Intelligent Tutoring System (ITS) for Mathematics using Computational Engines

by Jason Neville Stanley

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy in Mathematics

under the supervision of Prof Christopher Poulton

University of Technology Sydney
Faculty of Science

June 2025

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Jason Stanley declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Mathematical and Physical Sciences/Faculty of Science at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program (RTP) Scholarship doi.org/10.82133/C42F-K220.

Signature: Production Note: Signature removed prior to publication.

Date: 3rd November 2025

I would like to dedicate this thesis to my long suffering wife Jenny and our sons and daughters,

in particular our daughter Renee,

and to my long lost siblings; ours is a cross to bear, no one child should …

Acknowledgements

And I would like to acknowledge ...

Tim Langtry	I would like to give heart felt thanks to Tim without whom this work could not have even begun...
Chris Poulton	Thank you also to Chris who has breached the largest of gaps when needed.
Scott Alexander	Thank you for your horsepower in the field of mathematics, statistics and craft in the art of dissertation and your generosity
Yakov Zinder	It has been a great privilege to work along side a master of the scientific tradition
and	
Mary Coupland	My inspiration to take on the challenges of an academic life...

Statement of Format

This Thesis titled:

Intelligent Tutoring System (ITS) for Mathematics Using Computational Engines
is submitted as a Conventional Thesis.

Table of contents

List of figures

List of tables

Definition of Terms

AI Artificial Intelligence

ANOVA **An**alysis **o**f **Va**riance

Use ANOVA when:

You have one or more categorical independent variables (factors).

You have a continuous dependent variable.

You want to test whether differences in group means are statistically significant.

Types of ANOVA:

One-Way ANOVA: Compares the means of multiple groups based on a single factor. Example: Comparing test scores across different teaching methods.

Two-Way ANOVA: Compares group means based on two factors, and can test for interaction effects. Example: Comparing test scores by teaching method and gender.

Repeated Measures ANOVA: Used when the same subjects are measured under different conditions or over time. Example: Pre-test, mid-test, post-test performance.

Basic Hypotheses in One-Way ANOVA:

Null hypothesis H_0: All group means are equal.

Alternative hypothesis H_a: At least one group mean is different.

Key Output:

F-statistic: Ratio of between-group variance to within-group variance.

p-value: If $p < \alpha$ (commonly 0.05), reject the null hypothesis.

Example (One-Way ANOVA in Plain Terms): Suppose you test three different tutoring methods (A, B, and C) on students' maths scores. ANOVA helps you determine if the average scores differ significantly among the three methods.

API Application Programming Interface — A defined set of rules and protocols that allows different software components or systems to communicate and exchange data.

APOS Action → Process → Object → Schema

CAI Computer-Assisted Instruction

CAS Computer Algebra System

CB Constraint-Based Intelligent Tutoring System — uses domain-specific constraints to evaluate student responses.

CI Confidence Interval

DBR Design-Based Research

EO Epistemological Obstacle is a conceptual difficulty in learning arising from prior knowledge that is incompatible with new scientific or mathematical concepts.

Extensible **Extensibility** - A key design priority for the user interface is *extensibility*—the capacity of the system to accommodate future enhancements, user customisations, and evolving pedagogical requirements without necessitating significant architectural changes. This principle is realised through a modular design approach, wherein interface components are constructed as discrete, reusable units that can be dynamically arranged or substituted. By maintaining a clear separation between presentation, logic, and data layers, the interface permits flexible adaptation to diverse use cases, including personalised learner paths and discipline-specific content extensions. Moreover, the integration of plugin architectures and API endpoints supports external contributions to interface functionality, allowing the platform to scale organically in response to user needs and technological developments. In the context of an ITS, such extensibility ensures that the interface can evolve in tandem with improvements to Student Modelling, feedback mechanisms, and domain-specific instructional strategies. Historically, UI and UX are important design criteria for extensible web sites.

H5P HTML5 Package — H5P content can be embedded in platforms like `Moodle`®, `WordPress`®, `Drupal`®, and `Canvas`®, and is widely used in LMS platforms.

HTML HyperText Markup Language — an interpreter-based coding language with internet browsers locally serving user web pages.

HTTPS HyperText Transfer Protocol Secure. It is the secure version of HTTP, the protocol over which data is sent between a web browser and a website.
HTTPS uses TLS (Transport Layer Security) to encrypt communication. It ensures:

- Confidentiality – data is encrypted so third parties can't read it.
- Integrity – data is not tampered with during transit.
- Authentication – confirms the identity of the server via digital certificates.

All communication between the application layer and client devices is conducted over HTTPS, ensuring secure data transfer and protecting user input. The use of HTTPS supports encrypted API calls between the UI and backend services, including those used for symbolic computation, student model updates, and dynamic content rendering.

ITS Intelligent Tutoring System — computer-based systems with adaptive feedback to learners.

JSP JavaServer Pages, script files of suffix '.jsp'

LLM Large Language Model — a machine learning model trained on massive datasets to generate parameters describing next step solutions.

LMS Learning Management System

LTI Learning Tools Interoperability. A standard developed by IMS Global for integrating third-party tools with Learning Management Systems (LMSs).

MCQ Multiple Choice Questions

MPM The Multi-Processing Modules (MPM) library facilitates communication between Apache HTTP Server and backend servlet containers such as Apache Tomcat. This interaction is handled via transport implementations known internally within Apache as Connectors. One common example is the AJP 1.3 Connector (Apache JServ Protocol), which enables efficient binary communication between Apache and Tomcat. Connectors like AJP rely on workers, which are process or thread entities responsible for managing the actual data exchange. In this context, a worker refers to a configuration unit that defines how requests are forwarded to Tomcat, how connections are managed, and how load balancing or failover might be handled. Workers can be individually configured for specific roles—such as handling requests for particular contexts, balancing load across multiple Tomcat instances, or providing redundancy in case of failure. They form the operational backbone of the connector architecture, enabling scalable, high-performance integration between web and application tiers.

MRS The Mathematics Readiness Survey - is a diagnostic tool administered, by UTS, to first-semester Science and Engineering undergraduates, designed to assess mathematical under-preparedness at the point of transition to tertiary study. First implemented in 2013, the survey has run continuously since its inception and, at the time of publication, has assessed over 15,000 students.

MSP The Mathematica Script Processor (MSP) is a backend service responsible for executing Wolfram Language scripts (commonly referred to as M-scripts) within the system architecture. It functions as a middleware layer between the application server (e.g., Apache Tomcat) and the Mathematica computational engine.

The MSP supports:

Symbolic evaluation of student input, including algebraic simplification, equation solving, and expression comparison, Pattern-based equivalence checking, allowing for flexible assessment of mathematically correct but structurally different answers, Custom feedback generation, based on the nature of the transformation or mistake detected, Scalable API endpoints, callable by the interface layer to return results in real time.

The processor communicates via secure shell or HTTP API using WolframScript, enabling the remote execution of Mathematica logic scripts stored as .wl or .m files. By abstracting this layer into a dedicated MSP, the system gains modularity, language separation, and maintainability.

MT Model-Tracing Intelligent Tutoring System — compares student problem-solving steps against ideal solution paths.

MVP Minimum Viable Product - common in startups/tech:

The simplest version of a product that can be released to test a concept and gain user feedback.

Example: A basic app with just one core feature to see if users find it useful.

MySQL — An open-source relational database management system (RDBMS) used for storing, managing, and querying structured data using SQL (Structured Query Language).

Overlays An assistive approach in modelling assuming learner knowledge is a subset of expert knowledge.

Perturbation An intentional deviation from expected behaviour used to diagnose misconceptions in Student Modelling.

PHP (Hypertext Preprocessor) — A server-side scripting language designed for web development, used to generate dynamic page content and interact with databases.

PWYM Pathway to University Mathematics - An Australian Federal Government nationally funded grant, to encourage school age students to increase participation in tertiary studies in the mathematical sciences.

Scaffolding A support structure provided by a teacher or system to help a learner perform a task until they can perform it independently.

SR Symbolic Reasoning-Manipulating symbols according to formal rules as used in `Mathematica`.

STEMM Science, Technology, Engineering, Mathematics, and Medicine.

TTY **T**ele**ty**pewriter, but in modern computing it refers to terminal interfaces.
Historically: TTYs were physical electromechanical typewriter devices used to send and receive typed messages over long distances.
Today: In Unix/Linux, a TTY refers to a terminal session — either physical (like /dev/tty1) or virtual (like terminal emulators).
Administrative access to the server hosting the ITS is available via secure shell (SSH) connections to a virtual TTY interface, enabling direct interaction with system processes for maintenance or logging purposes.

UI User Interface **Extensibility (UI)** — The ability of a user interface to accommodate new components, layouts, or features without altering its core structure.

UTS University of Technology Sydney

UX User Experience **Extensibility (UX)** — The capacity of a user experience design to evolve over time, supporting new user needs, workflows, or content without compromising usability or coherence.

XML (eXtensible Markup Language) — A flexible text format for structuring and exchanging data between systems, designed to be both human-readable and machine-parsable.

ZPD Zone of Proximal Development — Vygotsky's term for the range of tasks a learner can perform with guidance but not yet independently.

Abstract

This thesis addresses a persistent challenge in mathematics education: the epistemological and pedagogical disconnect between secondary and tertiary learning. While secondary instruction often emphasises procedural fluency and examination performance, tertiary mathematics demands abstraction, symbolic reasoning, and conceptual understanding. This misalignment contributes to student underpreparedness, disengagement, and attrition in mathematically intensive disciplines.

To bridge this gap, the research explores the potential of Intelligent Tutoring Systems (ITSs) grounded in constructivist learning theories and powered by symbolic computation. The study positions curriculum intuition—a student's capacity to recognise structure, pattern, and mathematical form—as central to successful transition. It conceptualises learning as a dynamic, perturbation-sensitive process, where moments of cognitive disequilibrium indicate readiness for conceptual growth.

The thesis develops and evaluates a prototype Intelligent Tutoring System (ITS) that incorporates a symbolic reasoning engine (`Mathematica`®) to interpret student input, recognise mathematical equivalence, and provide adaptive feedback. This system is implemented using modular, LMS-compatible web technologies (notably Extensibility), supporting scalable deployment while retaining responsiveness to individual learners.

A design-based research (DBR) methodology underpins the project, integrating iterative development with theoretical inquiry. Findings from classroom trials demonstrate the system's capacity to scaffold conceptual understanding, detect meaningful perturbations, and support learners in transitioning from procedural to relational mathematical thinking.

The research contributes a novel hybrid Student Model sensitive to conceptual dynamics, proposes diagnostic uses of perturbation for feedback, and advances the integration of symbolic engines within ITS design. Future work is outlined, including the prospective augmentation of ITSs with large language models to combine conversational flexibility with formal mathematical rigour. The study affirms the importance of theoretically informed, technologically enabled interventions in addressing systemic challenges in mathematics transition pedagogy.

Chapter 1

Introduction

1.1 Developing Curriculum Intuition: An Epistemological and Pedagogical Framing

Helping students build a feel for mathematical structure, pattern, and form — what we might call *curriculum intuition* — is not just a teaching technique; it also reflects a belief about how knowledge in mathematics is understood and developed. In practice, when working with students, teachers often use deliberate instructional language — including prompts, questions, and explanatory cues — to foreground mathematical thinking, noticing, and intuitive reasoning [154, 114], and to guide attention and promote disciplinary engagement [73, 7, 119]. Epistemologically[1], this approach aligns with a constructivist view of knowledge, wherein mathematical understanding is actively constructed rather than passively received [189]. It challenges the procedural conception of mathematics, inviting learners to build intuition and internalise meaning through exploration and reflection [164, 79].

Such an approach flags support for relational understanding, emphasising connections between concepts rather than isolated techniques. It echoes work on mathematical noticing [114], and links with embodied and intuitive cognition, such as Tall's concept of *conceptual embodiment* [178]. It also reflects key aspects of Action, Process, Object and Schema (APOS)[2] theory, as learners interiorise actions, form processes, and build schema through abstraction and generalisation [18].

Pedagogically, the approach fosters a disposition of curiosity and critical sense-making. By valuing intuition, visualisation, and generalisation, it positions students as active partic-

[1]In education research, *epistemology* refers to how knowledge is understood within a field of learning. It is a widely used and foundational concept. See [92, 6, 143, 76].

[2]For convenience a list of definitions and acronyms is provided in Definition of Terms.

ipants in the discipline, rather than passive recipients. It invites learners to see mathematics as a connected, expressive, and human activity—aligning with contemporary calls for deep mathematical thinking and the development of flexible expertise.

Yet this pedagogical vision often meets resistance at a structural level, particularly in the transition from secondary to tertiary mathematics. Students frequently arrive at university with an understanding of mathematics shaped by procedural practice, high-stakes assessment, and an emphasis on correct answers over conceptual depth [183]. They may be proficient in applying algorithms but lack experience with formal abstraction, proof, and symbolic generalisation—features that define university-level mathematics [91]. These difficulties are often compounded by affective factors such as mathematics anxiety, diminished self-efficacy, and a perception of mathematics as elitist or inaccessible.

In response, various reforms have been implemented over the past decades: curriculum redesigns aimed at better alignment between school and university syllabi, foundation or bridging courses to remediate knowledge gaps, diagnostic testing to identify and target underpreparedness, and the incorporation of digital learning tools to personalise instruction. While each initiative has had value, many have struggled to address the deeper epistemological and pedagogical disjunct between secondary and tertiary mathematics. As Artigue [15] notes, school mathematics is typically presented as a closed, rule-bound system oriented around performance, whereas university mathematics demands flexible reasoning, a tolerance for ambiguity, and engagement with the disciplinary norms of argument and justification.

As a result, the challenge of transition remains unresolved. Underpreparedness continues to manifest in disengagement, low achievement, and high attrition—particularly among students entering mathematically intensive disciplines. Given the centrality of mathematics to science, technology, engineering, medicine, and more broadly to civic and professional reasoning, addressing this gap is not only a matter of academic concern but a broader imperative for educational equity and national capacity in STEMM fields.

At the core of this problem is a mismatch between students' prior learning experiences—often focused on procedures, routines, and examination-driven content—and the expectations of tertiary-level mathematics, which demand a deeper form of understanding. This includes the capacity to reason about abstract structures, manipulate symbolic expressions fluently, and engage reflectively with mathematical representations. This transition is not merely curricular; it represents a deeper epistemological shift in how mathematical knowledge is framed, moving from procedural fluency to conceptual understanding — a distinction central to constructivist pedagogies [39].

This thesis explores how an Intelligent Tutoring System (ITS), underpinned by symbolic computation and learning theory, can offer targeted support during this critical transition. The focus is not simply on error correction or procedural reinforcement, but on scaffolding conceptual development and detecting the nuanced moments in student reasoning that indicate learning potential. The project aims to move beyond behaviourist framings of learning as input-output performance, instead positioning Intelligent Tutoring Systems (ITSs) as interactive environments where knowledge construction is dynamically supported.

To frame this investigation, it is important to first consider what is meant by *theory* in an educational context.

In the physical sciences, theories carry a more rigorous connotation: they are well-substantiated explanations of phenomena, built on empirical validation and predictive power. Yet educational research operates differently. Here, theories are frameworks that help describe and explain complex, situated processes of learning. Their value lies not in their predictive generalisability, but in their interpretive power and their capacity to inform practice.

Educational theories often differ in their epistemological stance. Hofer and Pintrich [92] explicitly classify and discuss differing beliefs about the nature of knowledge — whether it is seen as simple or complex, certain or uncertain, transmitted, constructed, discovered, or negotiated. Rather than offering universal laws, educational theories provide structured ways of thinking about how knowledge is acquired, organised, and transformed. These theories guide observation, interpretation, and intervention in educational practice. In this sense, they are not hypotheses to be confirmed or falsified, but lenses through which educational phenomena are made sense of.

Historically, the development of psychology shaped early approaches to learning theory (for a timeline of progress see Driscoll [70]), particularly through the rise of behaviourism. Pavlov's classical conditioning and Watson's advocacy for an objective, observable psychology [191] marked a shift toward empirical study of behaviour as a proxy for mental processes. Educational psychology emerged from this foundation, contextualising experimental insights within school and instructional settings: the focus moved from pure stimulus-response to questions about motivation, understanding, and the conditions under which learning is most effective [27, 8].

Over time, educational research began to foreground the complexity and context-dependence of learning. This gave rise to pedagogical models that integrate psychological insight with curricular, social, and technological dimensions of teaching. Pedagogy—far from being a neutral delivery mechanism—became understood as the deliberate orchestration of environments, tasks, and interactions that foster learning. The International Commission on Mathematical Instruction (ICMI) gives a clear definition of pedagogy as:

> "...By pedagogy we mean the teachers' orchestration of teaching and learning environments and situations, examined both from the descriptive/analytic position (what is the case?) and the normative position (what ought to be the case?)..." [127]

Such a view recognises that educational practices are not only technically driven but ethically and philosophically situated. Theories of learning therefore do not simply inform instruction; they shape the very aims and assumptions that underlie educational design.

This characterisation of educational theory as non-predictive and context-dependent is not without critique. Among the most prominent challenges to this view comes from John Dewey, who reimagined the scientific method not as a fixed procedural canon exclusive to the physical sciences, but as a generalisable mode of inquiry. In Dewey's account [67], scientific thinking is characterised by reflective problem-solving, sustained observation, and iterative refinement—features that are no less applicable in education than in laboratory settings.

For Dewey, the significance of educational theory was not its predictive accuracy but its potential to guide meaningful change. He viewed education as a fundamentally democratic process—one where inquiry and experience play central roles in fostering critical thought and collective growth. Within this framework, the scientific method becomes a form of disciplined, pragmatic thinking: a tool for teachers and learners to investigate problems, generate hypotheses, and refine practices in response to evolving circumstances.

This vision legitimises educational research as an empirical activity, even in the absence of generalisable laws. Rather than undermining its validity, the contextual nature of educational inquiry is what enables it to remain responsive to the complexity and variability of learning environments. Dewey's legacy thus provides a philosophical foundation for approaches like design-based research, which embrace iteration, contextual sensitivity, and theory-practice integration.

It is within this philosophical and methodological framework that the present research proceeds. The thesis builds on constructivist learning theories—particularly APOS theory and Piagetian abstraction—to inform the design of a hybrid Student Model. Rather than treating student knowledge as a fixed set of correct or incorrect items, the model is sensitive to *perturbations*—the moments when a learner's current understanding is insufficient to resolve a new problem, triggering cognitive restructuring. These moments are diagnostically rich: they signal the boundaries of current knowledge and the potential for conceptual growth.

Current ITSs often struggle to model such dynamics. Overlay models, that track those concepts a student has mastered by comparing their responses to an expert model, treat student knowledge as a subset of expert knowledge [43, 41]. This limits their ability to recognise productive misconceptions or alternative conceptions. Bayesian networks, while more flexible [53], still assume a static ontology of domain knowledge and often lack symbolic interpretive capacity [109]. These limitations are particularly acute in tertiary mathematics, where multiple representations, symbolic transformations, and formal equivalences are central [161, 101]. To overcome this, the system proposed in this thesis employs a symbolic reasoning engine here implemented in `Mathematica`®, to interpret student input, recognise valid mathematical equivalences, and support open-form expressions. This enables the ITS to respond not only to correctness but to the structure and logic of student reasoning. Combined with web-based Hyper Text Markup Language (HTML) and subsequent advancements such as HTML5 Package (H5P) for modular interaction and LMS integration, the system enables scalable deployment while retaining adaptability and responsiveness.

The goal of the approach contained in this thesis is not simply to emulate human tutoring, but to create an intelligent learning environment grounded in both cognitive theory and symbolic verification. In doing so, the project explores how technology can meaningfully bridge the epistemological gap between secondary proceduralism and tertiary abstraction—a gap that remains one of the most persistent obstacles in mathematics education.

1.2 Research Questions

The overall research questions posed in this thesis are:

1. How can symbolic reasoning engines enhance Student Models in ITSs for tertiary mathematics?

2. How can perturbation be used diagnostically to inform adaptive feedback in conceptual learning?

3. In what ways can design-based research support the iterative development and evaluation of ITSs aligned with constructivist pedagogy?

The three research questions outlined above frame the inquiry at the core of this thesis. Each question targets a distinct but interrelated dimension of the project: the potential of symbolic engines to enrich Student Modelling; the diagnostic use of perturbation to enable adaptive, conceptual feedback; and the affordances of design-based research for developing

ITSs aligned with constructivist pedagogies. To address these questions, the thesis proceeds through a structured exploration — beginning with a critical review of prior work, then articulating a theoretical framework and methodology, followed by system design and empirical evaluation. Across this progression, key findings emerge: that symbolic reasoning engines can serve as both validators and explainers of student logic; that perturbation, when carefully structured, reveals latent conceptual misconceptions; and that design-based research enables iterative alignment between pedagogical intent and technical implementation. The following chapters are organised to reflect this trajectory, suggesting how each research question is engaged through theory, design, and evaluation.

1.3 Thesis Overview

The thesis is structured as follows:

Chapter 2 – Literature Review

This chapter surveys the landscape of prior research relevant to this thesis. It begins by examining the persistent challenges in mathematics transition pedagogy, identifying key barriers to conceptual readiness and symbolic fluency. It then reviews the architecture of intelligent tutoring systems, including constraint-based and model-tracing approaches, and assesses their applicability to tertiary-level mathematics. Attention is given to Student Modelling techniques —– overlay models, Bayesian networks, and perturbation-based diagnostics —– as well as the affordances and limitations of symbolic reasoning in educational technologies. The chapter concludes by identifying critical gaps in current ITS design, particularly in the areas of conceptual feedback, symbolic validation and equivalence of different mathematical expressions, and integration with modern learning management systems.

Chapter 3 – Theoretical Framework

Here, the research is based in constructivist theories of learning, particularly APOS theory and Piagetian models of abstraction and equilibration. These theoretical commitments inform how student knowledge is modelled and how perturbations in reasoning are interpreted. Further, the epistemological foundations of learning theories are explored, distinguishing them from predictive scientific models and positioning them instead as interpretive frameworks. Also traced is the influence of behaviourism and educational psychology on the evolution of pedagogy, setting the stage for an ITS design that is both theoretically robust and contextually sensitive.

Chapter 4 – Methodology (Research Design)
The research follows a design-based research (DBR) methodology, enabling iterative development, testing, and refinement of the tutoring system in authentic educational settings. This chapter details each phase of the DBR cycle—analysis of needs, design of prototypes, implementation in context, and evaluation of learning outcomes. Justification is provided for the choice of evaluation methodologies, with an emphasis on how these support both theoretical inquiry and practical design. The DBR approach aligns closely with Deweyan pragmatism, treating educational research as inquiry into complex, evolving systems.

Chapter 5 – System Design and Implementation
This chapter presents the technical and pedagogical architecture of the prototype ITS. It describes how Mathematica is used as a symbolic engine to interpret and validate student input, supporting open-ended responses and equivalence checking, that is, checking of the equivalence of different mathematical expressions. The Student Model is elaborated, including mechanisms for tracking knowledge states, identifying perturbations, and delivering adaptive feedback. The system's interface is implemented using extensible HTML, enabling modular, LMS-compatible deployment. Design decisions are explicitly linked to the theoretical framework and to the affordances of symbolic computation in mathematics education.

Chapter 6 - Authoring
This chapter presents the authoring environment developed to support the creation of intelligent, curriculum-aligned learning experiences within Tutoria. Emphasis is placed on how the interface enables pedagogical adaptability, allowing educators to design tasks that incorporate feedback logic, assessment criteria, and multiple modes of interaction. The authoring system reflects the underlying models discussed in earlier chapters, including the Domain Model, Student Model, and Teaching (tutor) Model, providing a structured yet flexible framework for content creation. This chapter supports the conceptual architecture already developed, its integration with the symbolic reasoning engine, and the influencing of Domain and Teaching Models directly.

Chapter 7 – Evaluation and Results
In this chapter, the system is evaluated through a pilot trial. Quantitative data were collected and analysed as part of the pilot trial. Initial analysis of the small sample data prompted bootstrap analysis. Some tentative conclusions were drawn from these results.

Chapter 8 – Discussion and Future Work

This final chapter synthesises the implications of the research for both theory and practice. It reflects on the effectiveness of the hybrid ITS model in achieving the aims of the project. Limitations are acknowledged, including the challenges of generalisability and system scalability. The chapter then looks ahead to future developments in intelligent tutoring, including the integration of Large Language Models (LLMs). While LLMs excel in generating human-like responses, they currently lack the capacity for deterministic symbolic verification—a key requirement in mathematics education. However, future work may explore how symbolic engines can be paired with LLMs to leverage their dialogic strengths while preserving mathematical precision. This hybrid architecture could enable more conversational, context-aware tutoring without sacrificing formal correctness, offering a promising direction for the next generation of Artificial Intelligence (AI)-enhanced educational tools.

Chapter 2

Literature Review

2.1 Overview

This review examines the background to, and theory of, Intelligent Tutoring Systems (ITSs) as they apply to mathematics in transition stages. In Section 2.2 we consider the general nature of students' experience of the transition from secondary to tertiary mathematics education and in Section 2.3 we consider the pedagogical challenges that this experience highlights, as well as some of the responses that these challenges have prompted, including various technology-based initiatives such as the development of ITSs. Section 2.4 explores in more detail the development of ITSs and their relationship with educational and learning theory. Section 2.5 focusses on the needs driving the development and deployment of ITSs, and the design principles informing their development. In Section 2.6 we examine some case studies of ITSs that have been deployed, largely in the USA. Drawing on these we then consider some specific opportunities for design enhancements that might improve the effectiveness of ITSs in the transition from secondary to tertiary mathematics education. Some concluding remarks are offered in Section 2.7.

2.2 The Transition from Secondary to Tertiary Mathematics Education

Published in 1998, 'Difficulties in the passage from secondary to tertiary education' (De Guzman *et al* [64]) compared various countries' approaches to mathematics teaching at the tertiary level, identifying evolving and changing practice as a persistent difficulty in student development. They concluded that there are widespread sociological, cultural and didactic changes taking place, which are not only local but global as well. The difficulties resulting from change are framed as being epistemological or cognitive. Epistemological difficulties relate to requisite knowledge or pre-knowledge and locate students in unfamiliar contexts for which they are often not prepared. Cognitive difficulties relate to the abstract nature of mathematics. Mathematics explored through proof requires experience in counter-example and deductive reasoning [87], increasing the cognitive load of students significantly compared to secondary schooling contexts.

