

Phoenix: A Novel Context-Aware Voice-Powered Math Equation Workspace and Editor

Kenneth Ge*

Assistivity

Edgemont, NY, USA

kennethkouge@gmail.com

Priscilla Zhang*

Assistivity

Villanova, PA, USA

zhang.priscilla34@gmail.com

Ryan Paul*

Assistivity

Brampton, ON, Canada

paulryan229@gmail.com

JooYoung Seo

School of Information Sciences

University of Illinois Urbana-Champaign

Champaign, Illinois, USA

jseo1005@illinois.edu

ABSTRACT

Writing mathematical notation requires substantial effort, diverting cognitive resources from conceptual understanding to documentation mechanics, significantly impacting individuals with fine motor disabilities (FMDs). Current limits of speech-based math technologies rely on precise dictation of math symbols and unintuitive command-based interfaces. We present a novel voice-powered math workspace, applying neuroscience insights to create an intuitive problem-solving environment. To minimize cognitive load, we leverage large language models with our novel context engine to support natural language interaction. Ultimately, we enable fluid mathematical engagement for individuals with FMDs—freed from mechanical constraints

CCS CONCEPTS

- Human-centered computing → Accessibility design and evaluation methods; Accessibility technologies; Accessibility systems and tools; Empirical studies in accessibility.

KEYWORDS

Accessibility, Large Language Models, Generative AI, Multimodality, Fine Motor Disability

ACM Reference Format:

Kenneth Ge, Ryan Paul, Priscilla Zhang, and JooYoung Seo. 2025. *Phoenix: A Novel Context-Aware Voice-Powered Math Equation Workspace and Editor*. In *The 27th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '25), October 26–29, 2025, Denver, CO, USA*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3663547.3759766>

1 INTRODUCTION

An injury six years ago rendered both my wrists intermittently unusable. Despite previously excelling in math and science, I experienced the limits of mathematical voice-to text software firsthand. Existing tools were unpredictable, translating my speech into linear fragments of

English and mathematical notation, and failing to capture nested and multidimensional equations. Even solving simple algebra problems required extra time and energy, without features that understood commands like "divide both sides by 2." Tasks previously requiring only insight and reasoning were frequently interrupted and constrained by translational inaccuracies and mechanical obstacles. My progress slowed, not due to differences in intellect or effort, but a lack of technical tools.

P's experiences—as a co-author on our mixed-ability design team—reflect those of many individuals with fine motor disabilities (FMDs), and directly inspired *Phoenix*'s development. Although foundational in STEM [4, 10, 17, 23], mathematical concepts and notation pose unique accessibility challenges due to the need to manipulate specialized symbols, spatially arrange formulas, and structure multilevel equations [11, 12]. Such processes often require handwriting or typing, placing high demands on physical dexterity [16, 34, 35].

Barriers emerge as early as preschool, where fine motor skills are strong predictors of mathematical success [9, 25]. Throughout schooling, understaffed disability services, rigid service protocols, and lack of faculty awareness compound previous difficulties [1, 8, 37]. Although scribes help accessibility, mismatches in packing and interpretation divert attention from core concepts [26, 30]. Fixed mindsets around math and widespread "math anxiety" particularly harm individuals with FMDs, who may internalize a sense of inadequacy and disengage due to reinforcing cycles [7, 14, 22, 27].

Existing solutions require users to dictate rigid grammar structures, introducing steep learning curves and challenging navigational demands. However, neuroscience research suggests that mathematical ability comes from symbolic and spatial reasoning, and is distinct from motor ability [15, 18, 32, 33].

Building on these insights, we present *Phoenix*, a voice-based math workspace designed from the ground up to minimize cognitive overhead. Similar to the mythical creature that is reborn from its ashes, *Phoenix* enables users to overcome physical obstacles and find renewal in mapping mathematical ideas in a novel way. We integrate structure, context, and naturally spoken mathematics by leveraging recent advances in large language models (LLMs).

*All three authors contributed equally to this research.

