grade computational math with performance and open matter
Observable	Browser/reactive JavaScript; weak symbolic core, strong frontend interaction	Extremely high for browser-native graphics and animation	High for web publishing and custom data apps	Medium to high; strongest when the mathematical object lives naturally in the browser	Hosted platform with free/pro plans; Framework is open-source ISC	Medium	Best for polished exploratory documents and visual essays
Jupyter	Language-agnostic notebook shell; symbolic or numeric depends on kernel	High, especially with widgets and plotting libraries	Very high; enormous extension ecosystem	Very high for reproducible research notebooks	BSD-3-Clause open source	Low to medium	Most versatile general research notebook platform
Manim	Programmatic animation, not a general CAS	High for pre-rendered mathematical storytelling	High if you are willing to script scenes	Medium; superb for exposition, less for live exploration	MIT open source	Medium	Lecture production explanatory videos, polished mathematical motion

Ecosystem	Symbolic vs numeric	Visualization power	Extensibility	Research suitability	Licensing	Learning curve	Typical spot
SageMath	Mixed CAS / numeric open stack built over many packages	Medium to high, especially through notebook integrations	High, but ecosystem integration can be uneven	High if open-source CAS breadth matters	GPL	Medium to high	Open-source alternative when algebraic number theory, and symbolic matter
WebGL / WebGPU pipelines	Visualization-first; symbolic layer external	Very high ceiling, especially for real-time interaction and shader-based systems	Very high for developers; low for non-programmers	High for bespoke mathematical visualization, lower for turnkey research workflows	Mixed: web standards and library-dependent	High	Custom browser-native visualization: fractals, immersive geometry, GPU den
Blender	Numeric/geometry content imported or scripted; not symbolic-first	Very high for 3D geometry, rendering, geometry nodes	High, especially for geometry pipelines and artistic production	Medium; strongest when mathematical insight benefits from high-end 3D form	GPL	High	3D modeling, mathematical art, topology, geometry presentations, XR assets

A practical reading of this table is straightforward. If the job is “I want the fastest path from formula to manipulable mathematical artifact,” Wolfram still has the most integrated answer. If the job is “I want transparent, extensible, research-grade open tooling,” Jupyter plus Python or Julia wins, with Julia in particular excelling when interactive numerics and performance matter. If the job is “I want the final artifact to live beautifully on the web,” Observable and WebGL/WebGPU are the right layer. If the job is “I want polished mathematical motion for explanation,” Manim remains unmatched. If the job is “I want embodied geometry or topology,” Blender and XR pipelines come into play. <sup>77</sup>

## Leading creators, communities, and emerging trends

A handful of creators and communities matter disproportionately because they are not merely making content; they are building reusable *mathematical media forms*. <sup>78</sup>

The most influential public-facing creator in this space is **Grant Sanderson** <sup>79</sup> of 3Blue1Brown, whose work helped normalize the idea that higher mathematics can be communicated as precise animated geometry rather than static blackboard prose. The related Manim ecosystem turned that style into a reusable toolchain. On the open-exhibit side, **IMAGINARY** <sup>80</sup> has become one of the most important cross-field platforms for interactive mathematical exhibits and openly reusable content. In open scientific software, **JuliaDynamics** <sup>81</sup> is one of the clearest examples of a research community building not just code

but pedagogy, meetings, and a coherent visual-computational workflow. For applied category theory, **AlgebraicJulia** <sup>82</sup> is distinctive because it pushes category theory toward computational practice rather than purely expository metaphor. In geometry/topology and mathematical making, **Henry Segerman** <sup>83</sup> represents the strongest current synthesis of topology, 3D printing, and XR. And for notebook-centric computation, **Wolfram Research** <sup>84</sup> and **Project Jupyter** <sup>85</sup> remain the two most consequential institutional poles. <sup>86</sup>