Various approaches have been adopted in addressing these concerns (see for example Hoyles *et al* [96] and De Guzman *et al* [64]). In particular, Hoyles *et al* [96] highlighted the uniformity in course management across several countries and suggested that students placed less value on rigor and precision in mathematics than in the past, preferring to calculate intuitively. A number of authors [95, 194, 151] have identified this as a problem that requires further exploration, as the number of students entering tertiary study with limited mathematical skills is increasing. Studies have been undertaken to remediate the situation on a number of fronts – for example Nair *et al* [130] investigated the classroom environment of 504 science classes in Australian and Canadian tertiary institutions using a modified and personalized form of the College and University Classroom Environment Inventory (CUCEI). They reported that students at the tertiary level would prefer a more favourable learning environment in all areas measured by the seven scales of the CUCEI. In summary CUCEI is an instrument designed to assess students' perceptions of their classroom learning environments, specifically at the senior secondary and tertiary levels of education. In [130] the standard CUCEI design was modified into a personal form to capture individual students' perceptions of their role within the classroom environment, rather than a general perception of the class as a whole. This modified CUCEI used seven scales to assess different aspects of the learning environment:

- Personalisation
- Student Cohesiveness
- Task Orientation
- Cooperation
- Individualisation
- Equity
- Innovation

Students and instructors completed both actual (their perception of the current environment) and preferred (their ideal environment) versions of the CUCEI using a five-point Likert scale. Statistical analyses, such as Confidence Intervals (CI) and paired t-tests, were used to compare students' perceptions across educational levels and genders. The findings regarding differences between secondary and tertiary education have highly interesting implications for transitions within and from higher education — particularly that higher education students generally had less favourable perceptions of their learning environments. The gender differences in perception are also noteworthy, especially how they converge at higher education levels. Key findings included:

- Students at the higher education level generally had a less favourable perception of their learning environment compared to those at the senior secondary level.
- Female students tended to perceive their learning environments more positively than male students, although these perceptions became more similar as students progressed to higher education.
- The CUCEI was able to differentiate significantly between the perceptions of students in different classrooms.
- Also revealed were differences between students' actual and preferred learning environments, suggesting potential areas for improvement in higher education learning environments. For example, tertiary students perceived less student cohesiveness and less favourable interpersonal relationships with instructors.

Overall, the authors demonstrated the value of the modified and personalised CUCEI as a tool for understanding students' perceptions of their learning environments during the transition from senior secondary to higher education. Identifying the secondary/tertiary transition problem as multi-dimensional, Skovsmose, Valero and Christensen [187] outlined and categorized many of the challenges of this transition. Following are some of the key dimensions that the study identified.

(a) *Changes in processes of teaching and learning*
Identified as key components in the change process are problem-based learning (PBL) and associated changes in assessment practices. Market demand for graduates increases with a PBL profile and represents a change in emphasis, as stated:

> "(the) approach did not put emphasis on schooling into the traditions of a particular discipline." [187, p. 3]

Changes in assessment practices also concentrate assessment focus, leading to a re-negotiation of the didactic contract between lecturer and student.

(b) *Changes in academic cultures*
Priorities and discourse attached to academic communities remain firmly in the hands of staff [187], prompting the question of how to assess the initiative and qualifications of staff to lead educational initiatives beyond a traditional approach concentrating primarily on the transfer of content knowledge.

(c) *Structural and administrative changes*
Universities are coping with change in their own way and Skovsmose *et al*, state that:

> "In the classic university, associated with the Humboldt-ian tradition, the main aim was to produce knowledge, and to do so independently of particular interests be it religious or economic". [187, p. 9]

However, other authorities [111] clearly link university priorities with economic growth, viz:

> "The knowledge economy and society stem from the combination of four interdependent elements: the production of knowledge, mainly through scientific research; its transmission through education and training; its dissemination through the formation and communication technologies; its use in technological innovation. At the same time new configurations of production, transmission and application of knowledge are emerging. [...] Given that they are situated at the crossroads of research, education and innovation, universities in many respects hold the key to the knowledge economy and society." [111, p. 32][1]

Metrics related to post-graduate funding, research and publication figure pervasively in university decision making [187, p. 10].

(d) *Changes in conception of science*
According to Skovsmose *et al* [187, p. 11] perhaps the most salient development in the university sector is the move toward institutions being competitive by nature, based in production, commercial confidences and patenting for profit and metric achievement around competition in addition to educational schemas.

These studies suggest that further work is needed to better structure and coordinate the transitional space between secondary and tertiary mathematics education [48, 147, 75, 20,

[1]The original article was published in Swedish and was translated into English by Google Translate.

95]. Here we refer to the intentional design of curricula, support mechanisms, and learning environments that address the cognitive, emotional, and epistemological challenges students face in their first year of university mathematics. A number of studies (see for example [42, 168]) have investigated the use of technology as an assistive agent in this context, particularly through ITS, learning management platforms, and diagnostic assessment tools. This is discussed in more detail in the following sections.

2.3 A Problem in Mathematics Education

The Secondary to Tertiary transition problem In Section 2.2 we considered elements of research connecting tertiary curricula objectives, tertiary transition pedagogy and students as secondary school actors. Considerations of curricula are particularly relevant in this context.

However, as we shall see in Section 2.5, research into the transition from secondary to tertiary education is not confined to the experiences or actions of those directly involved — such as students or teachers. Instead, it also includes investigations led by researchers, institutions, and policymakers who study transition challenges from systemic, technological, or theoretical perspectives. In this section, we consider in more detail the pedagogical challenges of the transition, as well as some uses of information technology in novel approaches to education, designed to address these challenges at the secondary/tertiary boundary and beyond. These approaches often leverage advances in computer science and, more recently, AI. Indeed, ever since the return of service personnel from World War II, researchers have explored how information technology — and now AI — can support individuals in transitioning into new learning or workforce contexts (for example see Sleeman, D. and Brown, J.S. [165]). Programs originally developed for vocational reintegration have since expanded and diversified, now intersecting with many sectors of education.

2.3.1 Mathematics Education as Addressed by Mathematicians

The demands of teaching tertiary mathematics have led to the emergence of several scholarly communities dedicated to advancing research and practice in mathematics education. Among the most prominent is the International Commission on Mathematical Instruction (ICMI), a commission of the International Mathematical Union (IMU) — an organisation that promotes international collaboration in mathematics, including education. Founded in 1908 at the International Congress of Mathematicians (ICM) in Rome, the ICMI has since played a central role in improving mathematics education worldwide. Through conferences, research programs, publications, and international initiatives, ICMI seeks to elevate the standards and visibility of mathematics teaching and learning at all levels [97].

A 1997 ICMI study, titled 'The Working Group: Trends in Curriculum, Holton et al. [94] considered curriculum through the lens of mathematicians with a professional interest in mathematics education. Hillel [90, p. 58] examined:

> "… the various forces which act on a mathematics curriculum, and on the curriculum trends, both at local and international levels. 'Curriculum' was considered in its widest sense to mean 'matters pertaining to the purposes, goals and content of mathematical education' … as well as the means of achieving curricular programs".

Forces which act on a mathematics curriculum in this way, may be considered external, i.e. acting *on* the staff member. Here the 'curriculum' information considered included students' backgrounds, organization of teaching, resources, technology, and external influences such as government policy, where it existed.

Additionally, a contrast in perceptions of curriculum objectives was found in the differing beliefs of mathematicians and mathematics educators which may be considered internal, bringing into relief differing behaviours not just beliefs, i.e. *of* the staff member. Mathematics educators were far more likely to engage in alternative approaches in pedagogy, as illustrated by Keynes and Olson [103, p. 115], who examined examples raised from the University of Minnesota - Calculus Initiative, viz.

> "The first example shows how to create group work to clarify some of the theory and rigor behind the definition of limits. Choosing three functions, two with a limit and one not so obviously without a limit at the designated point, students attempt to find δs for various values of ϵ. The function without a limit $h(x) = \frac{|x|}{100x}$, might appear to have the lim $\frac{1}{100}$ at x=0, until you choose $\epsilon < \frac{1}{50}$, while students can of course always find a δ for the other two functions (they have limits). This type of conceptually challenging group work cautions students not to just check $\epsilon = 0.1$ and $\epsilon = 0.05$ to see if a limit exists, and has helped to make calculus students more aware of the issues in the formal $\delta - \epsilon$ process."

Keynes and Olson also found that mathematicians were more likely to rely on information transmission modes of education which have a high cognitive expectation. However, Steen [170] argues that as new mathematics emerge, change and educational renewal must occur. By way of example, the use of technology to address data sets or innovations in numerical techniques represents change to the field of mathematics, requiring change in pedagogical approaches. Large numbers of computational cycles, large computer memory requirements and large computer data stores step the educational framework to a new paradigm. Nevertheless, technology advances have not been pervasively influential in facilitating pedagogical development. This was recognized by Borwein [30], with the statement:

> "… technology has repeatedly promised to transform mathematics pedagogically …"

Also,

> "… That said, mathematics in 1999 looks a lot more like mathematics in 1939 than is the case with any of its sister sciences …"

Hillel [90] makes the point that mathematics researchers themselves are the catalyst for change [ibid, p.68]. Developments in mathematics have helped ease in change in the development of curricula, with some undergraduate courses being redefined. For example, some change has occurred in selected courses for linear algebra, differential equations and abstract algebra (for example, see [17], [71] and [69]).

However, the changing profile of incoming tertiary students unprepared to cope with the rigor of a classical undergraduate mathematics program adds to the dilemma. Tall [176] recognised the conceptual challenges reasoning and proof presented to students. Hillel [90], favouring the 'theorem-proof' approach to teaching mathematics, emphasised the goals and outcomes around tertiary mathematics teaching from the point of view of a mathematician, focusing on a teacher-centric model. Schoenfeld [160], adopting a more student-centric approach, identified causes for dysfunction in students. When students encounter topics taught in secondary school in a different context, or presented abstractly, they are not able to apply strategies familiar from secondary school. Teaching skills as opposed to introducing new concepts makes a change of context or a problem-solving assessment more difficult. Well laid-out and predictable tasks which are easily recognized allow students to cope; however, anything requiring recognition when disguised in an alternative context prevents students from drawing on known resources and using them appropriately. Artigue [17] argues that the increasing gap between secondary and tertiary mathematics education has shaped the response of universities, leading departments to adopt teaching and assessment practices that are sometimes misaligned with students' needs and learning trajectories.

Assessment can also impact stridently on curriculum. An assessment schedule requiring participation in and passing of a single assessment (like a final examination) is quite different to a schedule where the requirement to pass a single final assessment is waived. Where aggregate marks accumulate to pass a course, students are relieved of a requirement to recall in-depth learning, let alone apply it in obscure contexts in a final examination. Students who are struggling to make sense of what is being taught, just passing aggregated assessments and then foregoing deep learning assessments, are unlikely to develop advanced mathematical thinking [17].

The problems students encounter in developing advanced mathematical thinking have been subjected to inquiry and Schoenfeld [159] developed and tested global and local cognitive models. Specific to the many mathematical areas encountered by students in the secondary/tertiary transition this work resulted in research-based teaching designs refined by implementation in experimental environments. For advanced mathematics, process alone is not sufficient for satisfactory achievement. In particular, the abstract nature of mathematical concepts makes them well-suited for analysis through cognitive theories of learning, which explore how students mentally construct and internalize mathematical ideas. Theories such as the Piagetian theory of reflective abstraction [145] provide a basis for pedagogical understanding. This theory underpins the development of tasks which reflect Piagetian notions of abstraction. Students generalise from prior actions and reorganise existing knowledge structures. Integrating these cognitive theories into task design aims to promote deep conceptual understanding, rather than mere procedural proficiency, supporting more durable and transferable learning outcomes. APOS theory which is an adaption of Piaget's base idea, developed a model of thinking around

$$\text{Action} \rightarrow \text{Process} \rightarrow \text{Object} \rightarrow \text{process} \rightarrow \text{Schema (APOS)},$$

for advanced mathematical thinking. See [176] and see [18] for a detailed discussion of APOS theory. Of course, no model is complete and APOS gives a partial vision of cognitive development in advanced mathematics. Similarly, Sfard [161] gave emphasis to the dialectic between process and the structure of mathematical concepts inside mathematical activity. Sfard does not rest in the cognitive framework alone but advocates explanation as being couched firmly within two distinct fields of educational inquiry: social and cognitive. The language used by Sfard focuses on the contrast between an acquisitive, cognitive model and a participative, social learning model.

A further epistemological obstacle that students may confront is the framing of scientific understanding as not only accepted 'knowledge' but also rejected 'knowledge', usually prior knowledge. Brousseau *et al* [33, 34] imported this idea into educational research, associating cognitive breaches as epistemological obstacles ([163], [59], [175], [158]). One example is the meaning of the word ‘limit’, whose everyday meaning induces a stop or quantitative end rather than the concept of coupling sufficient closeness in mathematical relations. The transition to tertiary mathematics, and in particular the mathematics of calculus, often requires a transition from an intuitive and pragmatic approach to learning to a classic formal approach, as commonly found in undergraduate studies. As Wood [197] found in the tertiary secondary transition space, fewer students are choosing mathematics-based courses, and she concludes that internal forces, like content, are not the only reasons students struggle with mathematics in transition. Wood also highlights the attractiveness to commencing students of technology-based programs in computing, suggesting that in a competitive space there remains a great deal to be done in the mathematics domain.

2.3.2 Interactive Technologies and the Role of ITS in the Secondary/Tertiary Transition

Rather than surveying the extensive literature on educational technologies in transition pedagogy, this section focuses on interactive systems — particularly those that leverage programmability and symbolic computation — as foundational elements in the development of ITS for tertiary mathematics. In particular, the availability of Computer Algebra Systems (CAS) and the extensibility of technology via programmability, have greatly facilitated baseline research in tertiary mathematics education utilizing artificial intelligence in the form of ITS. In particular, the opportunity to improve a number of academic metrics in the tertiary transition space by application of an ITS in the mathematical domain presents itself.

Several earlier investigations have concentrated on middle and secondary school mathematics education with initiatives such as the cognitive tutor developed from 1989 as the Pittsburgh Urban Math Project (PUMP), [58], the aim of which was to develop a new algebra curriculum consistent with the new standards introduced in that year. These were primarily the National Council of Teachers of Mathematics (NCTM) Standards. Specifically, the 1989 NCTM Curriculum and Evaluation Standards [131] for School Mathematics served as the driving force. These standards emphasized:

- Conceptual understanding over rote memorization
- Problem-solving and reasoning
- Real-world applications of mathematics
- Use of technology in mathematics instruction

That curriculum became the starting point for the PUMP Algebra Tutor (PAT). Following the Education Reform Law of 1993, the Massachusetts Comprehensive Assessment System (MCAS) was implemented and the ASSISTments Project, [150] was a response to the mathematics component of MCAS. By comparison with local Australian curricula [137], the targeted curriculum equates to the New South Wales (NSW), stage 4 (years 7 and 8) and stage 5 (years 9 and 10) secondary mathematics curriculum. More recently the Wayang Outpost Tutor, Algebra Tutor and Geometry Explanation Tutor have all been developed to serve the needs of middle and secondary schools, based in the United States [200]. However, while many ITS excel in procedural skill acquisition, they often falter in domains requiring symbolic flexibility and equivalence recognition [118, 125]. For example, systems frequently fail to recognize that factored and expanded expressions, though algebraically equivalent, represent the same mathematical object [23].

An alternative approach is the development of computer programs for use by students to investigate individual topics within these stages of the curriculum. However, whilst this can allow students to avoid complex numerical techniques, the computer program implementation can use routines developed using a wide variety of possible computer programming and scripting languages, including C++, Java, PHP, Python, Perl and Java-script. Advanced stages in years later than those mentioned, which study calculus, require a more complex approach requiring either programming languages with well-developed mathematical libraries or at least executive functions able to implement routines from languages that have those mathematical capabilities.

A contrasting alternative is to embed within an ITS a symbolic mathematical engine, such as Maxima © or Mathematica ©. One such system is the commercially available program Mathematica, which employs notebook files for user interaction. Far from being a simple CAS, Mathematica, employs an extensible interface built around an internal and developing programming language library which is both peculiar and unique. This is the approach adopted in this thesis.

Developing a system of their own, Langtry *et al* [112] produced a set of interactive notebooks to support student engagement with particular aspects of the transition curriculum. Their system used the native Mathematica notebook product and in a later work, Zinder and Langtry [206] employed the extensibility of Mathematica to create interactive software to assist student learning of higher-level mathematical techniques. The notebook interface, however, imposes an additional learning burden on students, which impacts its suitability as comprehensive ITS platform in the transition space.

2.4 Educational Theory and ITS

In this section we will consider the development of aspects of learning and educational theory that underpin the development of an ITS such as those referred to in Section 2.5.

In the post World War 2 period the United States passed laws designed to encourage returning soldiers to participate in educational programs. Commonly referred to as the GI bill, this enactment resulted in educational institutions undergoing unprecedented growth. Information technology played a pivotal role in the global conflict. Alan Turing's early work on what would later be called computing machines was central to deciphering enemy encrypted communications, particularly through his contributions at Bletchley Park during World War II [54].

Post-war educational pressure and psychological intervention In the post-war period, educational institutions faced significant pressure to accommodate the reintegration of returning service personnel into civilian and academic life. Although digital learning technologies were not yet available, psychologists played a pivotal role in meeting the demand for large-scale, efficient training. As Berliner notes, strategies developed to address this challenge in public institutions shaped textbook content and came to characterise mainstream educational practices of the time [26, p. 47].

Thorndike and the roots of behaviourist learning theory Among the most influential figures of this period was Edward Thorndike, whose early work laid the foundation for modern learning theory. Central to his approach was the concept of Stimulus–Response (S–R) association, formalised in his *Law of Effect*, which posits that responses followed by satisfying outcomes become more strongly connected to their stimuli, while those followed by discomfort are weakened [32]. Thorndike's famous experiments — such as placing cats in puzzle boxes — illustrated that learning was essentially trial-and-error behaviour reinforced by consequences.

From S–R to behaviourism in education Thorndike's principles became foundational for the behaviourist movement, which dominated educational psychology throughout much of the 20th century. Behaviourist instructional design emphasised observable outcomes, repetition, reinforcement, and feedback — elements that translated well into institutional settings focused on measurable progress and control. As Schunk explains, these foundations provided a conceptual framework for subsequent instructional models that prioritised behaviour change through structured input–output training cycles

Behaviourism's influence on ITSs These behaviourist principles not only influenced mid-century classroom practices, but also helped shape the architecture of early computer-based learning environments. As computing capabilities expanded, educational researchers adapted behaviourist logic to machine-based instruction, ultimately contributing to the development of ITS. While contemporary ITSs have since incorporated constructivist and cognitive models, their roots can be traced to this early emphasis on stimulus-driven feedback and reinforcement cycles.

Research into technology-based education systems grew due to the increased demand from the public education system. Computer-Assisted Instruction (CAI) systems that were generative [172] began to emerge, though these systems did not explicitly engage with theories of how people learn [185]. Instead, they were underpinned by behaviourist and early cognitivist models that framed learning as a process of information transmission and reception—analogous to the traditional lecture model. The underlying assumption was that simply presenting information would result in learning. Coincidentally, educational psychologists were questioning the assumptions of behaviourism which had the two prominent psychological theories of structuralism and functionalism. Theories of learning and constructivism [144] were successfully challenging the 'environment alone' schema of behaviorism. This learner-centric model included the notion of learner re-construction of information, leading to and including discovery by doing. Although language specialists like Chomsky [47], along with Newell [132] and others, rejected the constructivist model for language, they introduced their own ideas of symbolic information processing [83], ideas that dovetailed with the AI community's interests in linguistics and natural language processing.

Information Processing (IP) was emerging as a dominant paradigm for AI, conceiving human cognition as a "black box" of internal symbolic processes [133]. The processing model was already understood by computer science, and it was easy to understand how an educational view of cognitive processing was adapted for use in AI. Sleeman and Brown [165] reviewed the state of the art in computer aided instruction and are credited with introducing the term an Intelligent Tutoring System (ITS). This renaming of systems allowed them to be distinguished from the existing CAI systems. They classified ITS as being computer-based:

1. problem-solving monitors,
2. coaches,
3. laboratory instructors, and
4. consultants,

with the implicit assumption that the learner should now be focused on learning-by-doing.

The emphasis in these systems was still as research platforms for refining AI theories, but now researchers were thinking about representing student knowledge within these systems. Student Models (which we will discuss in detail in Section 2.5.2) became more complex [165] employing an abstract representation of the learner within the computer program. An interesting component of the model [41] was the descriptor *Perturbation* representing student misconceptions, rather than relying solely on the deficit model of student knowledge against instructor knowledge.

Early attempts to model student knowledge using a perturbation framework were based on a "buggy" model first proposed by [35]. Student errors, which could be thought of as a catalogue of assessment distractors, once correctly identified could be used by the system to focus on remediating student deficits. The underlying skill or knowledge was classed as an element and then the system would match the element with a task to redress the error.

Sleeman and Brown [165] mention some learning issues related to the problems involved in creating ITS. They acknowledged that much human tutor communication is implicit and they expressed the hope that ITS will provide an avenue for educational theorists to develop "more precise theories of teaching and learning" (ibid p.9). Their assumption was that such precision is possible, being necessary for the implementation of these theories within computer software. They also discussed the need to construct environments that encourage collaborative learning, while acknowledging that researchers (at that time) knew little about how such cooperation takes place in natural learning settings.

John Anderson's [13] work in cognitive science developed the Adaptive Control of Thought (ACT) theory of cognition. Example ITS systems implemented include the Geometry Tutor [106] and LISPITS (LISP Intelligent Tutoring System) [55].

The LISPITS system, a program for teaching LISP programming, was designed to implement the principles of "model tracing". LISPITS attempts to model the steps needed to write a LISP program. The program then compares the actual steps that the student takes with this model. Corbett and Anderson [55] call the monitoring and remediating process 'knowledge tracing'. They found that students using LISPITS completed the mastery model exercises considerably faster than students who worked alone, but not as fast as students who worked with human tutors. Such systems are often referred to as "Anderson-style tutors" [46, p. 352]. Model tracing is not the only paradigm for ITS development. Subsequent development has led to other paradigms such as constraint-based tutors. A summary of model tracing versus constraint-based tutors follows on the next page. These models will be considered in more detail in Section 2.5.

Table 2.1 A comparison of the main features of the most used approaches to ITS.

Feature	Constraint-Based ITS	Model-Tracing ITS
Knowledge Representation	Declarative constraints representing domain principles	Procedural cognitive models representing expert strategies
Evaluation Method	Checks for violations of constraints	Traces student steps against expert model paths
Student Error Diagnosis	Based on which constraints are violated	Based on deviations from expected procedural steps
Flexibility	Tolerant of multiple correct solutions	Requires pre-defined correct solution paths
Feedback Style	General (e.g., "This rule was violated")	Specific (e.g., "This step should be...")
Instructional Approach	Detects and Prevents rule violations	Guides student through step-by-step modeling
Domain Suitability	Well-suited for declarative domains (e.g., databases, logic)	Well-suited for procedural domains (e.g., algebra, programming)
Example Systems	SQL-Tutor, KERMIT	Cognitive Tutor (Carnegie Mellon)

The two most prominent paradigms in the development of ITSs are model tracing and constraint-based modelling. While both aim to provide adaptive, individualised instruction, they are grounded in distinct theoretical assumptions and technical architectures. Model-tracing tutors rely on procedural cognitive models that trace a student's problem-solving steps against a pre-defined expert path, enabling fine-grained feedback on each step. In contrast, constraint-based tutors evaluate student solutions based on domain-specific rules or constraints, allowing for multiple correct solution paths and more generalised feedback. Table 2.1 provides a comparative summary of the key features, diagnostic strategies, and instructional approaches associated with each model.

Rosenberg [153] noted that most papers about ITS make few references to the education literature; the majority are grounded in the computing literature. He asserts that much ITS work suffers from two major flaws: First, the systems are not grounded in substantiated models of learning. Second, ITS models should be validated by the teachers and students who will use the systems. Positive claims about educational outcomes should be critically evaluated as Rosenberg cautions [153, p. 11]. During the same time period, constructivism was becoming a dominant theme in educational psychology [193]. Constructivists [99] approached cognition from a more holistic perspective than either the behaviourists or the information processing paradigm, claiming that cognition could not be reduced to the interactions among a number of black boxes because doing so fails to account for the learner's reflecting on his or her cognitive strategy. Constructivists claim that the underlying computer metaphor — viewing the mind as a rule-based processor — makes it difficult to model reflection and self-awareness since computers are not self-aware [82, 189]. By 1987, a review of the field by Wenger [192] demonstrated how much it had evolved in the five years since Sleeman and Brown's synopsis. Wenger examined the goals of ITS designers. Significantly, Wenger focused on cognitive and learning aspects of these systems, in addition to the AI issues. Wenger proposed what might become the basis for a discipline that combines the work of researchers from AI, cognitive science, and education. Wenger points out two opposing views of ITS: the traditional view of computers as instructional delivery devices, and the then emerging view of computers as tools for exploratory learning. He claims that by viewing ITS's as knowledge communication tools it is possible to merge these apparently opposing views of ITS.

Student Modeling is the distinguishing feature of ITS research [93] and helps to distinguish ITS from CAI. There are impediments to ITS development: ITS are complex and the models are difficult to implement. Also ITS is not without critics, either because of technical limitations [116] or larger philosophical grounds [155]. Mitchell argues that an ITS must model the world, the learner, and the teacher-learner interaction [122]. There has been much research on different aspects of this problem (see [117] and [84]).

According to Wenger,

> "… [when] learning is viewed as successive transitions between knowledge states, the purpose of teaching is accordingly to facilitate the student's traversal of the space of knowledge states." [192, p. 365]

In this conception, an authentic ITS must both model the student's current knowledge state and support their progression toward more advanced understanding. Wenger suggests that achieving this requires the system to alternate between diagnostic and didactic functions, identifying learner needs while also delivering targeted instructional support. This dynamic responsiveness is essential for any system that aims to be both adaptive and pedagogically effective.

These ideas resonate with foundational insights from the Eastern European tradition of educational psychology — particularly the work of Vygotsky — which profoundly influenced Western thinking on how learning unfolds. As Cherry notes, Vygotsky emphasised that educational development cannot be separated from the cultural and contextual environment in which it occurs [44]. From this perspective, learning is not solely an intra-personal process but is fundamentally shaped by social interaction and participation in shared practices.

Central to Vygotsky's theory is the concept of the Zone of Proximal Development (ZPD) — the space between what a learner can achieve independently and what they can achieve with appropriate support. In the context of ITS design, this has powerful implications: the system must not only identify the learner's current state but also provide scaffolding that operates within this zone. Adaptive interventions aligned with the ZPD allow for the gradual internalisation of skills, helping learners to move beyond mere correction of errors toward genuine conceptual growth.

Constructivist Revisions in Early ITS Research: One eminent Western ITS research team, led by Elliot Soloway and colleagues at Yale University through the Cognition and Programming Project [167] focused on teaching computer programming to novice students — particularly with an interest in how problem-solving skills developed through programming could transfer to other domains [166]. This research acknowledged that programming often involved solving problems situated in various application areas, not strictly limited to computing itself. The insights gained from this work carried over into broader ITS research and led to the development of an extensive program at the University of Michigan known as the Highly Interactive Computing Environments (HiCE) Group.

The HiCE Group was established to explore how ITS technologies could support novice programmers. A central concern was effective Student Modelling — specifically, how to identify and classify student errors (or ”bugs”) and to assist students in recognising and correcting them. This bug-focused strategy aligned with what is often called perturbation analysis, wherein the system identifies meaningful deviations between student responses and expert solutions and provides feedback accordingly [9, 36].