We aim to transform mathematical problem solving into an intuitive and high-level process for users at all levels, from students to professionals. We focus on the following research questions:

- How can we help users with fine motor disabilities intuitively map out mathematical ideas?
- How can we enable new interaction paradigms and enhanced understanding through robust voice input?
- How can an editor optimized for cognitive load enable new levels of abstract problem solving?

2 RELATED WORK

As most FMD users prefer voice input modalities [19], speech-based math editors demonstrate accessibility potential. Early systems like *MathTalk* used structured pauses to convey meaning [28], while *CamMath* and *TalkMaths* progressed towards natural classroom-style speech [6, 36]. *Mathifier* further introduced a grammar, albeit highly structured and code-like [3]. Although these approaches establish voice as effective input, they introduce cognitive overhead by requiring specialized vocabularies and rigid grammar structures, enforcing protocols such as the NATO phonetic alphabet and introducing subtle timing errors. Fundamentally, such approaches do not accommodate natural speech variations. Recent LLM-powered tools (e.g. *MathGPT Pro*, *Mathful*, *StudyFetch*) demonstrate improved contextual understanding in highly structured tutoring contexts, but do not offer freeform mathematical authoring [20, 21, 29].

In our research, the closest existing tool to *Phoenix* is *EquatIO* [31], which enables robust voice input [24]. Although existing studies did not rigorously test this feature, as a regular user, P noticed it often created incomplete translations, demonstrating limited contextual understanding (see Appendix A for test details). Ultimately, while *EquatIO* is useful and feature-rich, it seemingly treats voice as one input modality among many.

We introduce *Phoenix*, a mathematical input system designed for FMD users to achieve high levels of flexibility and robustness, enabling new levels of abstract problem-solving. By combining LLM-powered contextual understanding with speech recognition, we eliminate the need for specialized vocabularies, enabling an intuitive authoring experience.

3 DESIGN PROCEDURES AND GOALS

3.1 Design Procedures

We applied the interdependence framework to our mixed-ability co-design process [5]. Our team consists of four researchers: one researcher, P, with a FMD and three researchers without an FMD (including one with visual impairment). P utilized English voice dictation software for 5 years, encountering limitations with commercial math dictation tools. Each member of the team has over 5 years of experience with LaTeX.

Our work consisted of bi-weekly online testing sessions from January 2025 to June 2025, during which P initially served primarily as a tester. These sessions focused on overarching goals and identifying areas to streamline Phoenix. Specific activities included testing low- and medium-fidelity prototypes through role-playing, conducting usability testing to refine features based on P's insights, and drawing on our broader experiences to creatively close gaps. Feedback was gathered through group discussions and recorded

think-aloud sessions, ensuring P's experiences directly shaped usability and design decisions. This feedback was then iteratively incorporated by adjusting prototypes, which we later validated in co-design workshops. In these workshops, P's role expanded from evaluating designs to actively generating them. Together, we brainstormed features, sketched new interface ideas, and explored alternative interaction flows, using insights from previous testing sessions as a foundation for the next iteration. This process resulted in a functional prototype by June.

3.2 Design Goals

Our fundamental philosophy is to harness conceptual ability through hands-free entry, reducing cognitive load. We aim to create space for deeper mathematical engagement by minimizing fine-motor input and its accompanying obstacles. Our design sessions – grounded in this philosophy – gave rise to four key design goals (DGs).

DG1: *Phoenix* should interpret a wide range of spoken mathematical expressions, symbols, and notation. Existing tools require users to learn narrow dictation rules, undermining natural speech variations. P expressed how frequently inaccurate transcriptions interrupted workflow, hindering full engagement in high-level mathematics and slowing progress. Thus, we aimed to interpret variability in spoken mathematics, reliably recognizing mathematical dictation in real-world settings.

DG2: *Phoenix* should enable flexible equation editing and exporting. *Phoenix* should allow various forms of equation editing without navigating complex commands or interfaces. P expressed the need for voice-powered equation editing and exporting work to various platforms like Microsoft Word. Thus, we aimed to enable various forms of equation editing and exporting to change variables or share work, ultimately improving problem-solving.