The leading emerging trends are also clear. AI-assisted notebook interaction is becoming real rather than aspirational: Wolfram’s Notebook Assistant explicitly promises natural-language help for code, computation, and interactive visualizations. Interactive proof interfaces are maturing from niche experiments into usable tooling: ProofWidgets makes Lean proof interaction visibly richer, while Alectryon turns proof scripts into explorable documents. Browser-native higher-category tools such as Globular and homotopy.io show that diagrammatic, direct-manipulation proof environments are now practical in specialized domains. At the geometry end, XR work such as Segerman’s curved-space simulations and the broader literature on manifold VR suggest that immersive mathematics is moving from novelty toward a serious niche medium.

<sup>87</sup>

What has *not* yet happened is equally revealing. There is still no comparably mature interactive layer for measure theory, abstract functional analysis, representation theory, homological algebra, or operator algebras. Where interactivity exists, it is usually mediated through discretization, PDE solvers, or matrix models. That asymmetry is one of the strongest findings of this survey. <sup>88</sup>

## Tailored recommendations

Because there is no single best stack, the right choice depends sharply on intellectual style and task.

User type	Recommended starting stack	Why this stack fits	Sources
Pure mathematician who wants visual intuition without losing rigor	complex-analysis.com, Needham, SnapPy/Regina, Curved Spaces, Catlab/Globular where relevant	Best mix of serious mathematics and nontrivial visualization without becoming “just software”	<sup>89</sup>
Visual thinker / geometer	3D-XplorMath, Geomview, Curved Spaces, Blender, Segerman’s 3D-print/XR ecosystem	Strongest route to embodied and navigable geometry	<sup>90</sup>
Operator theorist / functional analyst	Pseudospectra Gateway, EigTool, Chebfun, ApproxFun.jl, then PDE-adjacent notebooks	This is currently the least complete stack, but it is the best existing bridge from abstraction to computation	<sup>46</sup>
Dynamical-systems researcher	DynamicalSystems.jl + InteractiveDynamics.jl + BifurcationKit.jl; XPPAUT/MatCont if GUI-first	Best present balance of research depth, interactivity, and extensibility	<sup>91</sup>

User type	Recommended starting stack	Why this stack fits	Sources
PDE / mathematical-physics user	Dedalus, FEniCSx, Jupyter, QuTiP, Einstein Toolkit when domain-specific	Strong symbolic-to-solver-to-visualization workflows	92
Category theorist / rewriting enthusiast	Catlab first; then Globular and homotopy.io; treat Quantomatic as historical background	Best current ladder from applied categorical computing to higher-diagrammatic proof manipulation	93
Educator building explorable materials	Jupyter for open distribution, Observable for publication polish, Manim for prepared explanation, IMAGINARY for reusable exhibits	Covers classroom notebooks, web essays, animations, and public exhibits	94
Interdisciplinary scientist	Wolfram if integration speed matters; Julia/Jupyter if openness and long-term extensibility matter	These two poles still dominate practical tradeoffs	95

For a single “starter bundle” at advanced-undergraduate level, I would recommend one resource from each of four categories: a visual handbook, an executable notebook platform, a subfield-specific computational system, and an expository/communication layer. Concretely: complex-analysis.com or Needham; Jupyter or Wolfram notebooks; one of SnapPy, DynamicalSystems.jl, Dedalus, or Catlab depending on subject; and Manim or Observable for publication-quality output. That combination gives both understanding and production. <sup>96</sup>

## Gaps, open questions, and conclusion

The most consequential gap is in operator theory and functional analysis. The ecosystem currently has excellent resources for matrix non-normality, pseudospectra, and some function-space numerics, but very little that feels to operator theory the way complex-analysis.com feels to complex analysis or Curved Spaces feels to manifold intuition. The existing sources themselves point to the problem: the pseudospectra literature and Gateway explicitly acknowledge the extension from matrices to operators while simplifying to finite-dimensional settings for technical reasons, and the main interactive tools remain largely discretization-first. <sup>97</sup>