However, over time, Sack et al. [155] began to critically reassess their assumptions about the nature of learning and Student Modelling. In their later reflections, the HiCE researchers acknowledged that their earlier models were grounded in a relatively narrow, instructivist view of learning — one that conceptualised learning as the transfer of knowledge and the correction of “buggy” mental models. This framing assumed that student conceptions needed to be brought into alignment with expert solutions through systematic intervention.

Yet their experiences with real-world classroom implementations of ITSs prompted a fundamental shift in perspective. They came to recognise that learning was not best understood as simple information transfer or error correction, but as a more situated process of adaptation and participation. As students engaged with the learning environment over time, they gradually adjusted their thinking not simply by internalising expert models, but by participating in the practices and language of the surrounding knowledge culture. In this view, learning became a process of enculturation into a community of practice. The HiCE team thus moved away from a model of learning as the correction of misconceptions toward a model grounded in constructivist principles. As they note, “the direct consequence of this adjustment was to re-engineer their Student Models” to reflect a more nuanced understanding of learner development [155, p. 373].

More recent research into ITS has included affect models. This is a development of an established social construct around Student Modelling which is organized to establish and then keep students in their Zone of Proximal Development or ZPD [129]. Woolf [200] explored student-centric models for use in ITSs. His treatise explores learning, AI technology and the objectives of an ITS, whose design is, by its very nature, interactive. As stated by Corbett [58]:

> “The goal of Intelligent Tutoring Systems (ITSs) would be to engage the students in sustained reasoning activity and to interact with the student based on a deep understanding of the student’s behavior.”

The promise of ITS is significant. The realization of that promise is not without challenges. One example of the challenges in the field is the contrast in explanation and orientation of the researchers examining ITS. For example, Corbett [58] heads a section in his article as “37.2.1 Research Goals: AI vs. Education” whereas [200], embraces the pedagogical developments which are increasingly being seen as central and strategic to the development of ITS, not simply adjunct criteria to be fulfilled as a tactical response to educational implications.

Early evaluation of ITS, was designed to match AI criteria, [58, p. 850] and as early AI research was undertaken by psychology researchers this is not surprising. One feature of the research was to explore possibilities in a number of domains and this led to increased participation by researchers from fields of education, also known as ‘domain fields’.

2.5 ITS Development

2.5.1 Institutional and Transition Students' Needs

The landscape of mathematics education, particularly as it pertains to tertiary preparedness, has undergone substantial change in recent decades. One notable trend is the marked decline in enrolment in calculus-based courses among secondary school leavers, both in Australia and internationally [197] and more recently [149, 199, 198]. This shift may carry further consequences: while students are opting out of advanced mathematics, the resulting lack of engagement with foundational algebra and pre-calculus content has arguably contributed to a widespread erosion of procedural fluency in these essential areas. Although a definitive causal study has yet to establish a statistically significant link between the absence of calculus instruction and deteriorating pre-calculus competencies, anecdotal and institutional evidence suggest that the correlation warrants deeper investigation.

In the Australian tertiary sector, institutions are increasingly reporting dual challenges (see [174, 181, 180, 182, 184, 179]). First, a decreasing number of students are selecting STEMM (Science, Technology, Engineering, Mathematics, and Medicine) pathways at both secondary and tertiary levels. Second, and more alarmingly, many students entering university programs are doing so without any formal background in calculus —– and in a growing number of cases, without any recent exposure to mathematics at all. This academic underpreparedness is not an isolated national phenomenon; rather, it reflects a broader global trend in mathematics disengagement that has been noted by international bodies such as the International Commission on Mathematical Instruction [97]. These issues are compounded by socioeconomic, curricular, and pedagogical pressures, which collectively diminish students' confidence and ability in mathematical reasoning.

Domestically, the removal of the Stage 5 School Certificate in New South Wales (a benchmark assessment historically used to measure student performance at the end of Year 10) has left a vacuum in systemic diagnostics. In its place, the National Assessment Program – Literacy and Numeracy (NAPLAN) was intended to serve as a nationwide benchmarking tool, with the aim of offering policymakers and educators the data needed to refine curricular strategies and student interventions [19]. In principle, such a standardised instrument could identify achievement gaps early and enable targeted remediation. However, in practice, results have been mixed. Recent analyses show that despite the extensive data collected, NAPLAN has not reversed the trend of declining mathematics performance [152].

In the context of university mathematics readiness, the lack of robust preparatory structures seemingly contributes to a difficult entry point for many students. A range of studies have identified that under-preparedness in symbolic and algebraic reasoning is a significant barrier to success in tertiary mathematics programs [37, 81, 204]. Mathematics enrolments have also declined significantly in recent years, not merely due to waning interest, but because many students feel alienated from the discipline as a result of inadequate prior learning experiences [24, 2]. Research has shown that negative prior experiences and a lack of confidence in mathematics, especially among students from underrepresented backgrounds, contribute to reduced participation and engagement in mathematically intensive university courses [204, 81]. Sustaining broad and inclusive participation in university-level mathematics—particularly in subjects requiring algebraic and symbolic fluency—appears increasingly unsustainable without systemic intervention and better transition support mechanisms [81, 37, 204].

Consequently, universities are developing internally driven initiatives to mitigate this challenge. These programs often aim to identify underprepared students and provide early-stage support mechanisms that scaffold their mathematical development. One such initiative is the *Pathway Program* at the University of Technology Sydney (UTS), which integrates directly into the student onboarding process. Central to this program is the Mathematics Readiness Survey (MRS), a diagnostic instrument designed to evaluate incoming students' grasp of fundamental mathematical concepts required for success in their enrolled courses. Rather than serving merely as a passive placement test, the MRS influences actual enrolment and course placement decisions.

For some students, this intervention occurs at a highly transitional moment. The move from secondary to tertiary study already poses substantial cultural and academic challenges; to be immediately confronted with a mathematics diagnostic that determines their progression can amplify anxiety and self-doubt. Nevertheless, the program is a deliberate attempt to shift the narrative from reactive remediation to proactive preparedness. It reflects an emerging commitment within higher education institutions to design ITS and other digital interventions that are closely aligned with the actual needs and characteristics of their student body.

As the ITS research community continues to develop adaptive learning environments, there is a need to better understand the diverse and evolving needs of their students. This includes students who are not traditional consumers of advanced mathematics, but whose disciplines increasingly require quantitative reasoning and symbolic problem-solving. Mapping these trends is essential not only for system design but also for ensuring that technological interventions align with both institutional goals and learner needs.

2.5.2 ITS Design Principles

According to Murray [128], four main components are required to realize a basic or entry level ITS: the *student interface*, the *Domain Model*, the *Teaching Model*, and the *Student Model*. We note here that the term *Tutor Model* is often used in place of *Teaching Model* in the wider literature. In this thesis we use the term *Teaching Model*.

How they are formed, coupled and evolve all help to characterize the type of ITS being employed. The main models developed are the *domain*, *teaching* and *student*. However, there are differing approaches to formulating these and, as previously detailed, models first evolved through the analysis of student responses and mapping those to the intended curriculum objectives [9, 188, 56, 123, 128]. The main models have taken various forms, but a common thread across many approaches is their basis in empirical analysis of student responses and their alignment with curriculum objectives.

In model-tracing architectures, for instance, student actions are mapped onto expert-defined cognitive procedures, enabling real-time diagnosis of misconceptions in relation to instructional goals [9, 56].

Constraint-based tutors evaluate student inputs against domain constraints that explicitly encode the structure of the intended curriculum [123].

More broadly, the behavior of tutoring systems evolves through iterative refinement driven by observed student performance patterns [188].

Student interface The student interface begins at the ITS web page landing point and includes all subsequent web page transitions, interactions, and updates. Modern web-based techniques employ highly developed scripting libraries with high levels of interactivity made popular by commercial forces in the web market place. Such authoring tools are readily available to be tailored to create an educational interface with the latest social media style of interactivity.

Domain Model The Domain Model is the discipline field the ITS is addressing e.g., physics, chemistry, mathematics, biology, forensics etc. In any of those domains listed, curricula describe large bodies of knowledge and skills. The Domain Model maps out the knowledge and skills for a given curriculum. Domain expertise can include several types of knowledge, including:

- problem solving expertise,
- procedural skills,
- concepts and
- facts.

The content of the domain can be organized into networks to map domain information into the model of curriculum structures. Generally, knowledge and procedural skills lie at the core of mathematicians' staple requirements for success in the field, see Appendix H.

These intelligent tutoring approaches have been successfully applied across a range of disciplines, including mathematics, physics, chemistry, biology, computer science, and engineering. In physics, the Andes Tutor provides a coached problem-solving environment for Newtonian mechanics, using model-tracing to evaluate stepwise student input [80]. In chemistry, the ChemCollective offers virtual laboratories and cognitive tutors that align with curricular outcomes in topics such as stoichiometry and equilibrium [202]. In biology, ITSs like those developed by Koutsojannis et al. [110] employ interactive case-based reasoning and diagnostic activities to support the learning of complex systems . In computer science education, platforms like J-LATTE and CTAT (Cognitive Tutor Authoring Tools) support the teaching of programming and algorithmic thinking through scaffolded, feedback-driven tasks [4]. In engineering, ITS such as PAT (Practical Algebra Tutor) and general investigation of foundational sciences begins in secondary settings to guide learners through applied problem solving in design and analysis contexts [105].

Teaching Model The Teaching Model is the approach used in education as it is intended in the ITS. This model defines how feedback and support mechanisms are enlisted, as well as the type of feedback and the degree or granularity of the steps and knowledge being maintained. Some tutors use a rule-based representational method which requires a tree and branch approach in which the author of the system assigns and specifies the right and left-hand components of IF-THEN rules. Some tutors ([123], [11], [56] and [108]) use a flow-line based approach, not unlike a project management Gantt chart, to assign arbitrary instructional procedures. Another option is to use a fixed rule set defining the pedagogical behavior. The defined "teaching strategies," are settings for key pedagogical parameters such as "amount of student choice," "preference for specific (vs. general) information," and "amount of feedback". For example, a strategy called "Advanced learners" might have high student choice, low preference for specific information, and feedback.

Student Model The Student Model is less easily stated. A Student Model is an abstraction representing what a learner knows, misunderstands, or is in the process of learning. The representation is a reflection of the current position in the method being applied to navigate curriculum in the ITS. Updates to the model are triggered by student actions and used to inform instructional decisions at pre-defined method milestones and this aspect is discussed in more contextual detail in Section 5.5.3. In this project, the aforementioned milestones include interpreting symbolic steps as evidence of procedural understanding or conceptual progress. More simply, a stored session in a database record, houses a snapshot at the culmination of the student journey to and from the aformention milestone, i.e. a Student Model. The implementation of the Student Model used in Tutoria is discussed in Section 5.3.2.

In order to achieve and manage a snapshot as introduced above, Student Models can become very complex ranging from set groups of objects based on a myriad of combinations to more adventurous models invoking AI modelling techniques, a prevalent one being Bayesian networks [52]. To belay the possibility of over complicating models, Student Models [43] often use the idea of overlays to characterise the model. Overlays are transparent layers that form a cumulative representation of student knowledge by mapping which domain elements have been acquired. While elegant in their simplicity, overlays assume a student's knowledge is a subset of the expert's knowledge—a view that has been critiqued for its inability to represent alternative conceptions or incorrect reasoning paths [40]. Consequently, more sophisticated modelling approaches have evolved to account for the nuances of human cognition, learning variability, and pedagogical goals.

One such extension is the use of *buggy models* or *perturbation models*, which aim to represent common misconceptions or faulty reasoning strategies. Originally explored by Brown and Burton [35], these models allow ITS to respond not just to missing knowledge, but to specific types of errors, enabling corrective feedback tailored to a learner's internal logic. Such models can be pre-defined based on domain analysis or dynamically generated as the system observes the student over time. Although rich in diagnostic potential, buggy models are often labour-intensive to build, requiring substantial domain knowledge and learner data.

A prominent line of development in Student Modelling has been the integration of *model tracing* and *cognitive models*, particularly in systems like the Cognitive Tutor [11]. Model tracing involves following the student's step-by-step actions through a rule-based representation of expert strategies. If the student diverges from an expected rule, the system can intervene with targeted feedback. These models offer high fidelity to expert performance and allow granular tracking of problem-solving behaviours. However, they rely on extensive hand-coding of production rules and often struggle to accommodate novel or unanticipated student strategies.

In contrast, *constraint-based modelling* (CBM), introduced by Mitrovic [123], adopts a more flexible stance. Rather than tracing ideal sequences, CBMs define a set of domain constraints that must be satisfied by any correct solution. Student answers are evaluated against these constraints, and violations are interpreted as errors. This approach requires less scripting of ideal paths and can be adapted across domains with well-defined logical structures. Constraint-based models are particularly well-suited for domains like SQL, logic, and mathematics, where constraints can be formally expressed.

Another powerful framework is the use of *Bayesian networks* in Student Modelling [120]. Bayesian models represent knowledge and misconceptions as probabilistic relationships between concepts, allowing systems to infer the likelihood that a student has mastered certain topics based on observed behaviour. These models are robust in the presence of uncertainty and allow dynamic updating as more student data becomes available. Importantly, they also support decision-making under partial observability—an essential feature for adaptive learning environments. The drawback is that Bayesian networks can be computationally expensive and require carefully calibrated prior distributions and conditional dependencies.

With the recent growth in educational data mining, *machine learning* and *deep learning* approaches have entered the landscape of Student Modelling [66]. These data-driven methods aim to identify latent patterns in student responses without relying on expert-defined structures. Techniques such as knowledge tracing with recurrent neural networks, matrix factorization, and dynamic Bayesian modelling are now being used to track student understanding across time and tasks. While these approaches offer scalability and predictive accuracy, they also suffer from limited interpretability, often functioning as "black boxes" with minimal insight into the cognitive processes they approximate.

Despite the variety of modelling approaches, developing an effective Student Model remains a core challenge in ITS design. As Sani and Aris [157] note, the modelling task involves not only capturing what students know, but how they think, err, and improve. Effective Student Models must balance precision, adaptability, diagnostic power, and pedagogical relevance. The choice of model type is influenced by the educational context, available data, computational resources, and instructional goals.

Ultimately, the future of Student Modelling may lie in hybrid approaches that combine the explanatory clarity of cognitive models with the adaptability of data-driven methods. For instance, recent frameworks integrate symbolic reasoning engines (e.g., rule-based diagnostics or constraint checkers) with neural architectures that adapt based on learning trajectories. In mathematics education specifically, where symbolic and conceptual understanding are tightly interwoven, such hybrid models offer a promising pathway toward more nuanced, personalised, and effective tutoring systems.

Types of tutor

Model Tracing Tutor 'Anderson tutors', otherwise known as model tracers, have evolved to optimize the link between teacher intention and student performance. Typically, models of teaching revolve around methods of instruction and are also known as tutoring models. The resident entities include content, content sequence, items for explanation, coaching, remediation, summary, problem development and feedback. A number of representations are used to model tutoring expertise mostly around rules, constraints, plans and procedures [128, p. 107]. Model tracing tutors collate student responses against the catalogue of known answers and thus form a model of student performance, [3, p. 219]. This may sound like a manual teaching assessment, however the breadth and depth of the 'tested' unit is considerably greater than a one-off unit class test as the variety and number of entities is drawn from the intended curriculum. Each student has a reflection of the Domain Model individually mapped, representing a layered knowledge base for the student.

Constraint Based Tutor (CBM) Mitrovic *et al.* [126] detail the 'constraint based' approach based on Stellan Ohlsson's theory of learning [139]. Learning takes place in a *deficit model* where students must make mistakes and these mistakes are recorded as performance errors. Subsequent analysis resulted in the methodology known as constraint based modeling (CBM) [138]. Mitrovic states,

> "… knowledge should be represented in the form of constraints, which specify what ought to be so, rather than generating problem-solving paths. Domain knowledge is thus used as a way of prescribing abstract features of correct solutions, rather than as a recipe for performing tasks in a domain, the way it is done in model tracing (using production rules)".

Constraint-based modeling avoids the need to model students' misconceptions - that is, it is not necessary to attend to problem solving paths as in model tracing.

Constraints have three components:

- a *relevance* condition,
- a *satisfaction* condition and
- the *feedback* message.

If a student performs an action that is not on a known path, a model tracing tutor will typically assume the answer is incorrect (although it might not know why); CBM is more permissive than model tracing. Constraints are used to represent the domain knowledge by specifying features of correct solutions and they also serve as the basis for representing student's knowledge. When a student submits a solution, a constraint-based tutor analyses the submission using the available constraints; relevant constraints are identified, and their satisfaction conditions determine whether they have been satisfied or not. The lists of relevant, satisfied and violated constraints are used as a short-term student model, which is then used to update the long-term model of the student's understanding. A student's knowledge may be represented in constraint-based tutors in a variety of ways. As has already been mentioned, as an overlay on top of the Domain Model, as a set of performance histories for all constraints used by the student, or even a Bayesian Student Model. For a detailed discussion of tutors see [1].

2.6 Case Studies in Transition Mathematics ITS

2.6.1 State-of-the-Art Case Studies

Cognitive Tutor (Carnegie Mellon University) One of the most influential examples of ITSs is the Cognitive Tutor developed at Carnegie Mellon University. Based on ACT-R cognitive architecture [10], it tracks students' problem-solving actions step-by-step using model tracing. The system has demonstrated significant success in procedural domains such as algebra and geometry, helping students learn by monitoring their operations in real time and comparing them to an ideal solution path. However, the system is fundamentally constrained by its reliance on production rules that mirror expert solutions. When students approach problems using unanticipated or conceptually novel strategies—often common in advanced transition mathematics—the system may fail to validate these as correct, even if they are logically valid. Thus, while highly effective for skills practice, the model's rigidity limits its applicability for higher-order symbolic reasoning required at the university entry level [11].

SQL-Tutor and ActiveMath Constraint-based tutors such as SQL-Tutor [123] and open learner modeling systems like ActiveMath [118] are built on the assumption that learners will violate a known set of constraints. These systems are adept at pinpointing where a student's response violates a domain-specific rule, offering targeted feedback and opportunities for correction. In mathematics education, such systems can handle formulaic content and provide adaptive support to learners who make common algebraic or arithmetic errors. Nonetheless, when it comes to symbolic equivalence—such as recognizing that $x^2 - 4$ and $(x-2)(x+2)$ are interchangeable in an algebraic context—the underlying logic of these tutors often break down. They tend to assess surface structure rather than deep mathematical equivalence. Furthermore, open learner models still largely rely on pre-coded forms, making them unsuitable for expressions that deviate from canonical or anticipated forms [123, 118].

ALEKS and Knowledge Space Theory Systems ALEKS (Assessment and Learning in Knowledge Spaces) is an intelligent learning and assessment platform grounded in Knowledge Space Theory (KST), as formalized by Doignon and Falmagne [68]. Unlike traditional model-tracing or constraint-based systems, ALEKS uses a probabilistic representation of a learner's knowledge state to infer what a student knows, does not know, and is ready to learn next. The system is particularly effective for structured domains such as arithmetic, algebra, and introductory statistics, where knowledge can be organized hierarchically [60, 77]. The platform excels in diagnostic precision, assessing students through a series of adaptive questions that map them onto a specific knowledge state within a well-defined space. This enables targeted remediation and personalized learning pathways. Research shows that ALEKS can significantly improve student retention and success, particularly in foundational mathematics courses [85]. However, ALEKS's underlying design inherently treats mathematical knowledge as a collection of discrete, isolatable items. While this supports efficient assessment, it limits the system's capacity to recognize symbolic transformations or expressions of conceptual understanding in varied forms. For example, it may not recognize that $\sqrt{x^2}$ and $|x|$ are equivalent under certain conditions, or that $(x-2)(x+2)$ is an acceptable form of x^2-4 in a factorization context. Consequently, the system's binary evaluation of mastery can misclassify students' actual understanding, especially in domains where algebraic structure and symbolic reasoning are critical [121]. This makes ALEKS less suitable for higher-order or transitional mathematics, where flexibility of representation and conceptual generalization are essential.

2.6.2 Shortfalls in More Complicated Mathematics

While many ITS excel in procedural skill acquisition, they often falter in domains requiring symbolic flexibility and equivalence recognition [118, 125]. Systems frequently fail to recognize that factored and expanded expressions, though algebraically equivalent, represent the same mathematical object [23]. This inflexibility not only constrains the learner's expression but may also reinforce surface-level reasoning strategies [21, 45]. Furthermore, as learners progress into university-level mathematics, they are expected to shift from following steps to understanding structural relationships, a shift most current systems are not equipped to support [57].

Rigid Input Parsing A dominant issue in current ITS design is the rigidity of input parsing. Many systems are built around multiple-choice, number-entry, or templated text boxes that accept only one or two expected forms. This restricts students' ability to express partial understanding or propose novel strategies. It forces learners into an artificial mold of "correctness" defined by surface patterns rather than semantic meaning. As students progress into more sophisticated topics like calculus, symbolic algebra, or logic-based proofs, this restriction increasingly conflicts with the needs of expressive reasoning, where different but valid formulations are part of expert discourse.

Equivalence Blindness Existing ITS architectures often lack the ability to determine whether two mathematical expressions are logically or algebraically equivalent. This problem arises not only with factoring and expanding, but also in trigonometric identities, limits, derivatives, and integrals. The inability to validate equivalence means that students who demonstrate understanding through non-canonical forms—perhaps using distributive laws, substitutions, or alternative simplifications—may be penalized or confused. This blind spot becomes particularly problematic in transitional mathematics, where flexibility of form and symbolic manipulation are essential parts of learning how to think mathematically.

Step Inflexibility and Limited Feedback Many intelligent tutors are designed to track student progress along a predefined solution path, often derived from expert demonstration. As such, they may reject alternative solution paths as invalid. This issue is amplified in transition mathematics, where multiple approaches are valid and often encouraged. In addition, existing feedback mechanisms tend to be procedural rather than conceptual. Instead of explaining why an error occurred in terms the student understands, or recognizing a partially correct transformation, they provide template-based responses that miss the nuance of student reasoning. This both limits formative feedback and diminishes student agency.

2.7 Concluding Remarks

Although ITS implementation is complex and draws on a number of disciplines for technique and knowledge, the ITS field is grounded in three (+ one) disciplines:

- computer science
- psychology
- education.

then add to this the contextual discipline:

- progamming OR physics OR chemistry OR forensics OR pilots OR statistics OR soldiers OR surgery OR teaching OR (in this case) mathematics...

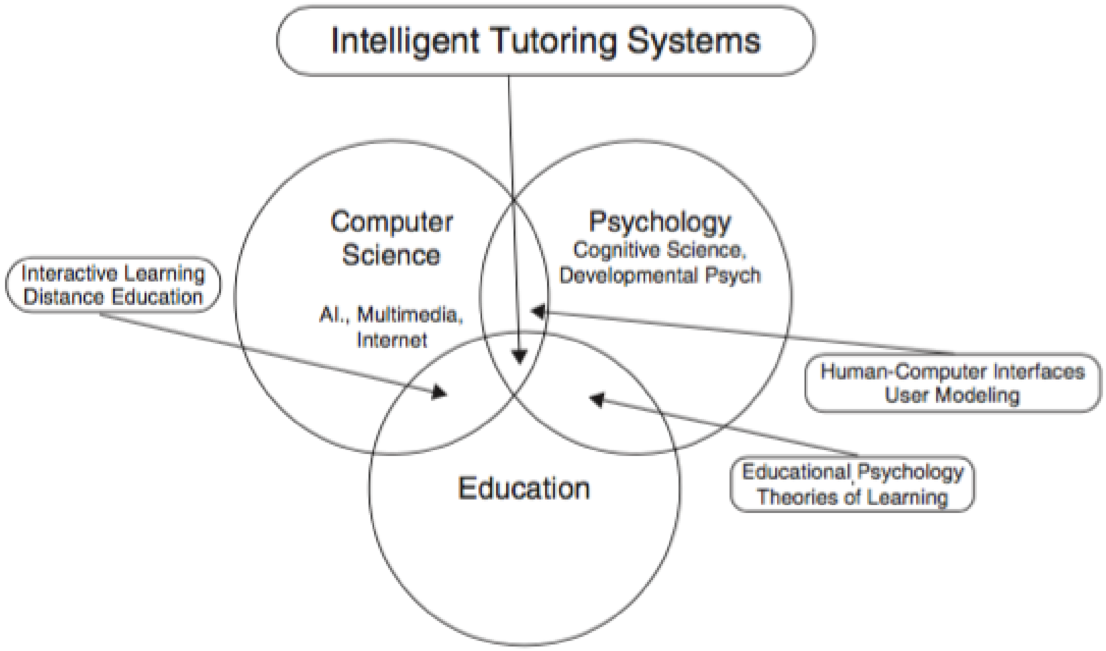


Fig. 2.1 System diagram of ITS components

As seen in Figure 2.1, each of the disciplines contributes to the development of an ITS. The link between psychology and education is well established, and the connection between psychology and computer science is steadily maturing. However, while considerable progress has been made in ITS for middle school mathematics, applications in advanced mathematical domains remain in their infancy.

Novel Intelligent Tutoring Systems offer significant potential to address the well-documented epistemological and cognitive obstacles that students encounter during their first year of university mathematics. Foundational research has shown that students struggle with the transition from intuitive, process-oriented thinking to formal, object-based reasoning, particularly in areas such as limits, functions, and abstract structures. By integrating advanced Student Modelling, adaptive feedback, and conceptual scaffolding, modern ITS designs could more effectively support learners in overcoming these deep-seated difficulties. Such systems could also help bridge the gap between everyday reasoning and the formal rigor demanded by higher mathematics, ultimately fostering stronger conceptual understanding and greater mathematical resilience [176, 161].

An effective Intelligent Tutoring System could be purposefully designed to serve a dual function: supporting current tertiary students in their university-level mathematics studies while simultaneously assisting secondary students who are preparing for the transition into higher education, see Appendix H. By tailoring the system's adaptive feedback, content progression, and conceptual scaffolding to address the needs of both groups, it could bridge the critical knowledge and skill gaps that often hinder success in first-year university mathematics. Such a system would not only provide remedial and enrichment opportunities for existing undergraduates but also proactively equip secondary students with the foundational understanding and learning strategies necessary for navigating the more abstract and formal nature of tertiary mathematics [197].

Specifically, two strategies suggest themselves:

- Symbolic engine integration, that is, the incorporation of a computational algebra system such as *Mathematica* or *Maxima* with strong symbolic computation and the ability to assess not only syntactic forms but deep equivalences across multiple mathematical domains. This allows the system to evaluate whether $\frac{x^2-1}{x-1}$ and $x + 1$ are equivalent under the condition $x \neq 1$, or whether two different forms of a differential equation solution represent the same general solution. Beyond checking correctness, such an engine enables fine-grained tracing of student steps through simplification, expansion, factoring, and other transformations, opening new possibilities for feedback and understanding.

- The incorporation of algorithmic advances to enhance conceptual modelling and pedagogical effectiveness. Drawing on perturbation theory [35], conceptual modelling identifies structural errors that correspond to common misconceptions, such as sign errors in distribution or misapplication of inverse operations. It also introduces symbolic equivalence tracing, in which student inputs are matched against sets of equivalent forms derived through known transformation rules. This enables the system to provide targeted feedback not just about correctness but about the student's thinking process. Unlike rule-based systems that merely detect the presence of an error, such a system would evaluate the mathematical intent and guide learners accordingly. The use of symbolic feedback scaffolding would ensure that students receive tailored prompts reflecting the structure of their work, helping to build both procedural fluency and conceptual insight.