DG3: *Phoenix* should intuitively convey logical flow of mathematical relationships. Mathematical problem-solving resembles a web, as complex problems are broken down into smaller, interrelated subcomponents [13]. Supporting this process, *Phoenix* should help users visually map ideas, making solutions easier to understand and manage. To this end, we aimed to make *Phoenix* both customizable and self-organizing.

DG4: *Phoenix* should be flexible to the user's problem-solving style. *Phoenix* offers structure for many basic problem-solving features. P expressed how most helpful tools allowed flexibility over rigid usage patterns. Thus, we aimed to maximize flexibility by allowing users to personalize and customize *Phoenix*'s behavior.

4 SYSTEM IMPLEMENTATION

To address the design goals established during our co-design sessions, we developed a prototype of the *Phoenix* system. In this section, we first provide a detailed explanation of end-user system interaction, followed by a technical description of system architecture. Figure 1 provides a high-level overview of the front-end interface. A detailed evaluation of *Phoenix*'s voice transcription accuracy using benchmark prompts is available in subsection A.3.

4.1 Overview

Context Engine (addressing DG 1) Our most important innovation is the context engine, which transcribes the user's equation

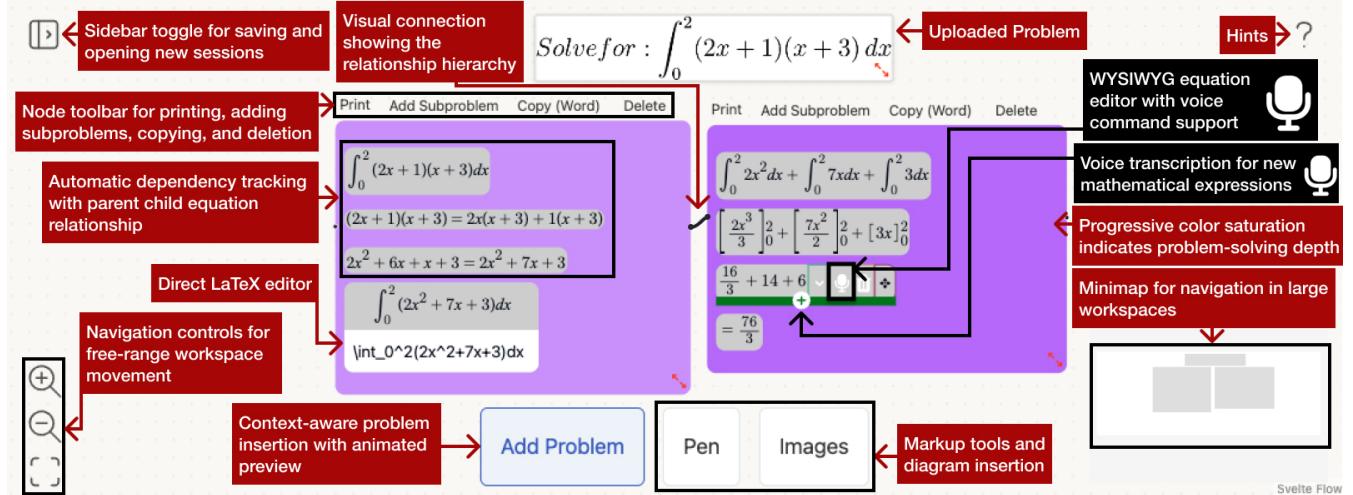


Figure 1: System Overview Diagram

while borrowing context from previous transcriptions (including variable names and equation fragments), well-known equations (e.g. the quadratic equation), and domain-specific vocabulary within STEM. Further, this engine accurately transcribes high-level STEM terminology including integrals, partial derivatives, and Greek letters. Ultimately, this engine achieves at least 3 advanced capabilities: (A) Voice-powered equation editing through commands such as “change the plus to a minus”, “substitute ‘x’ with 7”, or “move the denominator to the numerator”, (B) rendering varied expressions, such as “x over 3,” “x divided by 3,” and “x by 3,” to all yield the same final expression, and (C) isolating math from conversational speech; if a user dictated “I was thinking about the integral of e to the negative x squared”, only the math is transcribed.