That suggests unusually clear opportunities. The field would benefit from browser-based resolvent explorers, semigroup-growth visualizers, spectral-measure and functional-calculus demos, Hilbert/Banach geometry interfaces, and notebook libraries that make continuous-spectrum phenomena visible without burying the user in numerical analysis plumbing. There is also room for proof-aware exposition in advanced analysis: a hybrid of Alectryon/ProofWidgets-style explorable documents with operator-theoretic visual layers would be genuinely novel. <sup>98</sup>

A second open question is institutional rather than technical. Many powerful resources are still tied either to single creators, niche labs, or volunteer-maintained academic software. This produces brilliance, but also fragility. Quantomatic’s maintenance status is a reminder that elegant research software can become

historically important before it becomes infrastructurally stable. The healthiest ecosystems are the ones that already combine code, docs, examples, and community governance: Jupyter, the Wolfram notebook ecosystem, JuliaDynamics, FEniCSx, Dedalus, SnapPy, and IMAGINARY. <sup>99</sup>

The broad conclusion is that interactive mathematics is no longer peripheral. It is becoming a genuine parallel medium for advanced mathematical thought, somewhere between textbook, lab instrument, and proof assistant. The best systems do not replace rigor; they reorganize access to rigor by making hidden structure perceptible, manipulable, and reproducible. The next major gains are likely to come where the current ecosystem is weakest: abstract analysis, operator theory, and proof-integrated mathematical exposition. If those areas catch up to what complex analysis, geometry, and dynamical systems already have, the result will not just be better pedagogy. It will be a new normal for how advanced mathematics is explored and communicated. <sup>100</sup>

---

<sup>1</sup> <sup>4</sup> <sup>9</sup> <sup>17</sup> <sup>18</sup> <sup>83</sup> <sup>89</sup> <sup>96</sup> <sup>100</sup> <https://complex-analysis.com/>  
<https://complex-analysis.com/>

<sup>2</sup> <sup>20</sup> <sup>84</sup> <https://link.springer.com/book/10.1007/978-3-0348-0180-5>  
<https://link.springer.com/book/10.1007/978-3-0348-0180-5>

<sup>3</sup> <sup>7</sup> <sup>12</sup> <sup>68</sup> <sup>69</sup> <sup>77</sup> <sup>95</sup> <https://www.wolfram.com/mathematica/>  
<https://www.wolfram.com/mathematica/>

<sup>5</sup> [https://complex-analysis.com/content/domain\\_coloring.html](https://complex-analysis.com/content/domain_coloring.html)  
[https://complex-analysis.com/content/domain\\_coloring.html](https://complex-analysis.com/content/domain_coloring.html)

<sup>6</sup> <sup>52</sup> <sup>53</sup> <sup>92</sup> <https://dedalus-project.org/>  
<https://dedalus-project.org/>

<sup>8</sup> [https://www.mi.fu-berlin.de/en/math/groups/ag-geom/publications/db/ieee\\_article\\_old\\_low\\_v3\\_1.pdf](https://www.mi.fu-berlin.de/en/math/groups/ag-geom/publications/db/ieee_article_old_low_v3_1.pdf)  
[https://www.mi.fu-berlin.de/en/math/groups/ag-geom/publications/db/ieee\\_article\\_old\\_low\\_v3\\_1.pdf](https://www.mi.fu-berlin.de/en/math/groups/ag-geom/publications/db/ieee_article_old_low_v3_1.pdf)

<sup>10</sup> <sup>71</sup> <https://observablehq.com/>  
<https://observablehq.com/>

<sup>11</sup> <sup>72</sup> <sup>94</sup> <https://jupyter.org/>  
<https://jupyter.org/>

<sup>13</sup> <sup>42</sup> <sup>93</sup> <https://algebraicjulia.github.io/Catlab.jl/dev/>  
<https://algebraicjulia.github.io/Catlab.jl/dev/>