The question which now arises is how to implement these strategies effectively in an ITS designed to assist students in the transition from the secondary to the tertiary mathematics education environment.

Chapter 3

Theoretical Framework

3.1 Overview

In this chapter we introduce the research question to be addressed in this dissertation, and consider the theoretical framework within which our investigation will be undertaken. In Section 3.2 we draw on the insights from Chapter 2 to frame the central question that will be addressed, namely whether a suitably designed ITS can help to address some of the pedagogical challenges in mathematics education posed by the transition from the secondary education sphere to the tertiary sphere. We then examine in more detail in Section 3.3 the opportunities identified in Chapter 2, and consider the theories which underpin the strategies identified in Section 2.7 as well as how these strategies might be implemented in an ITS targeted at the secondary/tertiary transition. In particular, in Section 3.4 we consider the learning theories that principally inform the system design and in Section 3.5 we consider issues surrounding the integration of the symbolic engine *Mathematica*. In Section 3.6 we examine some issues involved in enhancing pedagogical effectiveness—in particular, student modelling—and in Section 3.7 we will focus on questions arising in the effective implementation of the system. Some concluding remarks are offered in Section 3.8.

3.2 The Research Question

As reviewed in Chapter 2, the transition from secondary to tertiary mathematics education presents substantial pedagogical challenges. These include significant gaps in conceptual understanding, increased abstraction, and a shift from procedural to formal mathematical reasoning. Despite numerous reform efforts, including curricular realignments and foundational bridging programs, many students continue to struggle with mathematical readiness, particularly in their first year of university [16, 50, 204].

A core theme emerging from the literature is that these challenges are not simply epistemological gaps but involve deeper cognitive and affective dimensions of learning. For instance, constructivist critiques [189, 82] argue that traditional pedagogical models often fail to support reflective abstraction and conceptual autonomy. Moreover, conventional instruction rarely provides timely, adaptive feedback, nor does it afford students opportunities for symbolic experimentation and productive failure in a low-stakes environment.

These concerns give rise to the central question addressed in this dissertation:

> ***Can a suitably designed ITS, based in sound pedagogical theory and equipped with a symbolic computation engine, help address the conceptual and cognitive challenges that students face in the transition from secondary to tertiary mathematics education?***

This question is not purely theoretical. As demonstrated in Chapter 2, existing ITSs such as ALEKS, ActiveMath, and SQL-Tutor exhibit certain limitations, particularly in domains requiring symbolic manipulation and abstraction. Chapter 2 also identified a gap in the current landscape: the lack of ITSs tailored to support the mathematical transition space with explicit integration of learning theories and domain-relevant symbolic reasoning engines.

Given these insights, and following on from the overarching research question addressed above:

1. How can symbolic reasoning engines enhance student models in ITSs for tertiary mathematics?
2. How can perturbation be used diagnostically to inform adaptive feedback in conceptual learning?
3. In what ways can design-based research support the iterative development and evaluation of ITSs aligned with constructivist pedagogy?

3.2.1 *Tutoria* for Transition Mathematics

In this thesis, we have designed, implemented, and evaluated a new ITS for mathematics education, which we refer to as *Tutoria*. In the following chapters we will describe the design and implementation of *Tutoria*, an ITS designed to operate specifically in the transition zone between secondary and tertiary mathematics. This space is marked by significant epistemological shifts: students move from concrete, procedural applications to abstract, structural reasoning. Many struggle with this transition, finding the new modes of representation and reasoning unfamiliar and cognitively demanding. *Tutoria* addresses this challenge by employing a model that prioritizes symbolic logic and expression equivalence, giving students the freedom to work in multiple valid forms while still receiving rigorous feedback.

The system supports tasks that are intentionally underspecified in terms of required form, inviting students to engage with the underlying structure of mathematical relationships rather than just plugging into formulas. For instance, students solving a linear equation may submit the result in slope-intercept form, standard form, or as an implicit relation, with the system checking all against a symbolic model of correctness. Using *Mathematica*, transformations are computed in real-time, and students' reasoning paths are logged for instructor analysis. The Student Model evolves dynamically, recording patterns of equivalence awareness, procedural fluency, and strategy use. These profiles inform adaptive feedback and personalized intervention within a session. From session to session progress is mapped showing the user the next point of entry.

As *Tutoria* the system is composed of three interconnected models: a Domain Model, a Student Model, and a Teaching Model. The Domain Model adopts a spiral curriculum structure, where key mathematical concepts are revisited at increasing levels of abstraction. This design is informed by Bruner's theory of the spiral curriculum [39] and aligns with constructivist views of knowledge development such as those of Piaget [146]. The student model combines model-tracing [10] and constraint-based approaches [125], enabling the system to identify both procedural errors and conceptual misunderstandings. This hybrid strategy draws on the strengths of existing ITS implementations such as the Cognitive Tutor [107] and SQL-Tutor [123]. Finally, the teaching model is rule-based, using predefined instructional logic to determine appropriate interventions based on the learner's actions and inferred understanding [188]. These architectural decisions form the core of *Tutoria* and are elaborated further in Sections 5.3, 5.5.6, and 5.3.2.

3.3 Opportunities Identified in the Literature Review

Several opportunities arise from the shortcomings documented in Chapter 2. First, there is a clear pedagogical need for systems capable of recognising algebraic equivalence across diverse symbolic forms. The rigidity of current ITS input parsing mechanisms and their reliance on surface-structure representations limits their ability to assess conceptual understanding. Systems like ALEKS and ActiveMath often fail to identify valid mathematical equivalences, thereby misclassifying correct but non-canonical responses [118, 125, 121].

Second, current ITSs struggle to accommodate multiple valid solution strategies. Model-tracing systems like Cognitive Tutor rely heavily on expert-defined rules and pathways, leading to limited feedback when students employ novel but valid approaches [11]. This restricts learner agency and may inhibit the development of flexible problem-solving skills essential in tertiary mathematics.

Third, student feedback in existing systems is often procedural rather than conceptual. This creates a missed opportunity for developing metacognitive skills, reflective abstraction, and deeper understanding. A well-designed ITS could instead scaffold conceptual insight through adaptive, symbolic feedback informed by the student's actual mathematical intent.

These opportunities suggest the need for a new kind of ITS—one that integrates symbolic equivalence checking, allows flexible input and solution strategies, and delivers feedback informed by cognitive and conceptual modelling.

Secondary-Tertiary Transition Disconnect Students transitioning from secondary to tertiary mathematics often experience a shift from pragmatic, intuitive secondary mathematics to formal, abstract tertiary mathematics, which increases cognitive load [64]. In secondary school, mathematical understanding is largely pragmatic and intuitive, relying on procedural fluency, concrete examples, and contextual reasoning. However, at the university level, mathematics becomes increasingly formal, abstract, and proof-oriented. This shift introduces new demands on learners, including the need to engage in symbolic manipulation, deductive reasoning, and the construction and interpretation of formal definitions and theorems. The departure from familiar, application-based tasks to more theoretical and generalized concepts can be disorienting and challenging. Consequently, this transition places a heightened cognitive load on students as they must not only acquire new content knowledge but also adapt to an entirely different epistemological framework for understanding mathematics. [87]

Foundational concepts such as limits are frequently misunderstood by students [59, 175], in part due to deep-seated epistemological[1] conflicts between the informal, intuitive meanings derived from everyday language and the precise, formal definitions used in mathemat-

[1]The term epistemological conflict is used here deliberately, rather than attributing student difficulties merely to semantic confusion. As detailed in Cornu [59] and Swinyard [175], these misunderstandings reflect not just misinterpretations of terminology, but deeper tensions between intuitive, experience-based reasoning and the formal, abstract modes of knowing demanded by calculus. This framing aligns with constructivist perspectives on conceptual change, where shifts in meaning are inseparable from shifts in epistemic framing.

ics. Students often approach terms like "approaches," "gets closer", or "infinitely small" with preconceived notions rooted in common-sense reasoning, which can conflict with the rigorous epsilon-delta definitions presented in formal instruction.

Pedagogical Limitations Traditional models of tertiary mathematics education continue to be predominantly lecturer-centred, [96] emphasizing the transmission of information from instructor to student, rather than fostering active engagement with the material. In such environments, the instructor is viewed as the primary authority and knowledge-holder, while students are expected to absorb content through lectures, note-taking, and replication of demonstrated procedures. This model often leaves little room for exploratory learning, peer discussion, or reflective thinking—elements that are central to constructivist pedagogies [197]. As a result, students may struggle to develop deep conceptual understanding or to transfer knowledge to novel contexts, which are key goals of meaningful mathematics education.

Assessment structures in many tertiary mathematics programs often fail to promote deep, meaningful learning. Instead of encouraging students to engage critically with underlying concepts, these systems tend to reward surface-level performance, such as procedural accuracy and rote memorisation. Aggregated marks—comprising scores from quizzes, assignments, and exams—are commonly used as proxies for understanding, yet they frequently obscure whether students have developed genuine conceptual insight. This reliance on summative grading can discourage risk-taking, curiosity, and reflective thinking, all of which are essential for deep learning and long-term mathematical competence [17].

In many university mathematics courses, the design of assessment frameworks does not adequately encourage students to pursue deep learning strategies. Rather than fostering a sustained engagement with fundamental ideas, assessments are often structured around the accumulation of marks through fragmented tasks, such as isolated problem sets and time-pressured exams. As a result, students may prioritise score maximisation over the development of coherent, transferable conceptual understanding. This system reduces learning to a performance metric, where aggregated numerical results substitute for evidence of meaningful cognitive growth [17].

A persistent limitation in traditional mathematics instruction is the lack of responsiveness to individual student needs, learning trajectories, and misconceptions. In many lecture-based environments, the pace and content of instruction are predetermined, offering little flexibility to adapt to the diverse conceptual difficulties students may encounter. As a result, students who struggle with foundational misunderstandings often fall behind, without timely intervention or opportunities for remediation. This one-size-fits-all approach overlooks the importance of formative assessment and real-time feedback, both of which are crucial for addressing cognitive obstacles and supporting personalised learning pathways [160].

Technology under-utilisation The application [58] of ITSs has been predominantly concentrated in middle and secondary education contexts, where platforms [150] such as the Progressive Achievement Tests (PAT) and ASSISTments have been successfully implemented to support foundational skill development and formative assessment. These systems are often tailored to scaffold procedural fluency, deliver immediate feedback, and diagnose student misconceptions within well-defined curricular domains. However, despite their proven effectiveness at these educational levels, there has been comparatively little extension of ITS technologies into the tertiary sector, particularly within the critical transition phase from high school to university mathematics. This gap represents a missed opportunity to support students grappling with the increased abstraction and epistemological demands of higher education [58, 150].

The Pathway to University Mathematics (PWYM)

project offers valuable diagnostic assessment tools that help identify student readiness and highlight areas of mathematical weakness prior to university entry. While this initiative contributes significantly to early benchmarking and awareness of learning gaps, its design is primarily evaluative in nature. It does not incorporate interactive learning pathways, adaptive feedback, or structured opportunities for conceptual scaffolding that might support students in addressing their identified deficiencies. As such, although the PWYM project plays a critical role in assessment, it falls short of functioning as a comprehensive learning environment that actively fosters student growth and conceptual development, see Appendix H.

While Mathematica® offers powerful capabilities for symbolic computation and advanced mathematical modelling, it presents notable usability barriers for novice or less experienced learners. The steep learning curve associated with its syntax, interface, and functional structure can discourage exploratory engagement, particularly among students transitioning from more guided or visual learning environments. As a result, despite its computational strengths, Mathematica's complexity can limit its accessibility and pedagogical effectiveness when deployed as an ITS platform. Without substantial scaffolding or interface simplification, it risks alienating the very learners who would benefit most from automated support in mastering abstract mathematical concepts [112].

ITS Development Challenges Research in the field of ITSs has often developed in relative isolation from the broader body of established educational theory [153]. Much of the foundational work in ITS has been grounded primarily in artificial intelligence, cognitive science, and computing literature, focusing on algorithmic efficiency, knowledge representation, and system performance. As a result, important insights from educational psychology, learning sciences, and pedagogical frameworks are frequently underutilised or only superficially incorporated. This disciplinary disconnect can limit the pedagogical robustness of ITS designs, reducing their potential to fully support the complex, socio-cognitive processes involved in human learning.

While significant technical progress has been made in areas such as knowledge tracing, adaptive feedback, and user modelling, many ITS initiatives have historically overlooked foundational principles from educational psychology, constructivist learning theories, and instructional design. This tendency to ground ITS development primarily in AI and computational frameworks can result in systems that are technologically sophisticated but pedagogically limited, missing critical opportunities to align more closely with how students actually learn and develop understanding [153].

Many ITSs continue to be built upon early behaviourist models of learning, [26, 165] which focus heavily on observable inputs and outputs, such as correct responses and error patterns. While these approaches allow for structured, measurable interactions, they often neglect the internal cognitive processes that underlie meaningful learning, such as metacognition, self-regulation, and reflective thinking. As a result, these systems may effectively reinforce procedural skills but fail to foster deeper awareness of learning strategies, problem-solving heuristics, or the ability to monitor and adapt one's own understanding—critical components of advanced expertise development [26, 165].

In many ITSs and educational models, students' misconceptions are frequently treated as simple deficits—errors to be corrected—rather than being recognised as valuable indicators of a student's evolving understanding. Traditional approaches often view incorrect answers solely as failures relative to expert knowledge, thereby missing the opportunity to leverage misconceptions as stepping stones toward deeper learning. More advanced perspectives, such as the perturbation model, suggest that student errors reflect underlying cognitive structures that, when properly addressed, can facilitate conceptual growth and refinement. However, many existing systems still default to deficit-based interpretations, limiting their capacity to support meaningful, constructive learning processes [41, 84].

Figure 3.1 illustrates a conceptual comparison between the traditional deficit model of student error and the more nuanced perturbation model. In the deficit model, student errors are typically treated as gaps or absences in knowledge that must be corrected, often leading to remedial instruction aimed at restoring an assumed ideal state. In contrast, the perturbation model interprets errors as productive deviations that reveal underlying reasoning processes. Rather than simply correcting mistakes, this approach seeks to engage with the logic behind them, using error as a diagnostic entry point for conceptual growth. This distinction supports a shift from instructional models that prioritise correctness toward those that value insight into student thinking — a key theme in this thesis.

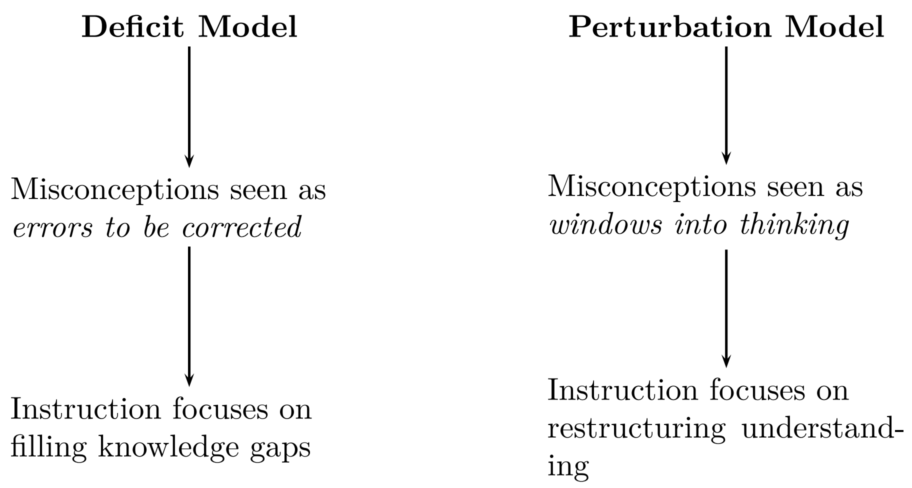


Fig. 3.1 Contrasting views of student error: deficit correction versus diagnostic engagement.

3.4 Theoretical Foundations of System Design

This section outlines the pedagogical theories that have informed the design of the *Tutoria* system, with particular focus on the Domain and Teaching Models. As noted in Section 3.2, the Domain Model adopts a spiral curriculum design inspired by Bruner [39], while the Teaching Model is structured as a rule-based system that delivers adaptive interventions. These choices reflect a constructivist stance on learning [146], where repeated exposure to deepening layers of abstraction is paired with timely and targeted instructional support.

The pedagogical framework for Tutoria is rooted in several intersecting theories. First, constructivist epistemologies [189, 142] argue that learners actively construct knowledge through experience, exploration, and self-reflection. This perspective critiques the computer-as-mind metaphor for its failure to model reflective thought [82]. Constructivism calls for environments that support discovery, symbolic experimentation, and self-directed learning.

Second, Sfard's dual-process theory [161] and Tall's procept theory [176] emphasise the transition from process to object thinking. These frameworks are critical in understanding the difficulties students face when moving from secondary algorithmic procedures to tertiary-level structural reasoning. Designing systems that allow learners to toggle between these modes supports the development of flexible mathematical cognition.

Third, theories of feedback and formative assessment [28, 156] reinforce the importance of providing students with timely, relevant, and adaptive feedback. This is particularly important for correcting misconceptions, supporting metacognitive reflection, and encouraging sustained engagement.

Together, these theories suggest that an effective ITS for transition mathematics should:

- allow for symbolic and procedural representations,
- support equivalence checking and transformation,
- respond to misconceptions with conceptual feedback,
- and promote student reflection and strategy exploration.

The transition from secondary to tertiary mathematics is a critical point at which many students encounter failure, disengagement, or a significant decline in confidence. Research shows that tertiary mathematics requires abstraction, proof-based reasoning, and epistemological shifts that are not sufficiently scaffolded by prior educational experiences [64, 197, 176]. Despite diagnostic tools like the PWYM platform, current interventions do not provide tailored, responsive, or adaptive feedback that meets individual student needs.

An ITS, if appropriately designed, could address these shortcomings. ITSs are capable of delivering individualised instruction, tracking misconceptions, and adapting content in real-time—functions not currently performed by institutional assessments or traditional teaching methods. Developing such a system for the tertiary mathematics context fills an urgent pedagogical need and provides a new avenue to improve student retention, learning outcomes, and transition readiness.

This study is informed by a synthesis of cognitive and educational theories and their application in intelligent systems:

- **Piagetian Reflective Abstraction and APOS Theory** [71, 18] frame how students progress through stages of understanding mathematical concepts as *Action* → *Process* → *Object* → *Schema*.

- **Sfard's Dual Nature of Mathematical Conceptions** [161] offers a sociocognitive perspective, balancing procedural fluency with conceptual understanding.
- **Constraint-Based and Model-Tracing Architectures** [123, 11] provide proven ITS frameworks, offering contrasting approaches to Student Modelling—either through violation of constraints or tracing procedural paths.
- **Constructivist Learning Theory** [193, 84] supports the notion that students actively build and revise internal representations, which can be facilitated through exploration-based ITS environments.

The incorporation of learning theories into the system is foundational to its pedagogical strategy, influencing the Domain Model, Teaching Model, and Student Model. Dabbagh [63] emphasizes the importance of a theory-based design framework in e-learning, highlighting how pedagogical models derived from learning theories inform instructional strategies and the development of learning technologies. Similarly, Hammad et al. [86] review various learning theories underpinning technology-enhanced learning artifacts, underscoring their role in shaping effective educational tools.

Domain Model The Domain Model structures mathematical content in a spiral configuration, where core concepts are revisited at increasing levels of abstraction and generality. This approach, inspired by Bruner's spiral curriculum [39], is adapted here to a mathematical context, allowing for progressive deepening of understanding. Constructivist learning theories, particularly those of Piaget and Vygotsky [44], inform the design logic, where new conceptual layers build upon previously internalised ideas. ZPD-informed scaling guides the introduction of more challenging tasks just beyond the learner's current level. This progression is supported by the perturbation model of error, which treats student misconceptions not as deficits to be corrected, but as diagnostic opportunities to inform task selection and scaffold advancement. This alignment with contemporary curriculum frameworks [97] ensures that conceptual development within the Domain Model remains both responsive and pedagogically principled.

The system draws from constraint-based [124] and model-tracing frameworks [11], offering stepwise feedback and hinting based on student input. The Teaching Model also leverages the idea of *perturbation* [188], which models student misconceptions explicitly, rather than merely identifying deviation from expert solutions.

Student model The Student Model, that is to say the snapshot, as discussed generally in Section 2.5.2, found in the stored database records (see Section 5.5.3) reflects a hybrid design that integrates overlay modeling with perturbation-based diagnostics [55, 188]. Student responses are compared with the correct answer and then, based on the student response, a decision is made dynamically as to the next point approached in the curriculum. In so doing, the overlay component tracks which concepts a student has mastered by comparing their responses to an expert model, updating dynamically, as new evidence is gathered. Perturbation modeling complements this by identifying patterns in student errors that reflect underlying misconceptions, rather than treating mistakes as isolated events. If a student has presented a correct equivalence, which falls short of complete correctness, then in the mode selected to perform *perturbation*' , *Tutoria* will load a "next remedial step" question matching the dynamically discovered need to develop completeness. This allows the system to infer likely misunderstandings and select targeted interventions. Together, support in adaptive task sequencing and responsive feedback is achieved, helping to guide learners within their Zone of Proximal Development while maintaining a clear picture of conceptual progress over time. Implementation can be found in Section 5.3.2.

Cognitive load consideration The design, by making the detection and implementation of student misunderstandings transparent, also addresses cognitive load theory [173], with efforts to reduce extraneous load through intuitive visual representations, incremental hinting, and progressive complexity.

Holistic cognitive strategy Constructivists argue that cognition cannot be reduced to black-box interactions [44]. Therefore, the system prompts metacognitive engagement by encouraging students to reflect on their methods and guides them to explore alternative solution paths. This guiding is practice of lower knowledge and skills.

3.5 Rationale for Integrating Mathematica

In this section, we discuss the rationale behind the technical design of the Student Model within *Tutoria*. The system adopts a hybrid model that draws on both model-tracing and constraint-based approaches, enabling it to detect procedural errors and diagnose conceptual misunderstandings. This combined approach builds on established methods from the Cognitive Tutor [10, 107] and SQL-Tutor [123], allowing the system to provide responsive and granular feedback that supports student learning.

A key differentiator of the proposed ITS is the integration of `Mathematica`® as its symbolic computation engine. As outlined in Chapter 2, Mathematica supports deep symbolic reasoning and can evaluate expressions for structural and algebraic equivalence under specified conditions. This enables the system to recognise, for example, that $\frac{x^2-1}{x-1}$ and $x+1$ are equivalent for $x \neq 1$, or that $x^2 - 4$ and $(x-2)(x+2)$ are interchangeable.

This capacity allows the ITS to provide more nuanced, semantically-aware feedback. It also opens the possibility for students to explore symbolic transformations freely, with the system interpreting their actions in terms of mathematical intent rather than form. This contrasts sharply with systems that rely on template-based parsing or surface matching, which often fail in higher-order domains like calculus or algebraic proofs.

Furthermore, Mathematica allows detailed step tracing. It can record and evaluate transformation chains, providing a basis for diagnosing strategies, identifying misconceptions, and generating tailored feedback based on student trajectories.

Symbolic engines and computational tools Tools such as Mathematica® and Maxima provide powerful symbolic reasoning capabilities that are particularly well-suited for domains characterised by procedural knowledge, such as calculus, algebra, and symbolic manipulation. These systems are capable of performing complex algebraic operations, solving equations, computing integrals and derivatives, and simplifying expressions, often with a high degree of accuracy and efficiency. Their strength lies in the ability to handle formal mathematical processes in a consistent, rigorous manner, making them highly valuable in supporting student learning in procedural areas. However, while they excel in algorithmic manipulation, their effectiveness depends heavily on the user's ability to interpret and understand the underlying mathematical concepts [112].

Tools such as advanced symbolic computation engines and dynamic mathematics platforms have the potential to power highly dynamic, high-fidelity ITS environments specifically tailored for advanced mathematics instruction. Their ability to perform exact, rigorous symbolic reasoning, alongside numerical approximation and visualisation, enables the creation of learning experiences that closely mirror the depth and complexity of authentic mathematical practice. By integrating these tools into ITS frameworks, it becomes possible to design systems that not only check correctness but also offer multiple representations, guide problem-solving processes, and adapt dynamically to student input. Such environments can support learners in developing both procedural fluency and conceptual understanding at higher levels of mathematical abstraction [29].

The choice of *Mathematica* as the core computational engine in this ITS is motivated by three primary factors:

Symbolic and numerical computation Unlike numerical engines, computational engines like Mathematica and Maxima support both symbolic and numerical computation, which is essential in mathematics education. Mathematica can verify equivalences through symbolic transformation, ensuring accurate assessment of student responses across a variety of representations.

Assessment of equivalence in multiple representations ITS systems must recognise that students may provide correct answers that differ syntactically but are algebraically sound. As noted in early ITS implementations like LISPITS [55], recognizing semantically equivalent expressions is a core requirement. Mathematica allows for this via commands such as `Simplify`, `FullSimplify`, and `Assuming`, enabling robust validation. The actual code (with return string processing redacted) follows:

```
<msp:evaluate>
  answers = FullSimplify[MSPToExpression[ $$answer, MathMLForm]];
  answer = ToString[answers];
  questions = FullSimplify[MSPToExpression[ $$question, MathMLForm]];
  question = ToString[questions];
  result = ToString[answers === questions];
  If[ result === "True",
    format return string and set truth state...
</msp:evaluate>
```

Cloud integration and accessibility The distributed accessibility of Mathematica through Wolfram Cloud and `WolframScript` enables integration into LMS platforms and web-based systems. Wolfram's educational technology platform [196], Quezzio, exemplifies this integration. Quezzio offers seamless compatibility with existing LMS platforms such as Canvas, Blackboard, D2L, and Moodle. It includes tools for question authoring, auto-grading powered by Mathematica and Wolfram|Alpha, and analytics, all accessible through a web-based interface. This allows scalable, cloud-based deployment, addressing the need for an accessible, high-performance computational backend for symbolic reasoning.

3.6 Enhancing Pedagogical Effectiveness

Effective Student Modelling lies at the heart of adaptive instruction. As surveyed in Chapter 2, traditional approaches include overlays [43], constraint-based models [123], buggy models [35], and Bayesian networks [120]. Each brings strengths and weaknesses:

- Overlay models are simple but fail to capture misconceptions.
- Buggy models diagnose errors but are costly to construct.
- Model tracing is precise but inflexible.
- Constraint-based models offer generality but limited insight into thought process.
- Bayesian approaches capture uncertainty and learning trajectories but are computationally intensive.

This system proposes a hybrid approach. The symbolic engine provides deterministic feedback on structural correctness, while an overlay approach models evolving conceptual understanding. For example, repeated success on certain transformations increases belief in mastery, while errors trigger perturbation logic to detect faulty generalisations. This dual-layered model allows both symbolic and cognitive diagnosis, addressing student knowledge with greater precision.