Self-Contained Subproblem Nodes (addressing DG2) In *Phoenix*, each subproblem forms a self-contained equation space. New subproblems start empty, with just a plus button to transcribe the first equation. An additional collapsible toolbar enables: print the subproblem, add a connected subproblem, delete the node, and copy the contents in Word or LaTeX format, complete with optional step-by-step annotations. Each equation is fully movable and arranged according to user preference (top-to-bottom by default). Equations can be edited by voice, with MathQuill (WYSIWYG) and LaTeX as fallbacks.

Graph-Based Workspace Design (addressing DG3) Complex problems contain independent subcomponents, and successful problem solvers use ample space to map ideas in a sensorimotor web [2]. Since P expressed the importance of visually mapping equation subproblems, *Phoenix* is graph-based, separating nodes and using color saturation to depict problem hierarchy. For example, a user may dictate, “integral from zero to two of $(2x + 1)(x + 3)$,” which the system parses into a root node. Additional nodes are added as users integrate step-by-step, allowing modular, branching thinking. Animated actions reduce visual complexity; hovering over “Add Problem” previews subproblem location.

Digital Thinking (addressing DG4) Beyond transcription, *Phoenix* supports active, non-linear mathematical thinking. Features like diagram/image uploads allow users to incorporate problems from other platforms. Pen-markup contains various pen colors and thicknesses, enabling creative interaction. Users can also save and load work for future reference.

4.2 Architecture

We built *Phoenix* with a Svelte frontend, a context engine that builds LLM queries, and a Node.js backend.

Client Side Data Structures We use persistent storage to store substantial data on the client side, such as images, markup, documents, and user preferences. We designed a top-down data structure to store subproblem nodes, including a custom JSON-based data structure tracking equation data and positioning, background color, and size. We used a robust vector-based approach for markup, saving a list of individual paths (rather than pixels), rendering only those visible onscreen. Finally, we store user preferences on the client side in a persistent store.

Export Algorithms To help users share and export their work, we developed a “print” function, and integrations to copy work into LaTeX and Microsoft Word. The print feature loads a new popup with all equations for a given subproblem in a tabular format, renders them using MathJax, and automatically triggers a system print dialogue upon successful render. We built a one-click button to copy annotated steps directly in LaTeX, formatting equations into a specially designed template, leveraging the system clipboard.

The most challenging export algorithm was integrating with Microsoft Word. Word uses a restricted subset of MathML, normally making it impossible to copy more than one equation to the clipboard (this will appear as text, rather than an equation object). To overcome this barrier, we leveraged the table feature to copy multiple equations at once, using ‘mtext’ elements to annotate steps.

Server Side User Management Our backend implements authentication using Google OAuth and Json Web Tokens (JWTs),

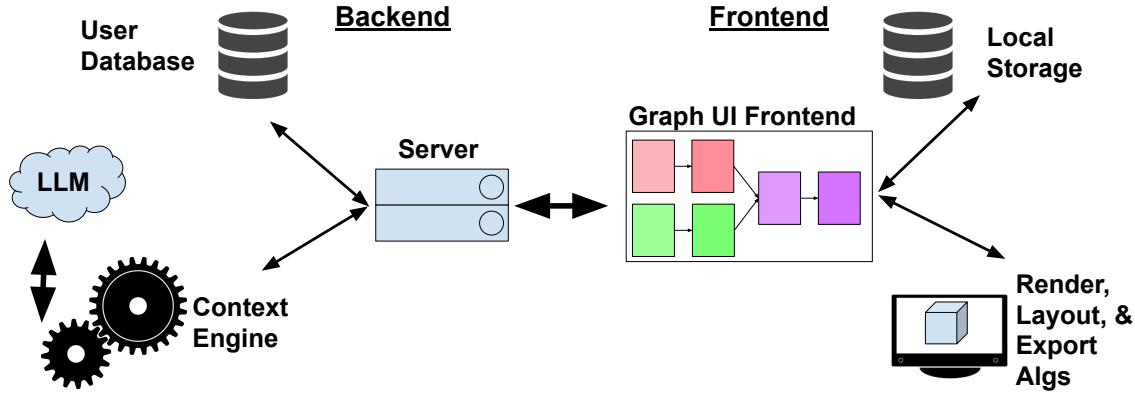


Figure 2: Architecture overview, showing the flow of data from the graph-based frontend to the backend, context engine, and beyond

managing server-side user info for the context engine, and rate limiting to ensure fair distribution of limited API queries. To further ease backend load, we use the Web Speech API to transcribe speech into text, before sending the text to the context engine to generate equations. Unlike prior work, we do not need a custom speech-to-text model because our context engine flexibly infers important details and correct mistakes.