<sup>14</sup> <sup>23</sup> <sup>24</sup> <https://www.geomview.org/>  
<https://www.geomview.org/>

<sup>15</sup> <sup>73</sup> <sup>80</sup> <https://www.manim.community/>  
<https://www.manim.community/>

<sup>16</sup> <sup>98</sup> <https://github.com/leanprover-community/ProofWidgets4>  
<https://github.com/leanprover-community/ProofWidgets4>

<sup>19</sup> <sup>85</sup> <https://global.oup.com/academic/product/visual-complex-analysis-9780192868916>  
<https://global.oup.com/academic/product/visual-complex-analysis-9780192868916>

- 21 <https://www.wolfram-media.com/products/essentials-of-complex-analysis/>  
<https://www.wolfram-media.com/products/essentials-of-complex-analysis/>
- 22 <https://pubmed.ncbi.nlm.nih.gov/24806991/>  
<https://pubmed.ncbi.nlm.nih.gov/24806991/>
- 25 90 <https://3d-xplormath.org/>  
<https://3d-xplormath.org/>
- 26 <https://www.geometrygames.org/CurvedSpaces/index.html.en>  
<https://www.geometrygames.org/CurvedSpaces/index.html.en>
- 27 <https://www.cs.cmu.edu/~kmcrane/Projects/DDG/>  
<https://www.cs.cmu.edu/~kmcrane/Projects/DDG/>
- 28 <https://geometry-central.net/>  
<https://geometry-central.net/>
- 29 30 91 <https://juliadynamics.github.io/DynamicalSystems.jl/dev/>  
<https://juliadynamics.github.io/DynamicalSystems.jl/dev/>
- 31 <https://juliadynamics.github.io/InteractiveDynamics.jl/dev/dynamicalsystems/>  
<https://juliadynamics.github.io/InteractiveDynamics.jl/dev/dynamicalsystems/>
- 32 <https://bifurcationkit.github.io/BifurcationKitDocs.jl/dev/>  
<https://bifurcationkit.github.io/BifurcationKitDocs.jl/dev/>
- 33 <https://sites.pitt.edu/~phase/bard/bardware/xpp/whatis.html>  
<https://sites.pitt.edu/~phase/bard/bardware/xpp/whatis.html>
- 34 <https://sourceforge.net/projects/matcont/>  
<https://sourceforge.net/projects/matcont/>
- 35 <https://www.chaosbook.org/>  
<https://www.chaosbook.org/>
- 36 <https://hirnsohle.de/test/fractalLab/>  
<https://hirnsohle.de/test/fractalLab/>
- 37 38 <https://snappy.computop.org/>  
<https://snappy.computop.org/>
- 39 <https://regina-normal.github.io/>  
<https://regina-normal.github.io/>
- 40 <https://knotplot.com/>  
<https://knotplot.com/>
- 41 <https://giotto-ai.github.io/gtda-docs/latest/modules/homology.html>  
<https://giotto-ai.github.io/gtda-docs/latest/modules/homology.html>
- 43 99 <https://github.com/zxcalc/quantomatic>  
<https://github.com/zxcalc/quantomatic>
- 44 <https://globular.science/>  
<https://globular.science/>

45 <https://arxiv.org/abs/2402.13179>

<https://arxiv.org/abs/2402.13179>

46 47 88 97 <https://www.cs.ox.ac.uk/pseudospectra/intro.html>

<https://www.cs.ox.ac.uk/pseudospectra/intro.html>

48 <https://www.cs.ox.ac.uk/pseudospectra/eigtool/>

<https://www.cs.ox.ac.uk/pseudospectra/eigtool/>

49 <https://www.chebfun.org/examples/linalg/TransientGrowth.html>

<https://www.chebfun.org/examples/linalg/TransientGrowth.html>

50 79 <https://juliaapproximation.github.io/ApproxFun.jl/latest/usage/operators/>