Alignment with Educational Reform The growing momentum in transition pedagogy, reflected in initiatives such as curriculum reforms, the development of diagnostic assessment tools, and the increasing focus on first-year experience programs, creates a supportive environment for the implementation of AI-assisted mathematics learning tools. Educational institutions are increasingly recognising the need to provide targeted support during the critical transition from secondary to tertiary study, aiming to address the academic, cognitive, and emotional challenges that students face. This shift aligns well with the potential of AI-driven technologies to deliver personalised learning pathways, identify conceptual weaknesses early, and offer adaptive feedback tailored to individual student needs. As a result, there is a strong foundation upon which intelligent, data-informed mathematics education systems can be successfully introduced and integrated into existing support structures [187, 111].

There is a significant opportunity to bridge the fields of computing, artificial intelligence, and education in order to address long-standing challenges associated with the transition from secondary to tertiary mathematics education. Historically, efforts to improve transition outcomes have been constrained by limited resources, inconsistent pedagogical approaches, and difficulties in personalising support at scale. However, advances in AI, particularly in ITSs, knowledge modelling, and adaptive feedback mechanisms, offer new ways to design responsive, learner-centred environments. By integrating insights from educational theory with technological innovations, it becomes possible to create systems that not only diagnose and remediate gaps in understanding but also actively foster the development of critical mathematical thinking skills needed for success at the university level [192].

3.7 Implementation Considerations

While the design vision is pedagogically ambitious, several implementation challenges must be addressed:

Interface Design The system must support flexible, open-ended input. Unlike multiple-choice or template boxes, the input interface should parse free-form algebraic expressions. This requires integration with Mathematica's parsing and simplification modules.

Equivalence Recognition Symbolic equivalence must be treated as a first-class construct. The system should evaluate equivalence under domain-specific constraints (e.g., defined variable ranges, assumptions), and provide feedback that recognises valid alternatives.

Feedback Granularity Rather than binary correctness, the system should detect and reward partial progress. For instance, factoring only part of a quadratic expression should be acknowledged, with scaffolded prompts guiding the learner.

Data Infrastructure A persistent Student Model must track knowledge acquisition over time. The system should log attempts, transformation sequences, and feedback responses to build a dynamic profile.

Scalability and Accessibility Cloud-based deployment with LMS integration (via LTI or API protocols) can ensure access across platforms and devices. Mathematica's cloud functionality supports scalable symbolic evaluation.

Design and implementation A key question in the development of ITSs for tertiary transition mathematics is identifying the core features that would make such a system both pedagogically effective and technologically feasible. To genuinely support learners, an effective ITS would need to incorporate adaptive Student Modelling, capable of diagnosing conceptual understanding and misconceptions in real time. It should offer personalised feedback pathways that scaffold both procedural fluency and deeper conceptual reasoning, rather than focusing solely on correctness. Additionally, the system should embed elements of metacognitive support, encouraging students to reflect on their problem-solving strategies and learning processes. From a technological standpoint, the platform must leverage symbolic reasoning engines, and dynamic feedback mechanisms while maintaining a user interface that is accessible and intuitive for diverse learner profiles.

Future versions may include probabilistic knowledge modelling (e.g., Bayesian networks). Balancing these pedagogical and technical demands is critical to designing an ITS that meaningfully improves the mathematics transition experience without overwhelming students or instructors.

Student Modelling A critical consideration in the development of effective ITSs for tertiary transition mathematics is determining which Student Modelling techniques—such as constraint-based modelling, overlay models, or model tracing—best capture and respond to the specific learning difficulties encountered during the transition phase. Transitional learning challenges often involve not only procedural errors but also deep-seated conceptual misunderstandings, epistemological conflicts, and difficulties in adapting to formal mathematical reasoning. Therefore, an ideal Student Model must go beyond surface-level performance tracking to diagnose underlying cognitive structures, misconceptions, and strategic thinking processes. Techniques such as constraint-based modelling, which focuses on identifying violations of domain-specific principles, overlay models, Bayesian models, which probabilistically infer knowledge states, and model tracing, which monitors sequences of cognitive actions, each offer distinct advantages. Evaluating their respective strengths and limitations in the context of transitional mathematics learning is crucial for designing ITS systems that are both diagnostically precise and pedagogically responsive. As discussed in Section 3.4, Tutoria is based on model tracing using overlay models which are deterministic and step sequentially through student responses. Tutoria, due to symbolic processing, can detect *perturbation* and offer an additional feature to model tracing, dynamic completeness and correctness evaluation.

An important design question for ITSs supporting tertiary transition mathematics is how the system can effectively detect and remediate epistemological misconceptions, particularly those that are commonly observed in early university calculus and algebra. These misconceptions often arise not merely from procedural errors but from fundamental misunderstandings about the nature of mathematical objects, processes, and formal reasoning. For instance, students may hold incorrect beliefs about the concept of limits, infinity, continuity, or the structure of functions and algebraic expressions. To address such issues, the system must be able to diagnose deep-seated conceptual conflicts rather than only surface-level mistakes. Implementation can be found in Section 5.3.2.

Example: Detecting and remediating misconceptions about limits In 1991, Cornu observes in *Limits* students present with:

> "... a conceptual misunderstanding regarding the definition of a limit... " [59]

A common epistemological misconception among early university calculus students is the belief that a limit at a point depends on the function's actual value at that point, rather than on the behaviour of the function as it approaches the point. To detect this misconception, the ITS could present diagnostic questions where the function is undefined at the target point but has clear limiting behaviour e.g.,

$$\lim_{x \to 2} \frac{x^2 - 4}{x - 2}.$$

If a student incorrectly asserts that the limit does not exist because the function is undefined at $x = 2$, the system could infer the presence of a conceptual misunderstanding regarding the definition of a limit. Adaptive feedback would then prompt the student to explore the function's behaviour as x approaches 2 from both sides, possibly using dynamic graphing tools or symbolic manipulation prompts [177].

Remediation would involve guiding the student through multiple representations (e.g., tables of values, graphical zoom-in, and symbolic factorisation) and reflective questions, such as "What happens to $f(x)$ as x gets very close to 2, even though $f(2)$ itself is not defined?" This approach encourages conceptual restructuring, helping students distinguish between function evaluation and limit processes.

Pedagogical Effectiveness An important evaluative question concerns the extent to which a newly developed ITS improves student outcomes compared to existing self-assessment platforms such as the Pathways to University Mathematics (PWYM) and similar systems. While PWYM provides valuable diagnostic information and supports student self-awareness of mathematical readiness, it primarily offers static feedback without sustained instructional adaptation or personalised remediation. In contrast, an ITS has the potential to deliver dynamic, adaptive learning experiences that respond to individual misconceptions, knowledge gaps and learning trajectories in real time. Measuring improvement would therefore require rigorous comparisons of student performance, conceptual understanding, and persistence rates across both systems, taking into account factors such as depth of learning, rate of progression, and student confidence. Understanding the relative impact of ITS versus traditional diagnostic platforms is essential for justifying the adoption of more sophisticated, resource-intensive intelligent learning technologies.

A critical research question concerns the student experience of interacting with ITS and how this experience influences key affective and cognitive factors such as engagement, self-efficacy, and perceived preparedness for tertiary mathematics. Beyond measuring academic outcomes, it is important to explore how students perceive the usability, responsiveness, and supportiveness of the ITS environment. A system that fosters positive engagement can encourage sustained interaction with challenging material, while improvements in self-efficacy —– students' beliefs in their capacity to succeed in mathematics —– may have a lasting impact on academic persistence and performance. Additionally, students' perceptions of their own readiness for university-level mathematics, shaped by their ITS learning experiences, are likely to influence their transition success. Understanding these dimensions is essential for evaluating the broader effectiveness and acceptance of ITS technologies in supporting transition-stage learners.

Institutional integration An important practical consideration is how an ITS can be effectively deployed within existing university diagnostic and support frameworks.

Integration with current institutional infrastructure would require careful alignment of the ITS's diagnostic capabilities with the competencies assessed in established readiness surveys, ensuring coherence between baseline assessment and ongoing personalised support. Ideally, the ITS would complement existing diagnostics by providing not only static reports but also dynamic pathways for remediation and extension, allowing students to transition seamlessly from initial evaluation to targeted learning interventions. Additionally, integration strategies must address logistical factors such as accessibility through institutional learning management systems (LMS), alignment with curriculum structures, staff training, and data reporting mechanisms that respect privacy regulations while enabling actionable insights. Thoughtful deployment would maximise the impact of the ITS by embedding it within the broader ecology of student support services rather than treating it as an isolated add-on.

3.8 Concluding Remarks

This chapter has outlined the theoretical foundation for an ITS designed to address the epistemological, symbolic, and cognitive challenges that characterise the transition from secondary to tertiary mathematics. It has framed the central research question, surveyed the limitations of existing ITS designs, and articulated opportunities for innovation.

By drawing on constructivist theory, dual-process cognition, formative assessment principles, and modern advances in symbolic computation and Student Modelling, this framework proposes a hybrid system that is both pedagogically rigorous and technically feasible. The next chapters will describe how this theoretical design is instantiated in practice, including the system architecture, components, and authoring model.

The proposed system, Tutoria, is envisioned as a transformative platform that aims to reshape the way students engage with tertiary mathematics content *in the transition from secondary education*. At the heart of this innovation is an intelligent tutoring framework capable of dynamically adjusting instructional pathways in response to real-time learner interaction. As students engage with conceptual explanations and procedural tasks, the system will simultaneously present interactive activities designed to assess their current understanding. Based on their responses, the system will adaptively redirect learners toward remedial or extended material, leveraging established models of adaptive instruction [56, 188]. This capacity to make intelligent decisions mid-interaction marks a shift from traditional static e-learning modules to an interactive, context-sensitive tutoring experience.

A key aspect of this project lies in its architectural design, particularly the integration of symbolic computation through *Mathematica*®. Unlike many existing web-based ITS platforms, which rely on client-server communication for validation and redirection, Tutoria will utilise an embedded Mathematica kernel hosted on an Apache-Tomcat server. This configuration enables asynchronous symbolic evaluation of user input—such as algebraic expressions, equations, or symbolic reasoning steps—without requiring full-page reloads or interruption to the user flow. While this may appear similar to asynchronous JavaScript calls used in standard web development (e.g., AJAX), the distinction is crucial: the symbolic engine serves not merely as a data handler but as a real-time reasoning partner, capable of evaluating mathematical equivalence, simplification, and validity against the Student Model [93, 203].

In doing so, the system is intended to go beyond traditional models of interaction by facilitating what could be called an *epistemic feedback loop* [162], wherein each learner response is not only recorded but interpreted mathematically and pedagogically. This enables contextually appropriate feedback, remediation, or enrichment to be offered, consistent with the learner's evolving conceptual model [108]. Furthermore, the integration of symbolic computation for assessment reduces dependence on rigid pattern-matching or finite databases of correct answers, which often limit the scope of formative feedback in existing ITS environments.

Chapter 4

Methodology

4.1 Overview

This chapter outlines the research design and methodological approach adopted to address the research questions. The study integrates educational theory, symbolic computation, and empirical validation to develop and evaluate an ITS for students transitioning into tertiary mathematics.

In Section 4.2 the discussion applies both to the development of the ITS software overall and, in particular, the methodologies chosen to implement the Student Model within the system. In Section 4.3 we consider the pilot deployment and data collection phase, including ethics approval for the pilot program, the recruitment of participants and data collection. In Section 4.4 the exposition of the practical outputs of the DBR cycle are in relief. Some concluding remarks, including comments on the iterative refinement phase, are offered in Section 4.5.

The implementation methodology employed an Agile software development approach, which is particularly well-suited to educational technology projects due to its emphasis on iterative cycles, rapid prototyping, and continuous stakeholder feedback [25]. Agile [89] development prioritises the delivery of working software in short, adaptive sprints, allowing for flexible responses to evolving user requirements and pedagogical insights—an approach consistent with the Design-Based Research (DBR) paradigm [190, 65, 22, 65], and according to Anderson [12], a paradigm that has often been employed in the learning disciplines in recent decades. Within this framework, features were developed incrementally, tested with pilot-like users — mostly the author, and refined through repeated evaluation cycles. Once satisfied that the prototype system was usable, the empirical component could begin as a pilot study proper. This methodology was instrumental in managing the complexity of integrating symbolic computation tools, such as `Mathematica`®, and ensuring that the system aligned with both technical constraints and educational goals.

Following identification of the issue to be investigated and a review of relevant literature, the phases of this methodology involve the design and development of an intervention, deployment of a pilot intervention and data collection, and evaluation and analysis with the process then being repeated as necessary in an iterative refinement cycle.

DBR as applied in this project will consist of four interdependent phases:

1. Design and development of an ITS to provide targeted support to students during the transition from secondary to tertiary mathematics study;

2. Pilot deployment of the system and data collection;

3. Evaluation and analysis;

4. Iterative refinement.

4.2 Design and Development

4.2.1 System

This section outlines the methodological principles underpinning the design and development of the pilot system, guided by a Design-Based Research (DBR) approach and implemented using Agile development cycles. It is structured in two parts: first, a technical account of system development, followed by the pedagogical and modelling frameworks that inform its instructional design.

Methodological Framework The system design was guided by DBR principles, embedding development within authentic educational settings and enabling iterative refinement through stakeholder engagement and empirical evaluation. The project was aligned with UTS's transition mathematics support programs, specifically targeting the interface between secondary and tertiary mathematical readiness. The development process proceeded through three key stages:

- **Initial Design Phase:** Drawing upon findings from the literature review in Chapter 2, an initial pedagogical specification and technical architecture were drafted to address the challenges faced by transitioning students.

- **Pilot Development Phase:** A minimum viable product (MVP) was constructed using Agile project management strategies, which emphasised iterative sprints, rapid prototyping. Another element, regular stakeholder feedback [25] was implemented as a self-reflective activity by the author. This approach enabled responsiveness to emergent needs and practical constraints.

- **Evaluation and Iteration:** Formative evaluation cycles informed successive design refinements, ensuring alignment with pedagogical objectives, institutional goals, and technical feasibility — again as a limited scope project the author was central in this process.

Design Constraints and Platform Considerations The methodology was shaped by several pragmatic constraints and strategic design decisions:

- **Institutional Integration:** The system was designed to mimic the UTS Mathematics Readiness Survey (MRS) and the Pathways to University Mathematics (PWYM) initiative, as these were tried and tested and also familiar, ensuring alignment with institutional diagnostic and support programs.

- **Symbolic Engine Integration:** *Mathematica*® was chosen as the core symbolic engine, offering powerful support for procedural and algebraic manipulation well beyond conventional learning management systems.

- **Web-Based Delivery:** A browser-based interface was adopted to maximise accessibility and eliminate the need for local installation, facilitating seamless student access.

- **Software Infrastructure:** Open-source platforms were employed to expedite implementation and ensure system modularity. An open source application was used for content management, while Apache and Apache-Tomcat handled server-side logic and integration.

Platform Modularity and Responsiveness The ITS was conceived as a modular platform designed to dynamically respond to student activity. Readiness assessments from the MRS inform content delivery, enabling targeted remediation based on diagnosed knowledge gaps. This alignment enhances the immediate utility of diagnostic tools and supports long-term learning trajectories. Embedded in the authoring section of the system were the choice keys mapping curriculum navigation. Chapter 6 details more on this.

Symbolic Computation Layer The integration of *Mathematica*® facilitates exact symbolic reasoning—enabling validation of mathematical equivalence across diverse student responses.

Reducing potential mismatches between algorithmic-style processing of student answers and equivalence-based evaluation was critical to avoiding false negatives—cases where students provided correct answers that differed in form or approach. Accordingly, the research methodology included an investigation into the relevance of incorporating equivalence checking as a feature within a constraint-based tutoring framework.

User Interface and Feedback The interface prioritises interactivity and accessibility and is extensible and scalable, a feature borrowed from the author's previous contribution to the PWYM national project. Future work, addressed in Chapter 8, may include real-time visualisations, structured feedback, and adaptive scaffolding as central to the user experience. This design scaffolds for that eventuality, supporting inquiry-oriented learning while also aligning with accessibility best practices.

4.2.2 Student Modelling and Pedagogical Framework

Student Modelling Approaches The pilot ITS investigates a hybrid Student Modelling approach combining model tracing [11] and elements of constraint-based modelling (CBM) [123]. Model tracing enables detailed monitoring of student problem-solving sequences, comparing them against expert strategies to diagnose procedural errors. In contrast, CBM evaluates solution states against fundamental domain constraints, accommodating multiple valid solution paths.

In the hybrid approach used in Tutoria, model tracing was fully implemented along with the ability to specify, within each domain topic, constraints that may be used by the computational engine to evaluate the validity of a student's responses. This evaluation is then used to inform the student's future pathway through the domain.

A fully developed hybrid design, using two approaches, Model Tracing and CBM, designed to enable both procedural precision and conceptual flexibility becomes future work.

Table 4.1 Comparative benefits of Model Tracing and Constraint-Based Modelling

Model Tracing	**Constraint-Based Modelling (CBM)**
- Tracks detailed step-by-step actions - Identifies specific procedural errors - Provides structured hints and corrections - Ideal for structured domains (e.g., algebra)	- Evaluates against domain principles - Accepts diverse solution strategies - Encourages conceptual reasoning - Ideal for open-ended reasoning
Synergistic Potential: Combining these approaches enhances diagnostic accuracy and supports diverse learner pathways.	

While full CBM integration was outside the scope of the pilot implementation, future iterations may extend the symbolic engine's capacity for constraint representation, particularly in tasks involving structural equivalence.

Cognitive and Pedagogical Design Instructional tasks are informed by constructivist theories, notably APOS theory [18] and Piaget's framework for reflective abstraction [145]. Activities are designed to scaffold the transition from action to object conception, encouraging learners to reorganise mental structures and develop schema-level understanding. Emphasis is placed on task generalisation, internalisation, and conceptual connection.

Diagnostic Feedback and Remediation Rather than implementing full perturbation techniques [35], the system adopts a simplified approach. When misconceptions or knowledge gaps are identified, tailored remediation offers users a change in curriculum paths. These may include re-explanations, alternative representations, or guided re-attempts. Though more sophisticated misconception tracing remains a future enhancement, the current model prioritizes actionable insights tied to curriculum expectations.

4.3 Pilot Deployment and Data Collection

4.3.1 Participants and Ethical Clearance

Participants were drawn from incoming STEMM students enrolled in first year, first stage, mathematics (calculus) subjects who required to complete a 'readiness survey', in particular the Mathematics Readiness Survey (MRS) mentioned in Chapter 2.

Ethics approval was obtained from the Faculty of Science and authorised by the Dean. Data collection occurred in January 2024 under rules outlined by the Ethics approval. Data is held in a MySql database in accordance with UTS policy [186].

To preserve participant privacy, no identifying student information was recorded. Students verified eligibility by presenting a valid student card. All data were anonymised by the system which simply allocated the user identifier with an incremented integer. No user information was stored during the pilot. Only a results table was generated from the redacted user sessions database. This decision was influenced by constraints imposed by internal IT data management protocols and heightened concerns around student data policy at UTS required no identifying information to be stored as a security measure.

4.3.2 Recruitment and Deployment Procedure

Students who successfully completed the MRS were randomly selected and invited via SMS to participate. Sessions were held in a dedicated room with a preconfigured laptop acting as a server. Participants completed a 40-minute session in two phases:

1. **Phase 1:** Students engaged with static LMS-style content followed by a short MRS-style quiz to assess understanding.
2. **Phase 2:** After a brief break, students transitioned to the interactive Tutoria environment, where instructional material was integrated with immediate response opportunities and adaptive prompts.

The goal was to simulate typical learning conditions in both LMS and Tutoria contexts while capturing performance and engagement data.

4.3.3 Theoretical Framing of the Study

While the design and deployment of Tutoria were informed primarily by cognitive theories, this study acknowledges the ongoing discourse between cognitive and sociocultural paradigms. Activity Theory, for instance, has gained traction as a successor to constructivist models, highlighting the socially situated nature of learning [161].

However, ITSs have historically been grounded in cognitive models, and this research maintains that focus. It does not propose a new educational theory to underpin ITSs, nor does it claim to resolve the tensions between paradigms. Rather, it positions Tutoria as a cognitive system that can operate within or alongside broader social learning frameworks [200].

The system's integration with the Wolfram *Mathematica*® engine connects this research to existing work on Computer Algebra Systems (CAS) and their role in mediating mathematical learning [61].

4.3.4 Evaluation and Methodology

This study adopts a primarily quantitative approach, drawing on established mixed methods principles [49, 148], although the qualitative component originally proposed was not pursued due to time constraints. The central research question is whether Tutoria demonstrates comparable or superior performance to a traditional LMS in facilitating mathematics learning.

Quantitative data in the form of scores answering questions posed on both systems, are divided into groups. Statistical analysis of the group means will help to identify whether or not group means are equal or not. The null hypothesis of the two groups will be assumed to be no difference between groups. In the case the null hypothesis is rejected, then statistically there will be a difference in group means. At that point, quantitative data will indicate there may be an improvement of one system over another.

Design In determining an appropriate research design for this study, several methodological options were considered. One such approach involves single-use studies, such as questionnaires or interviews, which aim to capture students' reflective judgements and provide insight into their experiences. Creswell [62] argues that such judgements—grounded in students' prior experiences—offer a valuable basis for evaluating the perceived merit and utility of an intervention.

Alternatively, a comparative design enables the collection of empirical data by having students engage with two distinct activities. This allows for statistical analysis of their performance or preferences. However, such designs raise potential concerns, including order effects and practice bias—where repeated exposure to similar material may inflate performance due to familiarity rather than genuine learning gains.

A blind study may mitigate such biases, allowing for more objective measurement of outcomes. Nonetheless, ethical concerns—particularly around equity and consent—have been previously raised within the UTS Faculty of Science, with some staff expressing strong opposition to blind or partially blind designs, see Appendix H.

Given these considerations, a comparative study was ultimately adopted. This approach aligned with prior institutional experience with online multiple-choice testing and accommodated differences in testing formats. The variation in task design was judged sufficient to reduce the influence of practice effects, though this limitation is acknowledged in the analysis and findings.

Possible Quantitative Methods of Data Collection and Types of Data

Quantitative Research	
Methods	Data type
Instruments (e.g., questionnaire, closed-ended interview, closed-ended observation)	Numeric Scores
Documents (e.g., census, attendance, records)	Numeric Scores

Table 4.2 Potential data collection and types

Quantitative data included scores; the system also gave time-on-task and completion rates. Qualitative data were planned to be collected through open-ended feedback prompts and Likert-style survey questions; time constraints meant this part of the plan did not proceed. Group means were statistically compared, and a null hypothesis of no difference was assumed.

Quantitative Data Collected

Quantitative Methods	
Instrument	Data type
Multiple-choice quizzes	Numeric scores
System usage logs	Time-on-task metrics

Table 4.3 Actual data collected and types

4.3.5 Limitations and Future Directions

Participants were selected from students who had successfully completed the MRS and were progressing into their initial tertiary mathematics subject. This intentional sampling introduced bias in favor of students who were already mathematically competent. While this limited the generalizability of the results, it provided a clear and focused comparison of systems under consistent content familiarity.

The long-term goal of Tutoria is to support students with weaker mathematical backgrounds—those excluded from this pilot. Future studies should incorporate these populations and employ longitudinal designs to measure Tutoria's efficacy over time.

Such research could establish not only comparative performance but also developmental trajectories and motivational outcomes. The author has a role in support, and future work will hone in on students in need of remedial intervention, such as those entering the transition space from a non-calculus background.

4.4 Evaluation and Analysis

Will Tutoria be a reasonable example of an ITS? Tutoria was subjected to a study which compared results from both Tutoria (an ITS) and PWYM (a non ITS system). The Null Hypothesis tests whether there will be a difference between groups using two different online assessment and learning systems.

- A Learning Management System, LMS, accessing HTML information and then using MRS style assessments and
- the alternative system, Tutoria.

The null hypothesis checks for no difference. In the event that this is rejected, then a statistically significant mean difference will exist post intervention.

Quantitative: To evaluate the effectiveness of the instructional intervention, learning gains were quantitatively assessed using analysis of Confidence Intervals (CI) and paired sample *t*-tests. These statistical procedures were selected to compare performance differences across distinct groups (e.g., control vs. experimental). The analysis of individual results across timepoints (pre-test MRS style vs. post-test Tutoria style) is left to further work as is

The *t*-tests were employed to detect significant differences in mean scores, particularly in contexts where normality and homogeneity of variance assumptions were met [78]. ANOVA is discussed in Section A.2.1. The analysis showed the shortcomings of a low sample size and prompted the use of Bootstrapping the sample set for further analysis.

Together, these methods provided an inferential framework for identifying the statistical significance of the null hypothesis, indicating learning gains and pedagogical impact.

Qualitative: To assess participants' perceptions of the system's utility and usability, the project plan called for qualitative feedback subjected to thematic analysis, a method well-suited for identifying, analyzing, and reporting patterns (themes) within data [31]. The time constraints of the project meant the planned qualitative data collection and its analysis must now form a component of future work. This work will enable the systematic examination of open-ended responses collected through surveys and interviews, facilitating the categorization of participant insights into coherent themes related to functionality, user experience, and pedagogical value.

4.5 Concluding Remarks

Dynamic The results obtained from the evaluation phase have informed possible refinements to the system's features, models, and user interfaces. While the outcomes were promising, limitations in the evaluation process—particularly the restricted sample size indicate areas for future development. To strengthen the generalisability and interpretive depth of the findings, future iterations of the project will prioritise transitioning to an online-anytime, from-anywhere deployment model. This will facilitate broader participant access and enable the integration of qualitative data collection to complement and contextualise the quantitative results.

Design-Based Research (DBR) is characterized by iterative, real-world testing of educational interventions in authentic settings. By engaging in successive cycles of design, implementation, analysis, and refinement, researchers can respond flexibly to emergent findings and contextual variables. This process-oriented methodology aligns closely with the need to develop robust, usable knowledge about learning and teaching while remaining responsive to practitioners' needs [65]. The iterative nature of DBR enables researchers to generate both practical and theoretical contributions through tightly coupled experimentation and observation.

Each iteration in a DBR cycle is intended to refine both the learning environment and the underlying pedagogical theory. For example, in educational technology development, early prototypes are deployed, studied, and modified based on student interaction data and stakeholder feedback [151]. This contrasts with more static experimental designs that aim to eliminate context as a variable. Instead, DBR embraces the complexity of the educational landscape, iterating in ways that aim to concisely meet this projects specifications completely [22].

Through this, the evolution of the intervention and associated theory becomes a co-constructed process with teachers and learners.

Over time, the DBR approach fosters a rich convergence between design goals and instructional insights. As each cycle progresses, researchers and practitioners develop a deeper understanding of the affordances and constraints of the learning environment. This supports not only the localized improvement of a given intervention but also informs broader educational theory and systemic change [190]. In this way, DBR's successive cycles of improvement are not merely a means to an end, but a rigorous epistemological strategy for embedding learning design within authentic contexts and theoretical frameworks.

Iterative refinement for this project is necessarily consigned to future work.