Context Engine The context engine is applied whenever the user edits an equation or generates a new one via voice (i.e. transcription or transformation). Context is managed as soon as the user interacts with the page. In addition to the overt dependencies *between* subproblems, *Phoenix* also maintains an implicit dependency order among the equations *within* each subproblem. When users transcribes a new equation, *Phoenix* automatically tracks its “parent” equation on creation. All this information is sent to the context engine.

Using these dependencies, the engine runs a graph traversal to order equations from most to least relevant. This graph traversal calculates a distance-weighted “fuzzy usefulness” to weight the context of each step. Unlike approaches like RAG, our graph traversal retains full detail, is more interpretable, and allows greater control over the subject domain.

After ordering the dependencies, we attach them to our LLM prompt, instructing the model to consider most relevant context. To generate transcriptions, we use few-shot prompting by providing example inputs and outputs to help the model understand how to handle complex commands and queries involving domain-specific knowledge (e.g. physics terminology). The result is then retrieved and sanitized by a regular expression before being sent back to the client.

5 FUTURE DIRECTIONS

Our next step is focused on helping *Phoenix* more fully realize our design goals. Future work will target universal voice control, improving occasionally inaccurate complex STEM equation transcription (e.g. quantum mechanics), respecting user preference for default lowercase letters, and enabling scroll-based navigation over

click-and-drag interaction. We are excited to gather feedback in our upcoming user study.

We have also released a public beta of *Phoenix*, free for users to try, at <https://assistivity.org>.

ACKNOWLEDGMENTS

This project did not receive any external funding, and was entirely done out of the passions of our coauthors. We wanted to especially acknowledge Kenneth Ge’s contributions with self-funding and leading the project, Ryan Paul’s contributions with designing interfaces and leading workshops, and Priscilla Zhang’s contributions with providing creative ideas that helped ground our work.

REFERENCES

- [1] [n. d.]. *Understanding Dysgraphia - International Dyslexia Association*. <https://dyslexiaida.org/understanding-dysgraphia/>
- [2] Dor Abrahamson and Raul Sanchez Garcia. 2016. Learning Is Moving in New Ways: The Ecological Dynamics of Mathematics Education. *Journal of the Learning Sciences* 25 (01 2016). <https://doi.org/10.1080/10508406.2016.1143370>
- [3] Salim N. Batlouni, Hala S. Karaki, Fadi A. Zaraket, and Fadi N. Karameh. 2011. Mathifier – Speech recognition of math equations. In *2011 18th IEEE International Conference on Electronics, Circuits, and Systems*. 301–304. <https://doi.org/10.1109/ICECS.2011.6122273>
- [4] M. Ben-Jacob. 2019. THE IMPORTANCE OF MATHEMATICS IN STEM EDUCATION. In *EDULEARN19 Proceedings* (Palma, Spain) (*11th International Conference on Education and New Learning Technologies*). IATED, 2496–2500. <https://doi.org/10.21125/edulearn.2019.0683>
- [5] Cynthia L. Bennett, Erin Brady, and Stacy M. Branham. 2018. Interdependence as a Frame for Assistive Technology Research and Design. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, Galway Ireland, 161–173. <https://doi.org/10.1145/3234695.3236348>
- [6] Elliott Cameron and Jeff Bilmes. 2007. Computer Based Mathematics Using Continuous Speech Recognition. In *Workshop on nonverbal acoustic interaction*. <<https://melodi.ee.washington.edu/vj/files/CHIWS-07-Elliott.pdf>>[Consulta: 12-04-2017].
- [7] Lianchun Dong and Yueyuan Kang. 2022. Cultural differences in mindset beliefs regarding mathematics learning. 46 (2022), 101159. <https://doi.org/10.1016/j.cobeha.2022.101159>
- [8] Peter W. Dowrick, John Anderson, Katharina Heyer, and Joie Acosta. 2005. Postsecondary education across the USA: Experiences of adults with disabilities. *Journal of Vocational Rehabilitation* 22, 1 (2005), 41–47. <https://doi.org/10.3233/JVR-2005-00272> arXiv:<https://doi.org/10.3233/JVR-2005-00272>
- [9] Pedro Flores, Eduarda Coelho, Maria Isabel Mourão-Carvalhal, and Pedro Forte. 2023. Association between motor and math skills in preschool children with typical development: Systematic review. *Front Psychol* 14 (Feb. 2023), 1105391.
- [10] Elizabeth N Forde, Latanya Robinson, Joshua A Ellis, and Emily A Dare. 2023. Investigating the presence of mathematics and the levels of cognitively demanding