<https://juliaapproximation.github.io/ApproxFun.jl/latest/usage/operators/>

51 [https://people.maths.ox.ac.uk/trefethen/publication/PDF/1997\\_72.pdf](https://people.maths.ox.ac.uk/trefethen/publication/PDF/1997_72.pdf)

[https://people.maths.ox.ac.uk/trefethen/publication/PDF/1997\\_72.pdf](https://people.maths.ox.ac.uk/trefethen/publication/PDF/1997_72.pdf)

54 <https://jsdokken.com/dolfinx-tutorial/>

<https://jsdokken.com/dolfinx-tutorial/>

55 <https://qutip.readthedocs.io/en/latest/guide/guide-bloch.html>

<https://qutip.readthedocs.io/en/latest/guide/guide-bloch.html>

56 <https://einsteintoolkit.org/>

<https://einsteintoolkit.org/>

57 <https://continuous-time-mcs.quantecon.org/intro.html>

<https://continuous-time-mcs.quantecon.org/intro.html>

58 <https://quantecon.org/>

<https://quantecon.org/>

59 82 <https://sdepy.readthedocs.io/en/v1.1.2/intro.html>

<https://sdepy.readthedocs.io/en/v1.1.2/intro.html>

60 81 [https://docs.sciml.ai/DiffEqDocs/stable/tutorials/sde\\_example/](https://docs.sciml.ai/DiffEqDocs/stable/tutorials/sde_example/)

[https://docs.sciml.ai/DiffEqDocs/stable/tutorials/sde\\_example/](https://docs.sciml.ai/DiffEqDocs/stable/tutorials/sde_example/)

61 <https://www.imaginary.org/program/colorful-brownian-motion>

<https://www.imaginary.org/program/colorful-brownian-motion>

62 <https://demonstrations.wolfram.com/BrownianMotionIn2DAndTheFokkerPlanckEquation/>

<https://demonstrations.wolfram.com/BrownianMotionIn2DAndTheFokkerPlanckEquation/>

63 <https://macaulay2.com/doc/Macaulay2/share/doc/Macaulay2/Visualize/html/index.html>

<https://macaulay2.com/doc/Macaulay2/share/doc/Macaulay2/Visualize/html/index.html>

64 <https://macaulay2.com/doc/Macaulay2/share/doc/Macaulay2/Macaulay2Doc/html/>

<https://macaulay2.com/doc/Macaulay2/share/doc/Macaulay2/Macaulay2Doc/html/>

65 <https://polymake.org/doku.php/documentation/latest/tropical>

<https://polymake.org/doku.php/documentation/latest/tropical>

66 [https://doc.sagemath.org/html/en/reference/polynomial\\_rings/sage/rings/semirings/tropical\\_variety.html](https://doc.sagemath.org/html/en/reference/polynomial_rings/sage/rings/semirings/tropical_variety.html)

[https://doc.sagemath.org/html/en/reference/polynomial\\_rings/sage/rings/semirings/tropical\\_variety.html](https://doc.sagemath.org/html/en/reference/polynomial_rings/sage/rings/semirings/tropical_variety.html)

67 <https://www.singular.uni-kl.de/>

<https://www.singular.uni-kl.de/>

70 <https://julialang.org/>

<https://julialang.org/>

74 <https://www.sagemath.org/>

<https://www.sagemath.org/>

75 <https://www.khronos.org/webgl/>

<https://www.khronos.org/webgl/>

76 <https://www.blender.org/about/license/>

<https://www.blender.org/about/license/>

78 86 <https://www.3blue1brown.com/about/>

<https://www.3blue1brown.com/about/>

87 <https://www.wolfram.com/wolfram-u/courses/notebooks/learn-to-use-notebook-assistant-nb004/>

<https://www.wolfram.com/wolfram-u/courses/notebooks/learn-to-use-notebook-assistant-nb004/>