Chapter 5

System Design and Implementation

5.1 Overview

As mentioned in Section 4.2, an Agile system development approach was adopted for the implementation of the pilot ITS Tutoria. In particular, the design brief was deliberately flexible and subject to modification during the development phase. It reflects both the functional requirements and the constraints on the system design described in Section 4.2. The functional requirements of the system were guided by both the archetypal four-component structure (user interface, Domain Model, Student Model and Teaching Model) commonly used in previous ITSs. The rapid implementation of these models was achieved through the exploitation of existing third-party software platforms. During the design process, the mapping of the functional requirements onto these platforms was conceptualized in terms of 'layers', loosely inspired by the well-known Open Systems Interconnection (OSI) model [205].

In the remaining sections of this chapter we will consider each of these stages in turn. In Section 5.2 we consider the design brief in more detail. Then, in Section 5.3 we concentrate on the design of each of the four models that comprise the 'high-level' structural elements of the system, and in Sections 5.4 and 5.5 we consider how these will be mapped onto a number of 'layers', each of which plays a particular role in instantiating the four models. Finally, we offer some concluding remarks in Section 5.6,

5.2 Initial Design Brief

Overall brief: to design and develop a pilot ITS to provide targeted support to students during the transition from secondary to tertiary mathematics study. In Section 4.2.1 we identified a number of constraints to which the design would be subject, which we repeat here for convenience.

- Integration with UTS Transition Programs: The system was designed to support and enhance existing UTS initiatives for transitioning students into first-year mathematics. Compatibility with that institution's Mathematics Readiness Survey (MRS) and the Pathways to University Mathematics (PWYM) program was prioritised.
- Symbolic Engine Integration: Mathematica© was selected as the symbolic computation backend due to its advanced capabilities in procedural and algebraic manipulation, supporting tasks beyond standard LMS functionality.
- Web-Based Interface: To ensure platform accessibility and reduce onboarding complexity, the system was implemented as a browser-based application requiring no user-side installation.
- Use of Readily Available Software Libraries: The development process leveraged existing open-source tools and libraries to expedite implementation. Joomla was used for content management, while Apache-Tomcat served as the backend application server. These technologies supported modular deployment, scalability, and ease of integration into UTS's IT infrastructure.

With these in mind we can refine the overall brief: the pilot system will satisfy the functional requirements detailed in the following paragraphs.

1. Curriculum and Task Mapping Engine The system shall include a Domain Model that maps tasks to curriculum learning objectives, aligned with national and tertiary transition standards. Tasks will be annotated with metadata describing prerequisite concepts, cognitive demands, and symbolic structures. This enables adaptive task sequencing, gap diagnosis, and curriculum coverage reporting.

2. Dynamic Student Modeling The ITS shall maintain a dynamic Student Model that records procedural fluency, symbolic flexibility, and strategy preference over time. This model shall update after each interaction, updating and classifying learner progress in terms of mastery levels mapped to curriculum concepts (e.g., algebraic manipulation, function transformation). Common misconceptions shall be diagnosed using perturbation templates as described in [35].

3. Adaptive Feedback Mechanism The system shall implement an adaptive feedback module that generates formative feedback based on the student's intermediate steps and final response. Feedback must be sensitive to the strategy used (e.g., completing the square versus using the quadratic formula), and capable of recognizing both procedural and conceptual errors. Partial credit shall be awarded based on the degree of symbolic alignment with valid transformations, using a graded similarity metric derived from the equivalence engine.

4. Multiform Input Interface with Expression Parsing The user interface shall allow students to input mathematical expressions in various formats, including math editors, or plaintext algebra. An expression parser will convert input into canonical form, validate it syntactically, and pass it to the equivalence engine for semantic evaluation.

5. Symbolic Equivalence Engine The system shall integrate a symbolic computation engine (e.g., *Mathematica*) capable of verifying the equivalence of mathematical expressions across a wide range of transformation classes, including factoring, expansion, simplification, and identity substitution. This engine must support equivalence checks modulo domain conditions (e.g., $x \neq 0$) and be accessible via an internal API to both the Student Model and feedback module.

Implementation will follow a phased development cycle. In **Phase 1**, pedagogical and technical requirements will be specified through curriculum alignment and user scenario mapping. **Phase 2** will develop and integrate the domain and Student Models, including symbolic equivalence checks and data structures for tracking learner progression. In **Phase 3**, the Teaching Model logic and adaptive pathways will be constructed, incorporating branching scenarios and misconception libraries. **Phase 4** will involve usability testing, pilot deployment with a student cohort, and evaluation of learning gains and usability metrics. Each phase will be iterative, drawing on empirical feedback to refine the system.

Tutoria's design is grounded in the principle that intelligent systems should reflect not only formal knowledge but also the fluid, partial, and sometimes ambiguous ways that students understand and express mathematical ideas. By integrating a symbolic engine at the heart of its architecture, Tutoria moves beyond simple assessment paradigms to embrace equivalence of forms, process-level understanding, and adaptive feedback. This aligns with broader goals in educational technology to support deeper learning, formative assessment, and inclusive instructional design [101, 156].

5.3 The Four Models

In this section we consider how the functional requirements inform the design of the conceptual models of which the system is comprised.

5.3.1 The Domain Model

The Domain Model defines the core mathematical knowledge embedded within the ITS, including both the curriculum content and the reasoning processes that students are expected to engage with. In Tutoria, the pilot implementation focused on the topic of simple linear functions. This domain was selected due to its foundational nature in secondary mathematics and its well-documented status as a persistent difficulty area for transitioning students in UTS's first-year mathematics subjects. It provides a manageable scope for prototyping, while still offering rich opportunities for conceptual reasoning, transformation, and equivalence checking.

The curriculum content has been decomposed into discrete domain topics, each represented as a conceptual node in a directed graph structure. Each node encapsulates a single unit of mathematical knowledge—such as the concept of slope, rearrangement of equations, or recognition of equivalent linear forms. Nodes contain not only explanatory content and worked examples, but also define a set of reasoning rules or constraints that can be used to assess student input against the expected mathematical structure.

The graph structure enables both sequencing and prerequisite mapping: certain nodes must be mastered before others can be meaningfully attempted. For example, understanding how to isolate a variable is a prerequisite to solving an equation or transforming between equivalent forms. This layered dependency structure allows for adaptive navigation and personalised remediation paths.

In terms of conceptual representation, each topic node is defined by a set of domain constraints—rules or properties that describe what it means to "know" the topic correctly. These constraints are not merely syntactic (e.g., "did the student write the correct final answer") but also semantic and structural, enabling the system to identify when a student's expression is mathematically equivalent to the target form, even if procedurally different. This constraint-based approach supports richer formative assessment and feedback, especially when integrated with symbolic reasoning.

Tutoria represents this directed graph as a conceptual curriculum schema within its system architecture. While implementation specifics (e.g., database schema or API endpoints) are handled at lower system layers, the Domain Model defines the logical structure: each node contains metadata about its content type (theory, worked example, assessment task), its prerequisite relationships (edges), and its associated constraints for evaluation. This structure is extensible, allowing for growth into more advanced topics or additional content modalities in future development cycles.

By embedding this curriculum model within a symbolic engine-backed environment, Tutoria avoids the often more rigid templates of most LMS platforms. Instead, it supports open-ended symbolic responses evaluated for form equivalence, meaning that students receive credit and feedback for algebraically valid steps and transformations even when incomplete or unconventional. This design choice directly supports formative learning and is aligned with cognitive research into mathematical understanding [104, 88, 134]. Most traditional learning management systems (LMS) rely on fixed response types—multiple choice, numerical entry, or simple fill-in-the-blank interactions. These response types are easy to grade and standardise but are inherently limited in their capacity to engage with the subtle nuances of how students reason through mathematical ideas [134]. As a result, question authors working within these systems often find themselves confined to formats that restrict student thinking, and do not allow for partially correct or representationally varied responses.

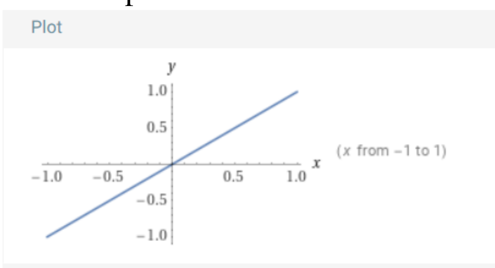
Tutoria, by contrast, allows students to submit answers in symbolic form and uses a symbolic reasoning engine to interpret whether those answers are mathematically equivalent to an expected solution. This means that a student who submits, for example, an expression equivalent to the standard form of a linear equation—even if it has not been rearranged into the canonical $y = mx + b$ format—can still be recognised as demonstrating valid conceptual understanding. In this way, Tutoria supports formative feedback based on structural equivalence rather than procedural conformity [88].

For example, suppose a student is working with the linear equation $2y - 3x = 6$. In a typical LMS environment, the question might require the student to solve explicitly for y, converting the equation into slope-intercept form. Unless the student completes all algebraic steps and submits $y = \frac{3}{2}x + 3$, the system may mark the response as incorrect. However, from a cognitive perspective, a response such as $2y = 3x+6$, while algebraically incomplete, reflects correct structural thinking. The student has made a key transformation by isolating the y-term and preserving equivalence [104]. In Tutoria, this partial correctness can be recognised and rewarded.

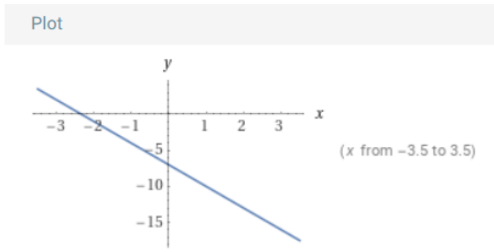
Example

Figure 5.1 illustrates the material contained in a domain node on the topic of simple linear functions. This contains descriptive instructional text with accompanying graphics, followed by a diagnostic exercise.

Consider linear functions. Such functions have a general form, $y = mx+b$ as the gradient intercept form, where m is the gradient and b is the y-intercept. Examples include the functions plotted below.



(a) Simple linear function $y = x$



(b) Simple linear function $y = -3x - 7$

It is a simple match to identify the gradient m and y-intercept b even if obscure...

Identifying constants

Example (a)	Example (b)
$y = x$	$y = -3x - 7$
then	then
m=1 and b=*0*	m=-3 and b=*-7*

Exercise After reading the material consider a question... The following function has an equation of the general form $2y+6x+12 = 0$. Write an equivalent form, i.e. rearrange the equation, with the gradient m and y-intercept b on the right hand side of the equation.

Fig. 5.1 Example content of simple linear function node

The exercise described in Fig 5.1 is a deliberately buggy question as it does not directly ask for the gradient or y-intercept — that step has to be inferred by the user. The system allows for the student's response to be incomplete and discerns the correctness or otherwise thereof.

Tutoria will accept a re-arranged equation that need only be of equivalent form. This error can then be used to guide the student to complete the final step. Although this is a very simple example, it illustrates the power of the ITS system in exposing the student's partial understanding, which be used to determine remedial steps.

Error checking is now able to become multi-layered: incomplete does not necessarily mean incorrect, and a penalty need not halt a student's progress through the curriculum, removing the stop/start nature of traditional assessment.

The architecture enables the system to interpret multiple mathematically correct representations as valid responses—a feature advocated by VanLehn [188] as essential for modelling ill-defined problem spaces. By offloading evaluation to a symbolic CAS, the Domain Model ensures consistency, accuracy, and depth in knowledge assessment.

5.3.2 The Student Model

The Student Model in Tutoria adopts a dynamic overlay approach that captures a continually evolving representation of a learner's understanding. This includes mastery of specific concepts, recognition of partially correct reasoning and the identification of persistent or emerging misconceptions. The model supports real-time instructional decisions by adapting curriculum navigation and feedback based on a student's conceptual profile.

As discussed in Section 3.7 and Section 3.4, the heart of this system is hybrid in nature. Model tracing, constraint-based modelling and overlays inform the Tutoria system.

Each domain topic is associated with a set of declarative rules or constraints that define correct conceptual behaviour. These rules and constraints are defined and selected through a combination of expert analysis by the system author of the mathematical domain (e.g., the algebra of linear functions), empirical observation of student errors, and pedagogical goals. Constraints are not procedural steps, but instead express what must be true of a correct solution—e.g., "the variable must be isolated on one side of the equation," or "both sides of the equation must remain equivalent under transformation."

Conceptually, as described in Section 6.3, constraints are stored as metadata within the domain node structure, linked to each curriculum concept. These constraints serve as formalised rules or expectations that guide the evaluation of student responses. They may be tagged with semantic categories (e.g., simplification, equivalence, isolation), expected symbolic patterns, and misapplication triggers, enabling the system to identify both correct reasoning paths and common misconceptions. These constraints are exposed to the symbolic engine (Wolfram Mathematica®) during evaluation, enabling the engine to test whether a student's response satisfies the relevant constraints or violates them in ways indicative of misunderstanding.

Model tracing is conducted by monitoring the sequence and structure of student interactions with each domain node. When a student responds to a task, their input is interpreted symbolically and evaluated against the constraints associated with the task's underlying concept. If all constraints are satisfied, mastery is inferred; if some are satisfied, partial understanding is logged; if key constraints are violated in identifiable ways, a misconception may be detected. Each evaluation instance produces a diagnostic trace that is recorded in the student's overlay model. This trace includes:

- the student's input,
- a list of satisfied and violated constraints,
- the concept node involved, and
- any inferred cognitive state updates (e.g., mastery increment, misconception flag).

These traces guide adaptive navigation: the next task presented to the student is selected based on the current state of the overlay model. For example, if a misconception is detected—such as by consistently failing to maintain equivalence during rearrangement—the system may route the student to a remediation node that explicitly addresses that conceptual gap.

Misconceptions are identified not merely by incorrect answers, but by patterned constraint violations that correspond to known reasoning errors. For instance, if a student repeatedly subtracts a term from only one side of an equation, violating the equivalence constraint, the system logs this pattern. Conceptually, these are stored as misconception profiles, associated with nodes and constraints, and flagged in the student's model. Once triggered, a misconception profile can:

- prompt tailored feedback in real time (e.g., a hint or explanation),
- trigger the presentation of targeted remedial tasks, and
- be available for later analysis in the teacher dashboard or system logs.

This infrastructure also enables flexible operational modes. Tutoria can be used as a simple LMS style quiz, with a predetermined sequence of questions with no dynamic adjustment. Or Tutoria can be used as an ITS where questions presented depend on student correctness. Thus Tutoria has two modes of use. In ITS mode, the Student Model drives intelligent feedback and dynamic progression. In assessment mode, the same tasks can be delivered in a fixed sequence without feedback, but student traces are still recorded for comparative evaluation. This dual use supports both formative and summative purposes and allows for cross-session analysis to measure student progression or recurring difficulties.

By combining constraint-based modelling, symbolic interpretation, and trace-based adaptation, Tutoria builds a responsive Student Model that supports dynamic development of mathematical reasoning — formative assessment, rather than summative assessment. This model foregrounds structural understanding and formative diagnosis —– crucial affordances for the transition challenges facing early university mathematics learners.

5.3.3 The Teaching Model

The Teaching Model governs the instructional decision-making processes within Tutoria, guiding how the system interprets learner input, selects tasks, and delivers pedagogical interventions. Drawing from principles established in the literature review (Section 2.5), the model is designed to foster conceptual understanding through structured feedback, adaptive support, and incremental task sequencing.

Tutoria employs a blended pedagogical engine that integrates both model-tracing and constraint-based approaches. Model tracing enables the system to compare student responses to idealised solution paths, identifying deviations that might signify conceptual missteps. Constraint-based logic, by contrast, checks whether a student's response satisfies a declarative set of domain-specific conditions. This dual mechanism allows Tutoria to distinguish constraint violations offering a more refined diagnosis of student thinking [123].

A core feature of the Teaching Model is its alignment with constructivist educational strategies, particularly Bruner's scaffolding [38]. As students interact with the system, they receive graduated instructional support in the form of worked examples, guiding prompts, or decomposed tasks. These supports are not static; rather, they are dynamically tailored based on the Student Model's evolving profile. If a student demonstrates partial understanding—e.g., isolating a variable correctly but failing to simplify—Tutoria recognises the valid step, updates the model accordingly, and provides targeted nudges toward further conceptual development.

In this way, feedback in Tutoria is formative, layered, and conceptual. It extends beyond simple correctness marking to acknowledge intermediate reasoning. For instance, starting a question of simple algebraic manipulation to make y the subject in $2y - 3x - 6 = 0$, when a learner submits $2y = 3x + 6$ instead of $y = \frac{3}{2}x + 3$, the system understands this as progress toward solving for y, even though it does not match the canonical form. This feedback model is inspired by Sadler's principles of formative assessment [156], which argue that learning is maximised when students understand where they are, where they need to be, and how to get there.

This pedagogical flexibility is further enabled by Tutoria's use of symbolic reasoning, which supports a wide range of mathematically valid forms. Unlike traditional LMSs that rely on rigid input types (multiple choice, fill-in-the-blank), Tutoria can evaluate the semantic equivalence of expressions using Mathematica. This allows the Teaching Model to reward structural understanding and recognise divergent yet valid forms of student reasoning, reflecting Kaput's insights into symbolic fluency as a hallmark of deeper mathematical understanding [101].

To embed effective teaching strategies into the system, task design in Tutoria is informed by APOS theory [72] and Piagetian abstraction (see Section 2.3.1). Each activity is scaffolded to promote a progression from Action (e.g., manipulating algebraic expressions), to Process (recognising patterns and transformations), to Object (internalising mathematical entities like equations), and finally to Schema (coherent networks of understanding). These stages are encoded through the curriculum node graph: early nodes target concrete manipulations, while later nodes require internalisation and application across varied contexts.

When a student struggles, the system does not merely loop them through the same task. Instead, the Teaching Model guides the learner to adjacent nodes representing simpler or precursor concepts. This vertical and lateral navigation supports conceptual backfilling and ensures that failure becomes a trigger for support rather than exclusion. For instance, persistent errors in rearranging equations may prompt redirection to tasks that focus explicitly on preserving equality through transformations.

Misconception handling is also central to the Teaching Model. Conceptual errors—such as assuming one can subtract from only one side of an equation—are identified via structured error patterns in the constraint base. Once a misconception is triggered, it is recorded in the Student Model and flagged for subsequent instructional action. This may result in immediate intervention (e.g., a corrective hint), or in deferred action, such as adjusting the sequence of future tasks. Over time, these entries build a diagnostic profile that supports longitudinal understanding of learner development.

In sum, Tutoria's Teaching Model operationalises research-backed strategies into a cohesive instructional engine. It offers a responsive, theoretically grounded, and technically enabled pathway for supporting deep mathematical learning.

5.3.4 The Student User Interface

In this section we will focus on the student user interface, which of course assumes that the ITS resources and content have already been developed and are ready for use. In fact, the ITS user interface also facilitates this development activity via a separate author user interface entry point. The authoring subsystem is discussed in Chapter 6. Designed to be up to date, the website behaviour includes the latest UI elements (see Definition of Terms), which also includes UX, as part of an extensible web platform.

The student-facing interface of the system was designed to satisfy several pedagogical and functional imperatives identified in preceding chapters. Conceptually, the interface acts as the primary access layer through which learners engage with content, feedback, and symbolic reasoning services.

Accessibility and Intuitiveness Given the diverse academic backgrounds of incoming students, the interface prioritises accessibility and intuitive design. This is realised through a layered architecture where user input is abstracted from underlying processing. The topmost layer presents a minimal cognitive load UI—simple input boxes, clickable controls, and clearly structured navigation—allowing students to focus on the mathematics rather than the mechanics of the tool. This layer leverages standard accessibility best practices such as keyboard navigation, responsive design, and compatibility with screen readers. Conceptually, this ensures that the interface accommodates students with differing levels of technological fluency and physical access needs.

Interactivity, Extensibility, and Scalability The system scaffolds for interactivity and adaptability to future extensions. As outlined in Section 8.2, the eventual inclusion of real-time visualisation, adaptive scaffolding, and structured feedback is anticipated. The current interface design supports this by modularising the interaction at the presentation layer, where interactive components (e.g., input fields, feedback panels, visualisation tools) communicate with server-side evaluators via an API abstraction. This separation ensures that pedagogical logic and constraint evaluation remain decoupled from the user interface, facilitating maintainability and scalability. This facilitates extension and decouples user interaction logic from domain logic. Scalability is supported through asynchronous communication and stateless request handling, allowing the same interface design to be deployed across a range of delivery contexts and user loads.

Mathematical Input Handling Recognising the unique demands of mathematical communication, the interface supports multiple forms of input. Students can enter expressions using a math editor (WYSIWYG) or plaintext algebraic syntax. An embedded parser processes the input at the syntactic level, converting it to a canonical internal form. This form is then semantically validated and passed to the equivalence engine for evaluation. From the user's perspective, this allows flexibility in expression while ensuring consistent system behaviour.

The student interface model provides the user-facing environment where learners interact with problems, input expressions, and receive feedback. It is designed to be minimal, intuitive, and symbolic by nature. Tutoria utilises a web-based input tool, 'jw-formulaeditor', to allow students to construct mathematical expressions naturally within the browser. Figure 5.2 illustrates the flexibility of the input mechanism: the expression for a well-known convergent infinite series, with sum 2, is entered as the gradient value for a simple linear function. It just so happens the gradient being sought in the question is also the number 2. The interface passes this input to a Mathematica kernel, which in turn affirms the correctness of this entry.

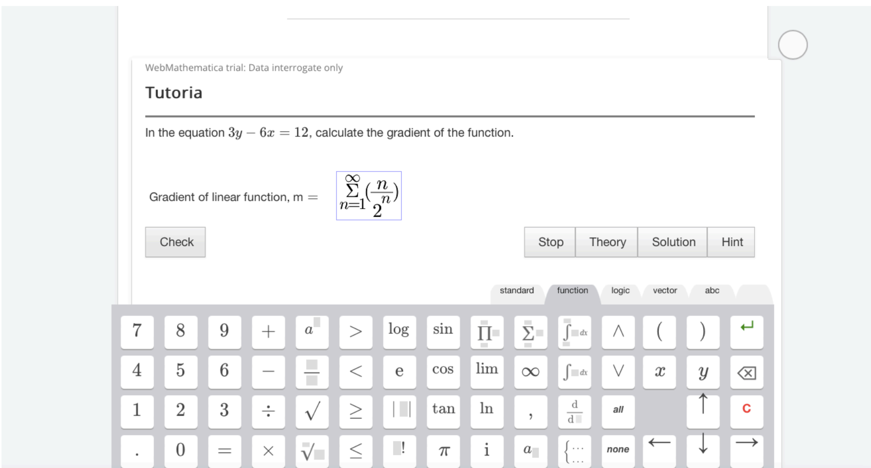


Fig. 5.2 Flexible mathematical input

More generally, input expressions are parsed and asynchronously evaluated via Java servlet calls to an Apache-Tomcat server, which hosts Mathematica kernel pools for symbolic computation. This configuration, inspired in part by environments like Andes [80], ensures that user interaction is smooth and uninterrupted by page reloads, while enabling immediate validation and feedback. The interface model is thus a crucial bridge between learner cognition and the computational concordances of the system.

Student user interface modes of operation Tutoria has two methods to the ITS interface.
Method 1: This approach is a straightforward list of links, which the user selects, either under instruction or self directed, 'to do' questions. See Section A.2.2 in Appendix A for figures depicting the entry point. Once a selection is made, the questions can be completed and scores counted. As has been previously mentioned, the list may use either standard MCQ type questions or dynamic Tutoria style questions, this is set by the author using authoring tools (see Chapter 6). Results release is also configurable, either letting the student see the result or not. Students were not shown results during the evaluation experiment.
Method 2: This approach is more complicated. Again, figures depicting the process are found in the link to the relevant appendix given above.

5.4 From Models to Layers

As we have seen, each of the models comprising Tutoria relies on services and resources provided by a number of underpinning software layers. We summarise these below, before giving a more detailed description of each in the following sections.

The Application layer represents the human computer interaction layer where ITS services can be accessed. Its primary function is to provide both the student and the ITS author with suitable ITS user interfaces. For the student, a typical interface page sits as the entry point on a student's journey into the ITS system, where they interact mainly with both the Domain Model and the Teaching Model. The Domain and Teaching Models themselves are realised within the ITS by nodes with resources and content located in the underpinning layers and, from the point of view of the student user, are already in place internal to the ITS. On the other hand, the ITS author requires an interface that facilitates the creation and configuration of the curriculum resources required by the Domain and Teaching Models.

The Presentation Layer translates the user interface input and output and coordinates the formatting and preparation of data for session responses. Its primary function is to provide a logical buffer between interface code and libraries supporting the formatting of data, data typing, minor computations, database management, security and inter-server connections.

The Session Layer safeguards the ITS models ensuring their integrity by managing database transactions. Its primary purpose is to faithfully store data according to the blueprint set forward in and during authoring. Such storage is a complex inter-related connection of resources related to content, parameters - holding characteristic information about questions, their use and deployment, meta information relating to curriculum maps and of course Student Models. Logically these interconnected entities are referred to as nodes. Each node can be thought of as encapsulating a data object that describes its associated parameters within both the Domain Model (representing the structure of curricular knowledge) and the Teaching Model (representing pedagogical parameters). Students have assigned to them a reflection of every node visited, the Student Model (representing a learner's evolving understanding), while interacting with the ITS, in the form of a database entry referred to as a primary key. This unique key is used to track student interactions within the ITS. A list or table of such keys (sessions in the session layer) cross references student activity against available nodes and represents the Student Model.

The Apache Layer hosts the HTTP server. The purpose of this is to host files populated with tags known as markup. Files of this type, HTML, are served to local machines with internet browsers. The browsers process the mark up and present pages to end users like authors and students. Such pages are known as static pages. If the server includes Hypertext Pre-Processing, PHP, then the server can serve pages dynamically. Files with a combination of PHP and HTML are rendered for the user to view, often with a form containing controls such as 'submit' buttons, tick boxes and the like. On form submission the data from the form is placed in an object which is passed back to the server for further processing. Post processing, the markup page is returned to the browser for viewing, with modification made by the PHP libraries called. This is a continually reoccurring process.

The Apache-Tomcat Layer hosts Java scripts and is also capable of rendering HTML markup pages for browsers to process. As well as this, Java applications like WebMathematica can be hosted. WebMathematica can pass data to and from an installed computer application like Mathematica. Mathematica can then be used to process input just as if a user had typed into the computer application. Apache-Tomcat has the primary function of passing data from Apache to the Mathematica program, retrieving the processed data and returning it to Apache. The data is the student response and so forms part of the Student Model.

The Mathematica Layer is responsible for the symbolic computation and mathematical processing at the core of the system. When invoked by the Apache-Tomcat Layer —

typically via WebMathematica, like hidden scaffolding —– Mathematica executes scripted computations which may include algebraic manipulation, function evaluation, plotting, and step-by-step solution generation. These computations are triggered in response to student input submitted through the PHP-rendered interface.

Once the input data is passed from the Apache-Tomcat servlet to Mathematica, it is parsed and processed within a Mathematica kernel instance. The results — be they numerical answers, symbolic expressions, or dynamically generated visualisations — are returned in a form suitable for rendering in a web environment, often encoded back into HTML with embedded MathML or images. The processed output is then passed back to the Apache Layer for display in the browser.