- mathematical tasks in integrated STEM units. *Disciplinary and Interdisciplinary Science Education Research* 5, 1 (2023), 3.
- [11] Kenneth Gu and JooYoung Seo. 2024. StereoMath: An Accessible and Musical Equation Editor. In *The 26th International ACM SIGACCESS Conference on Computers and Accessibility* (2024-10-27). 1–5. <https://doi.org/10.1145/3663548.3688487> arXiv:2501.01404 [cs]
- [12] D C Geary. 1993. Mathematical disabilities: cognitive, neuropsychological, and genetic components. *Psychol Bull* 114, 2 (Sept. 1993), 345–362.
- [13] Camilla Gilmore. 2023. Understanding the complexities of mathematical cognition: A multi-level framework. *Quarterly Journal of Experimental Psychology* 76, 9 (2023), 1953–1972. <https://doi.org/10.1177/17470218231175325> _eprint: <https://doi.org/10.1177/17470218231175325>
- [14] Eihab Khasawneh, Cameron Gosling, and Brett Williams. 2021. What impact does maths anxiety have on university students? 9, 1 (2021), 37. <https://doi.org/10.1186/s40359-021-00537>
- [15] Markus Knauff, Thomas Mulaik, Jan Kassubek, Helmut R Salih, and Mark W Greenlee. 2002. Spatial imagery in deductive reasoning: a functional MRI study. 13, 2 (2002), 203–212. [https://doi.org/10.1016/S0926-6410\(01\)00116-1](https://doi.org/10.1016/S0926-6410(01)00116-1)
- [16] Kenneth R. Koedinger and Mitchell J. Nathan. 2004. The Real Story Behind Story Problems: Effects of Representations on Quantitative Reasoning. 13, 2 (2004). https://doi.org/10.1207/s15327809jls1302_1
- [17] Mette Lindhardt Als Kristensen, Dorte Moeskær Larsen, Lars Seidelin, and Connie Svabo. 2024. The Role of Mathematics in STEM Activities: Syntheses and a Framework from a Literature Review. *International Journal of Education in Mathematics, Science and Technology* 12, 2 (2024), 418–431. <https://doi.org/10.46328/ijemst.3357>
- [18] Kerry Lee and Swee Fong Ng and. 2011. Neuroscience and the Teaching of Mathematics. *Educational Philosophy and Theory* 43, 1 (2011), 81–86. <https://doi.org/10.1111/j.1469-5812.2010.00711.x> arXiv:<https://doi.org/10.1111/j.1469-5812.2010.00711.x>
- [19] Franklin Mingzhe Li, Michael Xieyang Liu, Yang Zhang, and Patrick Carrington. 2022. Freedom to Choose: Understanding Input Modality Preferences of People with Upper-body Motor Impairments for Activities of Daily Living. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility* (Athens, Greece) (ASSETS '22). Association for Computing Machinery, New York, NY, USA, Article 39, 16 pages. <https://doi.org/10.1145/3517428.3544814>
- [20] Mathful. 2025. Mathful: AI-Powered Math Solver and Learning Platform. <https://mathful.com/> Accessed: 2025-06-01.
- [21] MathGPT Pro. 2025. MathGPT Pro: AI-Powered Math Tutoring Platform. <https://info.mathgptpro.com/> Accessed: 2025-06-01.
- [22] Maria Mokobane, Basil J. Pillay, and Anneke Meyer. 2019. Fine motor deficits and attention deficit hyperactivity disorder in primary school children. 25 (2019), 1232. <https://doi.org/10.4102/sajpsychiatry.v25i0.1232>
- [23] National Council of Teachers of Mathematics. 2003. Mathematics as a Foundation for a Productive Science and Engineering Workforce. In *Pan-Organizational Summit on the US Science and Engineering Workforce: Meeting Summary*, Marye Anne Fox (Ed.). National Academies Press (US), Washington, DC. <https://www.ncbi.nlm.nih.gov/books/NBK36367/>
- [24] Rachel O'Neill, Ruchi Palan, et al. 2023. A Trial of the Assistive Technology EquatIO in an Irish University. *MSOR Connections* 21, 3 (2023), 48–56.
- [25] Nicola J Pitchford, Chiara Papini, Laura A Outhwaite, and Anthea Gulliford. 2016. Fine Motor Skills Predict Maths Ability Better than They Predict Reading Ability in the Early Primary School Years. *Front Psychol* 7 (May 2016), 783.
- [26] V. A. Ressa, S. S. Lazarus, A. R. Hinkle, and C. M. Rogers. 2021. *Scribe: Research (NCEO Accommodation Toolkit #14a)*. Technical Report. National Center on Educational Outcomes. <https://ncoeo.info/Resources/publications/online-accommodation-toolkit/Scribe-research> NCEO is supported through Cooperative Agreement #H326G160001 with the Office of Special Education Programs, U.S. Department of Education.
- [27] Dmitri Rozgonjuk, Tiina Kraav, Kristel Mikkor, Kerli Orav-Puurand, and Karin Täht. 2020. Mathematics anxiety among STEM and social sciences students: the roles of mathematics self-efficacy, and deep and surface approach to learning. 7, 1 (2020), 46. <https://doi.org/10.1186/s40594-020-00246-z>
- [28] Robert Stevens and Alistair Edwards. 1994. MathTalk: The design of an interface for reading algebra using speech. In *Computers for Handicapped Persons*, Wolfgang L. Zagler, Geoffrey Busby, and Roland R. Wagner (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 313–320.
- [29] StudyFetch LLC. 2025. StudyFetch: AI-Powered Learning Platform. <https://www.studyfetch.com/> Accessed: 2025-06-01.
- [30] PAULO TAN. 2017. Building Inclusive Mathematics Classrooms for Students with Disabilities. *For the Learning of Mathematics* 37, 3 (2017), 21–24. <https://www.jstor.org/stable/26548465>
- [31] Texthelp Ltd. 2025. Equatio – Equation Editor to Create Accessible Digital Math. <https://www.texthelp.com/products/equatio/> Accessed: 2025-06-01.
- [32] Isabel Vale and Ana Barbosa. 2017. The importance of seeing in mathematics communication. (2017). <http://repositorio.ipv.pt/handle/20.500.11960/3762> Accepted: 2023-12-18T13:44:43Z Publisher: ETEN.
- [33] Delinda Van Garderen and Marjorie Montague. 2003. Visual–Spatial Representation, Mathematical Problem Solving, and Students of Varying Abilities. 18, 4 (2003), 246–254. <https://doi.org/10.1111/1540-5826.00079> Publisher: SAGE Publications.
- [34] P. C. Wason and Diana Shapiro. 1971. Natural and contrived experience in a reasoning problem. 23, 1 (1971), 63–71. <https://doi.org/10.1080/00335557143000068> Publisher: SAGE Publications.
- [35] Wahyu Widada, Dewi Herawaty, Mohammad H. Rahman, Dita Yustika, Elsa P. Gusvarini, and Abdurrobbil Falaq Dwi Anggoro. 2020. Overcoming the difficulty of understanding systems of linear equations through learning ethnomathematics. 1470, 1 (2020), 012074. <https://doi.org/10.1088/1742-6596/1470/1/012074> Publisher: IOP Publishing.
- [36] Angela M. Wigmore, Eckhard Pflugel, Gordon J.A. Hunter, James Denholm-Price, and Martin Colbert. 2010. TalkMaths Better! Evaluating and Improving an Intelligent Interface for Creating and Editing Mathematical Text. In *2010 Sixth International Conference on Intelligent Environments*. 307–310. <https://doi.org/10.1109/IE.2010.62>
- [37] Beth Witham and Gayle Brewer. 2024. “Giving the People Who Use the Service a Voice”: Student Experiences of University Disability Services. *Disabilities* 4, 1 (2024), 1–10. <https://doi.org/10.3390/disabilities4010001>