This layer plays a critical role in the Student Modelling process, as it enables the generation of system responses that are mathematically accurate and pedagogically informative. It supports not only the evaluation of correctness but also provides insight into the nature of student errors, which is essential for intelligent tutoring functionality. The Mathematica layer thereby functions as both an inference engine and an instructional tool, integrated seamlessly with the web infrastructure provided by Apache and Apache-Tomcat.

The Configuration layer has responsibility to initiate and maintain all of the servers, their behaviours and resources. Configuration is matched literally to hardware by special code libraries which interact with the operating system being used. This means there is a dependency on the shared libraries being supported by the operating systems used. This can become complicated as it is possible to host Apache on Windows and Apache-Tomcat on a Unix derivative, Linux, and vice versa. For this project all systems resided on the same operating system, an Apple laptop.

5.5 The Layers

5.5.1 The Application Layer

As indicated in Section 5.4, the Application layer represents the human computer interaction layer where ITS services can be accessed. Its primary function is to provide both the student and the ITS author with suitable ITS user interfaces. A student will interact mainly with both the Domain Model and the Teaching Model. On the other hand, the ITS author requires an interface that facilitates the creation and configuration of the curriculum resources required by the Domain and Teaching Models. These models themselves are realised within the ITS by nodes with resources and content located in the underpinning layers.

Each node has particular classes of content, including static data. Examples include question content and stored student answers to those questions. Nodes also contain data whose application determines dynamic responses during user interaction. Examples include decision pointers which determine the exiting path on leaving a node. Such content is stored in a database managed by the Apache layer. Data has in it foundational instructional materials, such as explanatory text, worked examples and assessment tasks, along with metadata reflecting the node as MySql database records. These resources are referenced by the Domain Model but retrieved and rendered through the application layer, which interfaces with the Apache and Apache-Tomcat and Mathematica layers to assemble and deliver content. Additions, like website enhancements such as UI and UX (see Definition of Terms), also discussed in Section 5.3.4, are encoded into the application layer's API routines which then present this content in a minimal and accessible form, consistent with the system's prioritisation of clarity, interactivity, and responsiveness for diverse learner profiles.

Nodes also require access to task logic and symbolic computation routines, particularly where dynamic content or automatic evaluation is required. A movement in a particular direction away from a node triggers these services which are provided via integration with the symbolic engine, detailed in Section 5.5.5, which is hosted within the Apache-Tomcat layer. When a user submits a mathematical expression or engages with an interactive task, the application layer mediates the request, passing data to the symbolic engine , via the Apache layer and retrieving processed output (e.g. feedback, equivalence checking, or solution steps). This tight coupling allows the system to support advanced reasoning tasks without sacrificing usability.

Navigation between nodes is managed through routing logic embedded in the dynamic response triggers of nodes, with progression rules grounded in the structure of the Domain Model. These rules are accessible to the application layer, which updates the user's path and orchestrates transitions based on either learner input or interventions from the Teaching Model. For instance, if the symbolic engine detects a systematic error pattern, the Teaching Model can trigger a redirection to a remediation node. These routing decisions are immediately reflected in the application layer's API, ensuring that content transitions remain pedagogically meaningful and technically seamless.

The application layer also supports ongoing synchronisation with the Student Model. As users interact with tasks, the system captures response data—including input patterns, time-on-task, and symbolic complexity—and passes it to model-tracing routines in the session layer. These routines, discussed in Section 5.3.2, update the learner's profile, which in turn shapes future node selection, hint generation, or feedback scaffolding. Through these mechanisms, the system realises its goal of adaptive, student-aware instruction, with each node acting as a touchpoint between domain structure, learner modelling, and instructional decision-making.

5.5.2 The Presentation Layer

The Presentation layer as mentioned coordinates the interaction between the Application layer and the Session layer. Its primary function is to provide opportunities for translation and interaction with code and libraries supporting the formatting of data, data typing, minor computations, database management, security and inter-server connections. By having a controller, so called, the interface code need only reference the controller for any number of tasks. Those tasks then being completed, where appropriate in the controller, are passed off to task specialists. The Presentation layer effectively acts as a conduit and switch for the ITS models, Domain, Learning and Student, passing them to the appropriate agent for finalisation.

Controllers Inputs to the interface are in the form of pre-testing study material and questions to be answered. The Joomla administration system controllers take page requests, interrogate the database store of resources and return them for processing by the user. A typical handler of such requests is created in PHP (see 5.5.4 for a brief description of this term) by a code snippet thus,

```
<?php
    class INconnectControllerNodes extends INconnectController
    {function __construct()
      {
        parent::__construct();
      }
      function display($cachable = false, $urlparams = false
      {
        JRequest::setVar('view','nodes') ;
        parent::display() ;
```

```
    }
?>
```

Views This PHP class loads the list of nodes. The primary purpose is to render the display of the view being entrained, in this case the ‘Nodes’ view, via the ‘display function’. This is an interrogation of the available nodes requested from the Domain Model. The view class has a code snippet:

```
<?php class INconnectViewNodes extends JViewLegacy {
  function display($tpl = null) {
    $model =& $this->getModel();
    $publicNodes = $model->getPublicNodes();
    $userNodes = $model->getUserNodes();
    $this->assignRef( 'publicNodes', $publicNodes);
    $this->assignRef( 'userNodes', $userNodes);
    parent::display($tpl);
 }
} ?>
```

Once the user has decided to request a particular Domain Model object, consisting of authored resources, data from the application layer is posted to the controller associated with the node being used.

Collecting data from the user As data is being returned controllers will format data for a save request to the Session Layer. Authoring additions of a ‘meta’ data type include the decision making process instructions stored with Domain Model objects. Attached as author set parameters this data holds the instructions as to where to go on success and where to go on failure.

‘Post’ to the Session layer A typical Domain Model data return, known as ‘post data’ appears below:

```
J!Dump - site 4/9/2024, 9:40 am
Application: site
Lock Window Refresh Close Window Expand all Collapse all
[array] $data_post_get:
[string] 34b549ec3a303232b09528e012cbd124 = "1"
[array] answers
[string] 737 = ""
```

```
[string] id = "53"
[string] openmath = "8y+10=-7x"
[string] option = "com_inconnect"
[string] q5_customTextbox5 ="http://localhost/Joomla_3/index.php?
option=com_inconnect&view=nodes&Itemid=106"
[string] result = "True"
[string] send = "Submit answers"
[string] sessionId = "7465"
[string] showChain = "1"
[string] showResults = "0"
[string] task = "submitAnswer"
[string] timeUp = "0"
[string] view = "node" ..." (Length = 160)
J!Dump v2012-10-31
 http://localhost/Joomla_3/index.php?option=com_dump&view=tree&format=raw
 Page 1 of 1
```

Visible in the code above are the node id, which is 53, and the response string. Also visible is the result "True", as there was a dynamic asynchronous JavaScript (or 'ajax') call made prior to the user submitting the result form. Thus the result from Mathematica is already known, hence the result = True.

The next step is to save the data. The code snippet for this is:

```
<?php  function submitAnswer()
       {JRequest::checkToken() or jexit( 'Invalid Token' );
          $data_post_get = (Array)JRequest::get('post');
          dump($data_post_get, '$data_post_get: ');
          JLog::add(JText::_(INconnectHelper::indentjson(
            json_encode($data_post_get)).' = $data_post_get
            line 129 inconnect site node controller '), JLog::WARNING, 'com_inconnect'
          );
       $model =& $this->getModel('node') ;
       $showchain = $model->isShowChain($data_post_get['id']);
       $showchain1 = $model->isShowChain1($quizId);
       $show = ($bool)? 1 : 0 ;
       if ($model->storeAnswers()) { . . . } ?>
```

We continue the discussion in the following Section concerning the intention of this 'post', which has been saved in a relational database, and can now be viewed as a session. The Session layer, as we shall see, deals with the storage and processing of session records.

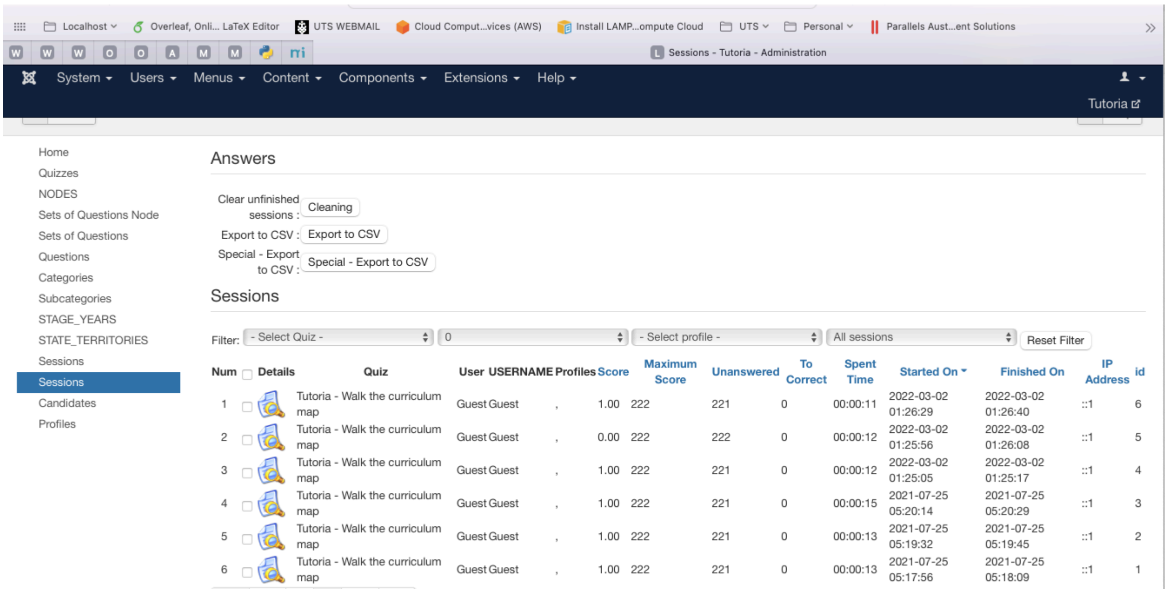


Fig. 5.3 Sessions from Tutoria

5.5.3 The Session Layer

The Session layer's primary purpose is to faithfully store data in a relational database, according to the design set forward in and during authoring. Such storage occurs using a complex inter-related collection of resources. These resources are used for all three ITS models and the student interface. Typically, Domain Model data is populated with parameters. Parameters are content about content, the parameters are designed to hold characteristic information about questions, their use and deployment. For example, parameters determine if a question is to be used in the MRS style or ITS style. This meta information relates to curriculum maps and of course the Student Model. Nodes are formed representing the Domain Model, where curriculum is stored. Each node can be thought of as having a data object, populated with parameters describing the Models of Domain and Teaching.

Students have assigned to them a reflection of every node visited while interacting with the ITS. One parameter, in the form of a database record field, is referred to as a primary key. This unique key is used to track student interactions within the ITS. A list or table of such keys, cross references student activity against visited nodes. Sessions are formed by tracking this data. The session manger in this project is comprehensive, recording sessions of student activity. Figure 5.3 shows a log of sessions undertaken by a student. Student interaction data can be exported from such logs through the session manager, via a PHP web page service or a command-line HTTPS TTY, for subsequent analysis. Both methods were used.

Data from evaluation of the pilot-deployment of Tutoria, which will be discussed in Chapter 7, was accessed[1] via the Apache server. Both:

- by Putty© (tele-type client) and
- as part of the authoring procedure (PHP service).

Both methods then access MySql where selective queries run to obtain the necessary data. The saved exports were then compared and then imported into Excel, for further comparison, and from there imported into R-Studio, again for further comparison. A snippet of the code used to retrieve the session information follows.

```
SELECT quizSession.id AS id, quizzes.title,
  ( SELECT sum(score) FROM dthvx_inconnect_quizzes_answersessions
    WHERE quizsession_id = quizSession.id AND status <> -1 ) AS score,
      ( SELECT count(distinct(question_id))
        FROM dthvx_inconnect_quizzes_answersessions
       WHERE quizsession_id = quizSession.id ) AS maxScore,
      ( SELECT count(id) FROM dthvx_inconnect_quizzes_answersessions   WHERE status=-2
   AND quizsession_id = quizSession.id ) AS unanswered,( SELECT count(id)
   FROM dthvx_inconnect_quizzes_answersessions
   WHERE status=-1  AND quizsession_id = quizSession.id ) AS evaluate,
          quizSession.spent_time, quizSession.started_on,
          quizSession.finished_on, quizSession.ip_address,
          sessionWho.user_id, sessionWho.givenname,
          sessionWho.familyname, sessionWho.email
FROM dthvx_inconnect_quizsession AS quizSession
  LEFT JOIN dthvx_inconnect_users_quizzes
    AS users_quizzes ON users_quizzes.id = quizSession.affected_id
  LEFT JOIN dthvx_inconnect_quizzes
    AS quizzes ON quizzes.id = users_quizzes.quiz_id
  LEFT JOIN dthvx_inconnect_quizzes_answersessions
    AS quizSessAns ON quizSessAns.quizsession_id = quizSession.id
  LEFT JOIN dthvx_inconnect_sessionwho
    AS sessionWho ON sessionWho.session_id = quizSession.id
  WHERE quizSession.id = session_id_of_interest...
```

[1] Accessing an Apache sever can be by PHP code server to a web browser but also by secure HTTPS TTY tele-type entry from the command-line. During development both methods were used to debug and make code production ready.

5.5.4 The Apache Layer

As mentioned in Section 5.4, the Apache layer hosts the Apache HTTP server, the purpose of which is to host files populated with tags known as markup. Files of this type, HTML, are served to local machines with internet browsers. The browsers process the mark up and present pages to end users, such as authors and students. Such pages are known as static pages. If the server includes Hypertext Pre-Processing, PHP, the the server can serve pages dynamically. Files with a combination of PHP and HTML are rendered for the user to view, often with a form containing controls such as submit buttons, tick boxes and the like. On form submission, the data from the form is placed in an object which is passed back to the server for further processing. After processing, the markup page is returned to the browser for viewing, with modifications made by the PHP libraries called. This is a continually reoccurring process.

Joomla The open-source Joomla content management system was selected to manage HTML content due to its extensibility, user management features, and ease of integration with custom plugins [100]. By capitalising on the already developed code base in Joomla many of the class-loading libraries required by Tutoria were already established and needed little adjustment to suit the role of an HTML content manager, for MRS style testing. The recommended procedure for programmers interacting with Joomla is to code a component, module and plugin, since then the application does not interfere with the Joomla platform. Applications made compliant in this way can be commercialised fairly readily. In the case of the ITS this was not possible as the Joomla core did not adequately meet requirements in areas of database management, Java interaction and UX libraries contributing to user experience. The Joomla system running the Tutoria ITS was modified to rectify these gaps, and must be considered a prototype system.

A summarised top level directory tree shows the important elements of the prototype system:

```
/Joomla
├── core
├── administrator ........................ ITS system author tools
│   └── components
│       └── com_inconnect
└── components ........................... ITS system client site code
    └── com_inconnect .................... client site component
```

5.5.5 The Apache-Tomcat Layer

This layer has the primary function of hosting WebMathematica. It uses the Apache Tomcat open-source software to handle data exchange between backend components and the Apache HTTP/PHP server, enabling servlet-based communication and integration with Java-based services [14]. The Apache-Tomcat server manages communications with the Mathematica program by submitting and retrieving the processed data and returning it again to the Apache HTML/PHP server. The data are student responses and so, once stored, form part of the Student Model.

WebMathematica uses 'so' libraries on unix style bsd/linux systems and 'dll' libraries on Windows platforms to pass data to and from an installed version of Mathematica©. This project tested both types.

For testing purposes, any Mathematica logical code can be checked in the proprietary Mathematica application notebook. Once tested it is a relatively simple matter to create the MSP (see Definition of Terms) scripts (as *.jsp files) required to invoke WebMathematica from the server. The MSP scripts reside deep within the Web-Mathematica code tree:

```
/Apache_Tomcat
└─ webapps
   ├─ examples ............................................ e.g. integration
   ├─ webMathematica
   │  └─ evaluate_ajax
   │     └─ answer ...................................... Tutoria evaluation code
   │        └─ Answers_jsp
   └─ WEB-INF ............................................. further local config
```

PHP/HTML calls from the Joomla system come in to the MSP script file, Answers jsp. The code takes the questions as provided by the authoring system and the answers as provided by the client user and determines a comparison using a Boolean statement. The return to the Joomla system then processes the result and the decision concerning navigation within the curriculum system subsequently takes place. The actual code with return string processing redacted follows:

```
<msp:evaluate>
  answers = FullSimplify[MSPToExpression[ $$answer, MathMLForm]];
  answer = ToString[answers];
  questions = FullSimplify[MSPToExpression[ $$question, MathMLForm]];
  question = ToString[questions];
  result = ToString[answers === questions];
  If[ result === "True",
```

```
    format return string and set truth state...
</msp:evaluate>
```

5.5.6 The Mathematica Layer

The purpose of this layer is to enable Tutoria to interact with Mathematica kernels running in the background. The main requirement for this is the installation of a valid license and setup information, as in the following script.

```
$PasswordFile
### returns ###
/Users/jasonstanley/Library/Mathematica/Licensing/mathpass
$BaseDirectory
### returns ###
/Library/Mathematica
```

The 'mathpass' license file needs to be copied to the Base directory tree on the computer running the Tutoria system. In the case of this project, this is:

```
/Library/Mathematica
├── Applictions
├── Autoload
├── FrontEnd
├── Kernel
│   └── init.m
├── Licensing
│   └── mathpass ............................ this one needs to be copied here
├── Paclets
│   └── Repository
└── SystemFiles
    └──
```

Adjustments are made to the XML file for correct operation.

```
    <KernelPool><KernelPoolName>General</KernelPoolName>
        <URLPattern>/*</URLPattern>
    </KernelPool>
### Edit and change this section to:
    <KernelPool><KernelPoolName>General</KernelPoolName>
        <KernelExecutable>
          /Applications/Mathematica.app/Contents/MacOS/MathKernel
        </KernelExecutable>
```

```
        <URLPattern>/*</URLPattern>
    </KernelPool>
```

Once these changes are made, along with further configuration settings (see Section 5.5.7), calls from PHP code to Mathematica will initiate the relevant MSP script and return values.

5.5.7 The Configuration Layer

Configuration can be considered the foundational layer in this context; without it, the system cannot function. It also represents the component with the highest maintenance burden in the ITS application. Configuration is matched literally to hardware by special code libraries which interact with the operating system/s being used. This means there is a dependency on the shared libraries being supported by the operating systems used. It can become complicated as it is possible to host a variety of Apache servers from the family of available Apache products on a variety of hardware servers. For this project all systems resided on the same system, an Apple laptop. A brief description of the hardware and software used follows.

The system was implemented on an Apple© Macbook Air. Mac-Ports was used to implement the required support libraries from native Free-BSD or Mac-Ports repositories. Generic open source Apache, Apache-Tomcat and Java software tools were compiled from source as this allowed for higher reliability among dependencies used in the operating system library interfaces.

Open source applications like PHP, Java and Mathematica scripting (MSP), Perl, Mysql (for database purposes) and any ancillary libraries were then attached.

The Apache server serves applications, with this application code base using Hypertext Pre-processor, PHP, which supports auto class loading of PHP code classes. This feature motivates the adoption of class-based architectures, such as the Model–View–Controller (MVC) pattern implemented in this project. Joomla© formed the basis application for client side server interaction and the server side Authoring and configuration tools.

The system is a complex system in that Joomla hosted on an Apache Server interacts with Apache-Tomcat, a Java server, to provide transparent transactions with Mathematica, performing symbolic engine processing computations silently. Notwithstanding the complex interplay Joomla has with the Apache-Tomcat server and Mathematica, Joomla also needed to be highly extensible, to allow for the implementation of the models required for an ITS.

In particular, a custom code library was developed for use as a basis for the question/answer authoring system which will be discussed in Chapter 6. A brief description of the configuration requirements for each of these systems can be found in Appendix C and Appendix D.

5.6 Concluding Remarks

The SOWISO CAS Tutoria's conceptual and technical design exists within an ecosystem that includes several commercial intelligent mathematics platforms. Among these, SOWISO—a proprietary system developed in the Netherlands—is notable for its symbolic evaluation capabilities and scalable infrastructure.

Originally developed as a spin-off from the Eindhoven University of Technology, SOWISO employs a symbolic backend based on Maxima, an open-source computer algebra system derived from DOE-Macsyma [115]. Maxima enables symbolic differentiation, simplification, integration, and equation solving, and is lightweight enough for web-based deployment. SOWISO delivers symbolic evaluation via Java EE application servers and structured APIs, facilitating real-time client-server interactions within a Microsoft-based environment [169].

While not a direct competitor, SOWISO and Tutoria share certain architectural elements, such as the adoption of 'jw-formulaeditor', a JavaScript-based mathematical input tool for HTML interfaces. However, their computational backends differ substantially. SOWISO uses a customised, Maxima-integrated version of the formula editor, whereas Tutoria retains a standard open-source implementation connected to a bespoke evaluation pipeline powered by Wolfram Mathematica®.

Tutoria's use of Mathematica introduces several technical distinctions. Symbolic input is parsed through an Apache-Tomcat server that manages pools of Mathematica kernels, enabling asynchronous symbolic evaluation without page reloads. This design supports advanced algebraic equivalence checking and function-level comparison, leveraging the full expressiveness of the Wolfram Language [195]. In contrast to rule-based equivalence systems, this allows for flexible handling of diverse mathematical expressions in real time.

The decision to integrate Mathematica over Maxima was informed by compatibility with institutional infrastructure and the need for robust documentation, cloud-integration capabilities, and validated symbolic accuracy. While Maxima offers a viable open-source alternative, its development trajectory and tooling support were less aligned with project requirements.

These distinctions reflect differing design philosophies and technical priorities rather than value judgements. Tutoria's architecture aims to facilitate high-fidelity symbolic interaction in a research-oriented context, with a particular emphasis on modularity, precision, and asynchronous responsiveness.

Future developments While Tutoria is currently designed to provide individualised mathematical support, its pedagogical framework is extensible to collaborative learning contexts. Future implementations may incorporate principles from Activity Theory, enabling paired or small-group work mediated by the system. In such scenarios, shared tasks could be used to scaffold peer interaction, with the platform prompting dialogue and encouraging co-construction of mathematical understanding. This aligns with contemporary educational paradigms that emphasise dialogic learning and collective problem-solving [140, 98].

A planned extension to the interface involves the visual or structural rendering of the Domain Model as a dynamic node graph. This would provide learners with a clearer sense of their progression through the content, highlight conceptual dependencies, and reveal areas of difficulty or conceptual gaps. Visualising the graph may also support self-regulated learning by making the structure of mathematical knowledge explicit.

In addition, the domain content is designed to be modular, allowing for the gradual expansion of supported topics beyond the current curriculum scope. Future iterations may include topics from calculus, statistics, or discrete mathematics, adapting the same symbolic and pedagogical infrastructure. Enhancements to the Student Model—such as integration of learning analytics or fine-grained misconception tracking—are also under consideration to further personalise feedback and instructional trajectories.

These directions reflect an ongoing commitment to pedagogical adaptability, cognitive responsiveness, and architectural extensibility, and are revisited in Chapter 8.2.

Chapter 6

Authoring

6.1 Overview

A key feature of the Tutoria system is its extensible authoring interface, which enables rapid construction and iterative refinement of pedagogical goals within domain structures. Authoring in Tutoria is not simply content creation, but a formal expression of the system's underlying architecture. That is, each authoring task corresponds directly to one or more components of the domain, teaching, or student models. Consequently, the design of the authoring interface directly affects the system's adaptability to curriculum change, pedagogical revision, mode switching (ITS tutoring mode versus summative assessment mode), and new insights into student learning behaviour, including misconception handling.

The aim of this chapter is to describe how the authoring interface supports the ITS intentions. The emphasis is placed on how authors create and interconnect nodes, define constraints, and determine system behaviour in both instructional and assessment contexts. The Tutoria Student Modelinterface is best understood not simply as a content editor, but as a visual interface to the internal models of the system. Its structure enables authors to scaffold learning pathways, encode pedagogical intent, and refine the student experience in light of theoretical and empirical developments.

6.2 Other Authoring Systems

While a range of ITS authoring systems have emerged since the 1990s, many face trade-offs between flexibility, pedagogical richness, and ease of use. Murray's 1999 classification provides a useful taxonomy of early systems [128], ranging from curriculum sequencing tools to intelligent hypermedia environments. More recent systems include CTAT [5], which supports model-tracing, and ASPIRE [126], which uses constraint-based modelling. Although powerful, these systems typically require significant technical overhead or impose pedagogical rigidity.

Tutoria was designed with a different set of priorities: to tightly integrate symbolic reasoning, misconception handling, and flexible curriculum progression in a way that remains accessible to non-programmer authors. Unlike systems which presuppose either fixed curricula or implicit pedagogical sequences, Tutoria treats authoring as a dynamic, model-aligned process in which instructional content, remediation paths, and constraint-based feedback are all editable within a single unified interface. This chapter illustrates how these capabilities are exposed through the authoring interface and how they map onto the underlying architecture.

Some tools are summarized in tables below, both reproduced from Murray:

Table 6.1 ITS authoring tools by category, from [128].

	CATEGORY	EXAMPLE SYSTEMS
1	Curriculum Sequencing and Planning	DOCENT, IDE, ISD Expert, Expert CML
2	Tutoring Strategies	Eon, GTE, REDEEM
3	Device Simulation and Equipment Training	DIAG, RIDES, SIMQUEST, XAIDA
4	Domain Expert System	Demonstr8, D3 Trainer, Training Express
5	Multiple Knowledge Types	CREAM-Tools, DNA, ID-Expert, IRIS, XAIDA
6	Special Purpose	IDLE-Tool/IMap, LAT
7	Intelligent/adaptive Hypermedia	CALAT, GETMAS, InterBook, MetaLinks

Table 6.2 ITS authoring tool strengths and limitations by category, from [128].

CATEGORY	STRENGTHS	LIMITS	VARIATIONS
Curriculum Sequencing and Planning	Rules, constraints, or strategies for sequencing courses, modules, presentations	Low fidelity from student's perspective; shallow skill representation	Whether sequencing rules are fixed or author-able; scaffolding of the authoring process
Tutoring Strategies	Micro-level tutoring strategies; sophisticated set of instructional primitives; multiple tutoring strategies	(same as above)	Strategy representation method; source of instructional expertise
Device Simulation and Equipment Training	Authoring and tutoring matched to device component identification, operation, and troubleshooting	Limited instructional strategies; limited Student Modeling; mostly for procedural skills	Fidelity of the simulation; ease of authoring
Domain Expert System	Runnable (deeper) model of domain expertise; fine grained student diagnosis and modeling; buggy and novice rules included	Building the expert system is difficult; limited to procedural and problem solving expertise; limited instructional strategies	Cognitive vs. performance models of expertise
Multiple Knowledge Types	Clear representation and pre-defined instructional methods for facts, concepts, and procedures	Limited to relatively simple fact, concepts, and procedures; pre-defined tutoring strategies	Inclusion of intelligent curriculum sequencing; types of knowledge/tasks supported
Special Purpose	Template-based systems provide strong authoring guidance; particular design or pedagogical principles can be enforced	Each tool limited to a specific type of tutor; inflexibility of representation and pedagogy	Degree of inflexibility
Intelligent/adaptive Hypermedia	WWW has accessibility and UI uniformity; adaptive selection and annotation of hyperlinks	Limited interactivity; limited Student Model bandwidth	Macro vs. micro level focus; degree of interactivity

For a discursive analysis of those ITS authoring tools as listed see [128].