A DETAILED TEST CONDITIONS AND RESULTS

A.1 Test Environment

- **Device:** Windows laptop
- **Browser:** Google Chrome
- **EquatIO Version:** Free version (Accessed June 12, 2025)
- **Access Method:** Via redirect link <https://equatio.texthelp.com> (sourced through ChatGPT)
- **Installation Process:**
 - Installed Chrome extension and desktop application.
 - Attempted sign-in via desktop app, which redirected to a browser login page that repeatedly failed to load.
 - Signing in with Google also led to non-loading page.
 - Ultimately accessed EquatIO through the redirect link above; this link was not easily discoverable through standard web search.

A.2 Testing Process - EquatIO

Equations were dictated using EquatIO's speech input feature. Below are the prompts used and their corresponding outputs.

Prompt 1: "The integral from zero to infinity of x squared, dx".

- **Expected Output:**

$$\int_0^{\infty} x^2 dx$$

- **Actual Output:**

- **Notes:** Integral notation and limits were not recognized; the output was a partial, incorrect transcription.

Prompt 2: "The integral from zero to pi over two, cosine of x, dx".

- **Expected Output:**

$$\int_0^{\frac{\pi}{2}} \cos(x) dx$$

- **Actual Output:**

$\int_0^{\frac{\pi}{2}} \cos(x) dx$

- **Notes:** Similar issues with parsing. Fractional upper limit and function notation were partially recognized but inaccurately rendered.

Prompt 3: "Index of refraction one sine of theta one equals index of refraction two sine of theta two".

- **Expected Output:**

$$n_1 \sin(\theta_1) = n_2 \sin(\theta_2)$$

- **Actual Output:**

$n_1 \sin \theta_1 = n_2 \sin \theta_2$

- **Notes:** EquatIO did not associate “index of refraction” with the variable n .

A.3 Testing Process - *Phoenix*

The same prompts were dictated using *Phoenix*'s speech input feature in the identical test environment. The outputs below were generated using the latest prototype version of *Phoenix* as of June 2025.

Prompt 1: "The integral from zero to infinity of x squared, dx".

- **Expected Output:**

$$\int_0^{\infty} x^2 dx$$

- **Actual Output:**

$\int_0^{\infty} x^2 dx$

- **Notes:** *Phoenix* successfully recognized and formatted the integral expression, including both bounds and the integrand.

Prompt 2: "The integral from zero to pi over two, cosine of x, dx".

- **Expected Output:**

$$\int_0^{\frac{\pi}{2}} \cos(x) dx$$

- **Actual Output:**

$\int_0^{\frac{\pi}{2}} \cos(x) dx$

- **Notes:** The system accurately parsed both the limits and the function. “Pi over two” was correctly rendered as a LaTeX fraction.

Prompt 3: "Index of refraction one sine of theta one equals index of refraction two sine of theta two".

- **Expected Output:**

$$n_1 \sin(\theta_1) = n_2 \sin(\theta_2)$$

- **Actual Output:**

$n_1 \sin \theta_1 = n_2 \sin \theta_2$

- **Notes:** Unlike EquatIO, *Phoenix* correctly mapped “index of refraction” to n and handled symbolic input well.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009