6.3 Tutoria Author User Interface

Tutoria's authoring system is accessed via the Joomla backend interface, authenticated through the Apache layer described in Section 5.5.4. Once logged in, users are presented with an administrative interface served by the Application Layer (Section 5.5.1), providing access to tools for navigating, editing, and creating nodes within the Domain Model. The interface makes visible the dual-layer logic of Tutoria's instructional design: one layer capturing conceptual structure, and another encoding the rules that govern pedagogical flow.

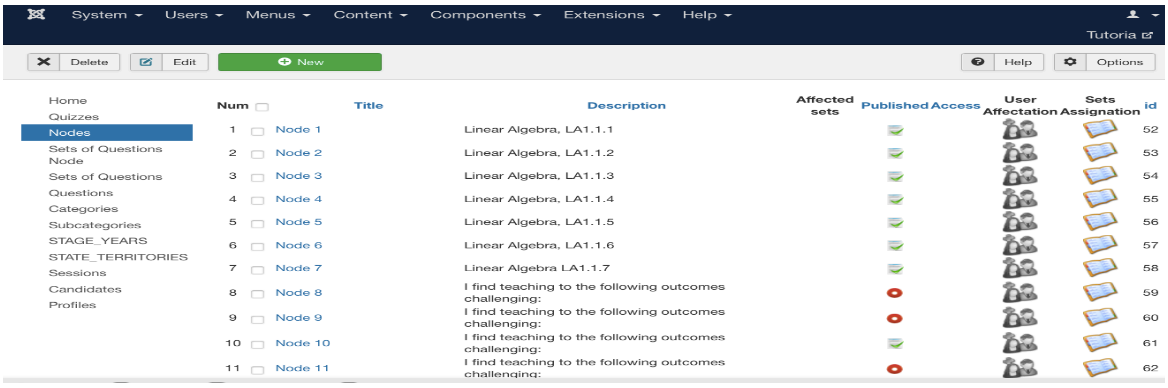


Fig. 6.1 Select node for editing

Node Content Each node serves as a unit of content and interaction, comprising descriptive material, exercises, and optional constraints that encode known misconceptions. Descriptive content is entered via a rich-text editor supporting HTML formatting, allowing authors to introduce new concepts or review prior knowledge. Nodes may be assigned metadata categories and linked tags that help structure content thematically or align it with syllabus standards.

Exercises are created using structured form-based input, with support for multiple input types including numeric fields, multiple-choice selections, checkboxes, and symbolic expressions. Symbolic input is handled through the 'jw-formulaeditor', which enables students to enter expressions in conventional mathematical notation directly into the browser interface. Authors specify the correct answer(s), configure grading tolerance, and may add targeted feedback or hints triggered by specific patterns of input.

Constraints are optionally installed to model common student misconceptions. These are entered as symbolic patterns, matched against student responses to detect conceptual errors. Each constraint is tagged with a pedagogical label and may trigger branching behaviour, such as redirecting the student to a remediation node or issuing scaffolded guidance. The constraint system draws on principles from constraint-based tutoring literature [126], and is directly linked to the Teaching Model. Constraints thus represent an intermediate layer between domain knowledge and pedagogical strategy, providing a mechanism for interpreting student responses with fine granularity.

Node Connections Node connections in Tutoria reflect the system's Teaching Model. Each node may link to one or more successors, depending on the selected operating mode. Connections may be based on correctness, constraint violation, partial understanding, or static sequence position.

In **ITS mode**, successor nodes are conditionally activated based on the outcome of the student's interaction. Authors define links for fully correct responses, partially correct attempts, and specific constraint violations. For each constraint, authors specify which node the student should be redirected to, along with a priority ordering in case of multiple matches. This allows the system to handle nuanced misconceptions with targeted feedback. The control structure for these transitions is visualised within the authoring interface using labelled edges between nodes, which reflect the pedagogical logic embedded in the Teaching Model.

In **Assessment mode**, node transitions follow a predetermined sequence reflecting curriculum structure. Each node connects unconditionally to the next item in the sequence, enabling its use in linear assessments or structured revision pathways. In this mode, feedback is minimal or delayed, and constraints are either inactive or used only for analytics.

The mode toggle—set via a simple switch in the node authoring interface—controls whether the system interprets node transitions adaptively (ITS mode) or statically (assessment mode). This flexibility supports both formative tutoring and summative evaluation within the same content infrastructure. It also allows for hybrid deployment models in which different parts of the curriculum use different instructional strategies.

6.4 Concluding Remarks

The authoring interface in Tutoria encapsulates core elements of the system's domain and Teaching Models, exposing them through an accessible and structured editing environment. It enables both fine-grained control of individual interactions and broader pedagogical planning across the domain graph. By supporting constraint-driven adaptation, symbolic input, and mode-switchable transitions, the authoring system allows for rapid development and flexible revision of intelligent tutoring content.

Unlike traditional content management systems, the Tutoria authoring interface is designed to make pedagogical logic transparent and editable. Authors do not simply manage learning resources—they encode instructional intent. This supports alignment between instructional design and Student Modelling, making it easier to implement, analyse, and revise ITS strategies. Its design reflects a commitment to conceptual clarity, extensibility, and architectural alignment with the broader system.

Chapter 7

Evaluation and Results

The pilot deployment of Tutoria and evaluation of the system was undertaken at UTS. Two components of the testing included differing styles of testing.

LMS-style quizzes are widely used at UTS, not only within individual subjects but also in institutional readiness initiatives such as the Mathematics Readiness Survey (MRS), developed under the Pathways project. The MRS is delivered via multiple-choice questions (MCQs) and administered as a co-enrolment activity for commencing students. It is a compulsory component for all new students in Science, Engineering, and Information Technology who are enrolling in a core, first-stage mathematics subject. Hosted on a purpose-built Canvas site, the MRS assesses foundational mathematical knowledge recommended by UTS for successful participation in these programs. Recommended qualification for such students is equivalence to the N.S.W. Stage 6 curriculum — Extension 1 Mathematics [136]. Students who have this level of qualification generally cope well in the first year offering. However, students without this level are redirected by the Pathway to an intervention subject designed to complete the required level of mathematics education before progressing to the usually planned first year offering.

ITS-style quizzes are new to UTS and as there is a similarity in the user interface, users participating do so without noteworthy comment.

7.1 Overview

This chapter presents the data collected during the pilot study and provides a detailed account of its analysis and the findings derived from it. The narrative connects with the research design set out in Section 4.3.4, which established the framework for comparing the effectiveness of two instructional approaches: the traditional MRS-style online quizzes and the newly developed Tutoria system. While the original research design anticipated a broad participant base, the actual implementation was constrained in scale. Nonetheless, the central research hypothesis remained testable and the results of testing provided some tentative insights through statistical analysis.

7.2 The Data

7.2.1 Planned vs Actual Design

The study was designed to compare the effectiveness of two different mathematics learning systems by assessing student understanding of linear functions. The original plan called for a large-scale deployment, aiming for at least 250 participants completing online quizzes from remote locations.

However, due to unforeseen logistical constraints and scheduling conflicts, the actual implementation was limited to a small cohort of 17 participants in a controlled, on-campus setting. The project adapted to the circumstances and proceeded with modified expectations. See Appendix A.2 for the student recruitment notice used and examples of the test instruments used.

7.2.2 Data Collection Procedure

Students were first exposed to a reading-based instructional segment, followed by a sequence of structured questions targeting linear function concepts. The question set was derived from the MRS-style assessment, covering key learning objectives such as:

- Identifying equivalent forms of linear equations (e.g., determining whether $2x + 3y - 6 = 0$ conforms to the general form $Ax + By + C = 0$),
- Rewriting equations to isolate terms, such as making $3y$ the subject in $2x + 3y - 6 = 0$,
- Determining the gradient from the standard form,
- Solving for the y-intercept.

Participants answered both multiple-choice (MRS-style) and free-response (Tutoria-style) formats. In the Tutoria system, MCQs were replaced with open-ended short-answer fields to better assess constructed understanding (see Appendix A.2.1 for examples).

7.2.3 Factors Affecting Collection

Several internal and external factors influenced the data collection:

- **Internal:**

  The availability of lab space was reduced by an unreported scheduling clash. The overall time for collection was reduced as a result, from three weeks to one.

  The removal of a critical piece of system support architecture, late the previous year, stopped the system from being online and available anytime — from anywhere. The collection was then required to downscale to a single lab over a week outside of the teaching semesters, with attendance required.

- **External:**

  Participant recruitment was limited due to the time of the year and reduced engagement outside standard class times, that is, most students were on holidays.

7.2.4 Summary of the Data

From the 17 participants, each completed a five-question linear function assessment in each format, yielding a total of 85 responses in each format. Each participant experienced both the MRS and Tutoria environments sequentially. Responses were marked against a rubric assessing both correctness and method completeness.

Rubric for Assessment of Linear Function Questions

To evaluate student responses to linear function problems, a two-dimensional rubric was developed assessing both the correctness of the final answer and the completeness of the method used. Table 7.1 outlines the specific criteria used to classify student performance along these two dimensions. The rubric supports comparative analysis between different assessment modes by identifying whether students not only arrived at the correct solution but also allowed for partial progress demonstrating valid mathematical reasoning.

Table 7.1 Rubric: correctness and method completeness

Criterion	**Level**	**Description**
Correctness	Full credit (1)	Correct final answer given; expression is mathematically valid and simplified (if required).
	No credit (0)	Incorrect answer or response not interpretable as a valid mathematical statement.
Method Completeness	Fully complete (1)	All necessary algebraic steps shown or implied; reasoning consistent with standard method.
	Equivalence complete (1)	May be valid but incomplete
	Incomplete (0)	Not valid but also incomplete

MRS Rubric Alignment: As shown in Table 7.2, the MRS assessment model, which uses structured multiple-choice or multiple-response formats, prioritises correctness. Since student working is not visible in such formats, method completeness is not assessable. This limits diagnostic insight but supports scalable benchmarking.

ITS Rubric Alignment: By contrast, the ITS framework enables assessment of both dimensions. Student responses are parsed symbolically, allowing evaluation of correctness and inference of the method used. This supports nuanced feedback and the identification of common misconceptions.

Table 7.2 Alignment of rubric dimensions with assessment modes

Rubric Dimension	**MRS (Multiple Response Structured)**	**ITS (Free-Response / Symbolic)**
Correctness	✓ Central to scoring; based on selected response.	✓ Core feature; symbolic input is parsed and evaluated for equivalence or solution correctness.
Method Completeness	× Not assessable; no working steps visible in structured responses.	✓ Assessable; inferred from response structure or constraint-based analysis. Enables fine-grained feedback.

7.3 Analysis

A standard approach to pilot evaluation was undertaken, although with a reduced sample size ($n = 17$). The data collected consisted of paired test scores from two tests, MRS and Tutoria. The initial goal of analysis was to test the null hypothesis that there was no difference between the means of the two corresponding population distributions of test scores.

To apply a paired-sample t-test, it is necessary to assume that the population distribution of the paired differences is approximately normal. This assumption is commonly checked by evaluating whether the sample of paired differences is adequately described by a normal distribution. However, due to the small sample size, there were insufficient data to robustly assess this assumption. As a result, a t-test could not be reliably applied.

Bootstrapping was adopted as an alternative inferential method. This approach allows estimation of confidence intervals and hypothesis testing under weaker assumptions. Specifically, bootstrapping requires only that:

- The sample is randomly drawn and reflects the population of interest;
- The observations are independent.

Although bootstrapping is agnostic to normality, diagnostic checks remain useful. To test the original hypothesis, a large number of bootstrap samples ($n \geq 1000$) were generated from the distribution of paired differences. For each resample, the mean paired difference was calculated. These means formed a sampling distribution, which approximates the distribution of the sample mean under the null hypothesis.

This approach enabled us to test whether the population mean of the paired differences equals zero. The resulting bootstrap distribution did not support this null hypothesis: the analysis indicated a statistically significant difference between the two population means.

Assumption Checking: Residuals vs Fitted Values

Although the analysis was not regression-based, a standard diagnostic technique was applied to explore model assumptions to demonstrate distribution suitability (normality checking), in particular, plotting fitted values versus their residuals. This provides a visual check for key model assumptions:

Linearity: Residuals should scatter randomly around the horizontal axis. Patterns suggest model misspecification (a structured, non-random pattern in a diagnostic plot (like a residuals vs fitted values plot), then a statistical model might not be capturing the true relationship in the data).

Homoscedasticity: Constant variance is expected. Funnel-shaped residuals suggest heteroscedasticity.

Outliers: Points far from the horizontal axis may indicate outliers or influential cases.

Model Fit and Independence: Curved or structured residual patterns may indicate missing terms or dependency violations.

The residual analysis for this sample violated all of these assumptions, confirming that a parametric regression approach would be inappropriate. This further justified the use of nonparametric methods such as bootstrapping.

7.3.1 Research Hypothesis

The null hypothesis for the study was:

$$H_0 : \mu_{\text{MRS}} = \mu_{\text{Tutoria}}$$

The alternative hypothesis being:

$$H_A : \mu_{\text{MRS}} \neq \mu_{\text{Tutoria}}$$

where μ_{MRS} and μ_{Tutoria} denote the population mean for correctness scores under each system.

7.3.2 Initial Analysis

The mean of the *raw collected data scores* under each system was calculated as follows:

MRS: Sample Mean = 2.588, Std. Dev. = 1.460

Tutoria: Sample Mean = 3.059, Std. Dev. = 1.391

Preliminary observations suggested that students performed slightly better under Tutoria conditions. The raw data, that is, the correctness scores achieved by each student in each test, are listed in Table 7.3 and show some interesting characteristics. The box plots shown in Figure 7.1 have the same median scores, but differing distributions, as illustrated by the

Table 7.3 Collected raw data

#	mrs	tutoria	diff
1	4	5	-1
2	3	4	-1
3	3	3	0
4	0	1	-1
5	4	4	0
6	2	2	0
7	1	2	-1
8	1	1	0
9	3	3	0
10	2	3	-1
11	4	4	0
12	5	5	0
13	3	4	-1
14	2	2	0
15	0	1	-1
16	3	3	0
17	4	5	-1

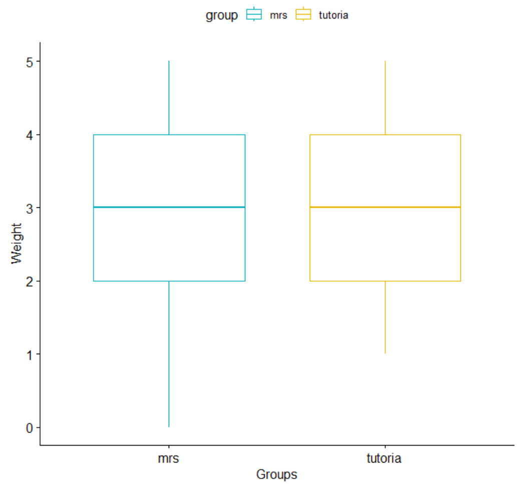


Fig. 7.1 Boxplots of raw collected MRS/Tutoria scores

first quartiles. When examining the differences between paired test scores, several students recorded a difference of zero, indicating that their performance on the second test, delivered through Tutoria, matched their original MRS score. A larger number of students showed negative differences, reflecting improvement on the second attempt. Notably, there were no positive difference scores—no students performed worse on Tutoria than on the MRS. While this asymmetry could be attributed to the small sample size and random variation, it is statistically surprising. In any sufficiently large or representative sample, one might expect at least a few students to perform worse on retesting due to chance, motivation, or testing conditions.

The aim of the analysis is to draw inferences about the population mean of the paired differences, in order to test the research hypothesis that there is a true effect—namely, that performance improves when using Tutoria. However, the small sample size precludes reliable use of a traditional paired *t*-test. This test assumes that the distribution of paired differences in the population is approximately normal, which is not verifiable with so few data points.

The histograms of the correctness scores from the MRS and Tutoria assessments in Fig 7.2) further illustrate the risk of assuming normality. The scores are not only sparse but also discrete, making it unreasonable to treat them as though they were drawn from a continuous normal distribution. The low p-value of the Shapiro-Wilk normality test reported in Fig. 7.3 indicates the distribution is not normally distributed. These considerations necessitated the use of a nonparametric approach, such as bootstrapping, to estimate the sampling distribution of the mean paired differences without relying on strong distributional assumptions.

See Appendix F for the full statistical analysis that was completed, including residuals and in particular the test for normality on the raw data F.1.

```
> # Shapiro-Wilk test.
> shapiro.test(res.my_data)

        Shapiro-Wilk normality test

data:  res.my_data
W = 0.68231, p-value = 7.339e-05
```

Fig. 7.3 Raw data is not normally distributed.

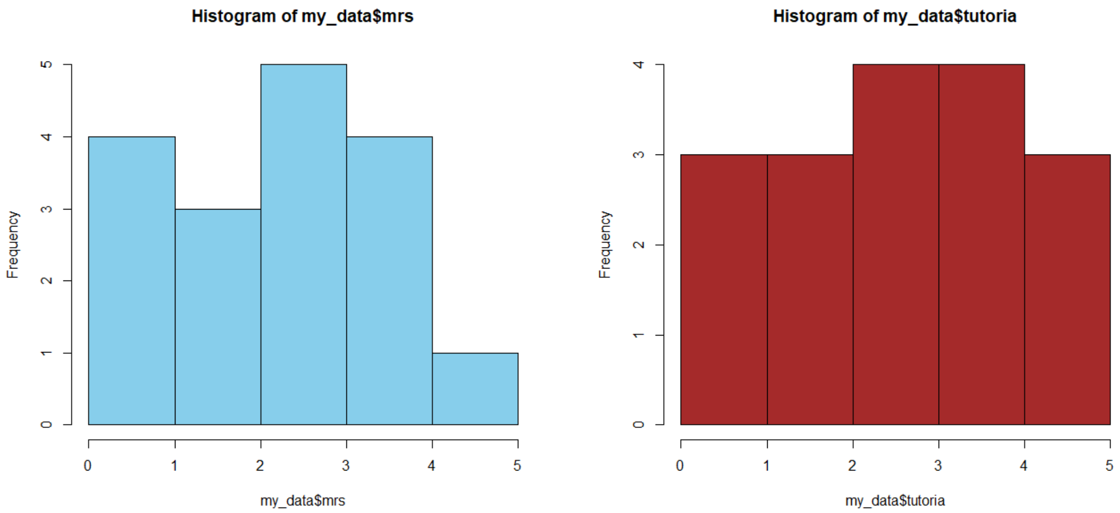


Fig. 7.2 Histograms of correctness scores obtained in the MRS test (left) and Tutoria test (right).

7.3.3 Bootstrapping Analysis

An alternative methodology for drawing inferences on the population distribution is bootstrapping, a non-parametric resampling technique that does not require the assumption of normality. Instead, it relies on two key conditions. First, the original sample must be representative of the population — that is, it should be randomly drawn and sufficiently reflective of the broader distribution from which it comes. Second, the individual observations in the sample should be independent of one another, and each should be drawn from the same underlying distribution. This means that no observation should influence another, and all observations should be generated under consistent conditions. When these assumptions are met, bootstrapping allows for the construction of an empirical approximation to the sampling distribution by repeatedly resampling (with replacement) from the observed data and recalculating the statistic of interest across these resamples.

Although the assumption of normality is not required for bootstrapping, we nevertheless tested for normality of the bootstrapped sampling distribution and found it satisfied — this justifies the use of a t-test on the bootstrapped sampling distributions.

This project used the bootstrapping approach to pursue the data analysis further. Initially, n=10,000 samples were generated. The Shapiro Wilks test used for normality checking of the data set has a restricted maximum n=5000, so some experimentation was done on both sizes of (n) to determine if the lower (n) value could be used. The graphs of the bootstrapped sampling distributions shown in Fig. 7.5 are for the higher (n) value. The eventual use of

```
nboot <- 100000
data_mrs <- matrix(0,ncol =17,nrow = nboot)
data_mrs
for(i in 1:nboot) {
  data_mrs[i,] <- sample(x= mrs, size = length(mrs), replace = T)
}

data_row_mean_mrs <- rowMeans(data_mrs)
```

Fig. 7.4 Bootstrapping MRS raw to get MRS boostrapped means

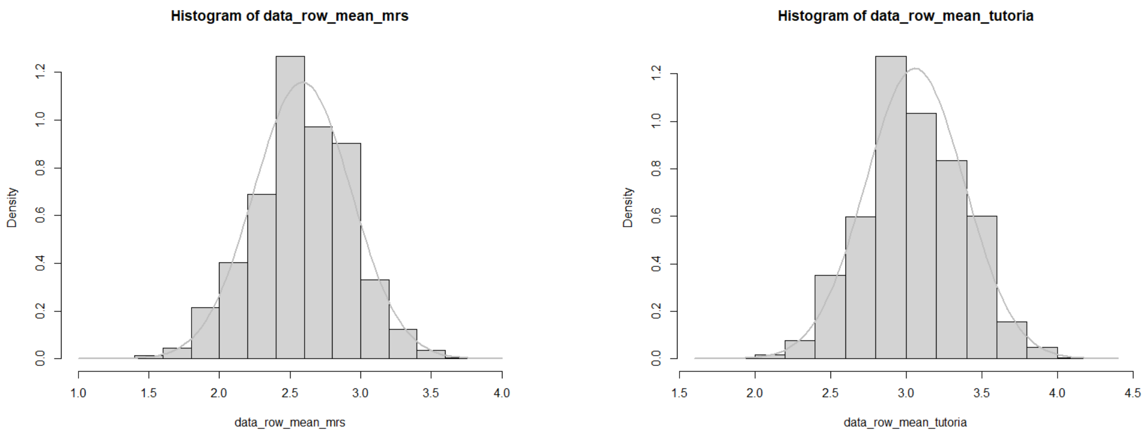


Fig. 7.5 Binned histograms of the means of bootstrap samples from the MRS (left) and Tutoria (right) data.

the lower (n) value for the Shapiro Wilks test showed no appreciable difference in results obtained so the discussion revolves around the lower value. While there are restrictions and qualifications to our conclusions, t-tests done on the bootstrapped data indicated significance.

The R implementation of the bootstrapping procedure used to generate bootstrapped means from the MRS data is shown in Fig. 7.4. The procedure applied to the Tutoria data is similar. The bootstrapping process was first trialed using Mathematica, not as part of the project but as a separate activity using the standard product interface, see Appendix H and also Appendix G. Figure 7.5 shows the results of the bootstrapping process. The figure suggests the underlying normality of the sampling distributions of means. This was confirmed by application of the Shapiro-Wilks normality test, results of which can be found in Appendix F with a complete listing of all the work done, as tested in R-Studio using R code.

Fig 7.6 indicates that the differences of paired means appear to have a very acceptable nearly normal distribution. A t-test was redone, showing significance, in that the null hypothesis is rejected on the basis of three indicators, all of which are in agreement:

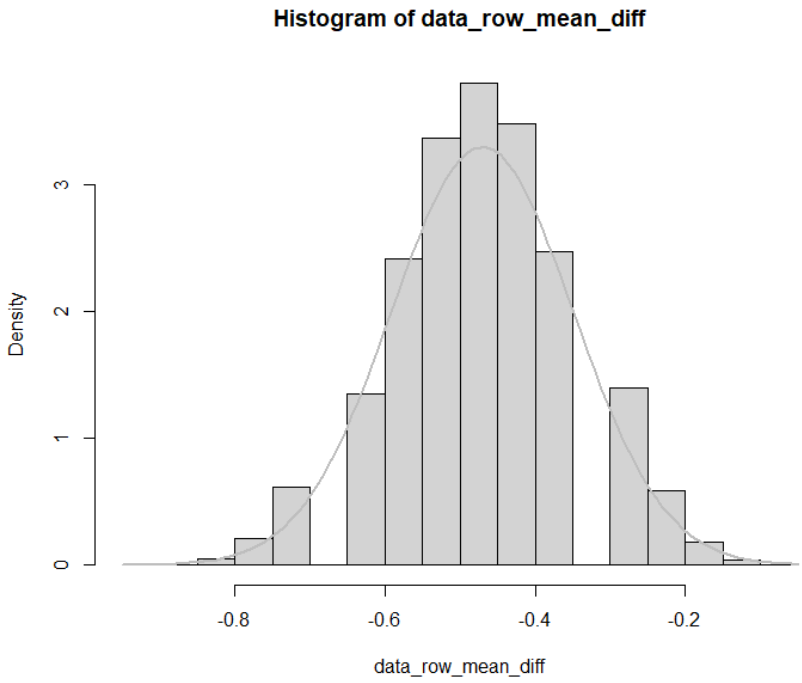


Fig. 7.6 Binned histogram of the differences of paired means of bootstrap samples from the MRS and Tutoria data.

p-value: p-value indicates 0 difference being too unlikely to have occurred,

CI: 0 is outside the confidence interval, and

Rejection region: the t statistic at -68.122 is well into the rejection region.

R-code used in bootstrapped t-test:

```
  ### Compute t-test
    res <- t.test(mrs, tutoria, paired = TRUE)
  ### Compute t-test
    res <- t.test(weight ~ group, data = my_data_)
    t.test(weight ~ group, data = my_data_, alternative = "less")
Welch Two Sample t-test data:  weight by group
      t = -68.122, df = 9983.8, p-value < 2.2e-16
      alternative hypothesis: true difference in means between
      group mrs and group tutoria is less than 0
      95 percent confidence interval:
           -Inf -0.4479964
```

```
sample estimates:  mean in group mrs      mean in group tutoria
                            2.592247               3.051329
```

Further analysis A summary of further investigations appears in Appendix F, Section F.2. Preference is given to the use of the Confidence Interval (CI) as the statistically significant indicator of means differences, rather than a t-test. In applying a t-test to the bootstrapped data and in order for the test to make any sense (in terms of t-tests), the alternative hypothesis has to be changed to one-sided, i.e. "less". This is due to all of the difference data being below zero. A paired t-test was conducted, with

$$t = -68.122, \quad df = 9983.8, \quad p < 2.2e - 16.$$

Although the t-test is suggesting a statistically significant difference in favour of Tutoria and the bootstrapped 95% confidence interval for the difference ($\mu_{\text{Tutoria}} - \mu_{\text{MRS}}$) excludes zero ($-\infty, -0.4479964$), this analysis is not in keeping with the intention of the original hypothesis. Instead, in keeping with intention of the initial hypothesis, the confidence interval is the preferred measure of significance (statistical). The summary in Appendix F Section F.2 gives justification to this choice. As the CI does not contain zero, the NULL hypothesis given in this thesis, in particular in Section 7.3.1, is rejected. Based on this, and keeping in mind the limitations of the approach, there is a statistically significant difference between the two groups as tested.

7.3.4 Results of Analysis

The bootstrapped difference of means CI, indicates significance, so we reject the null hypothesis. The two group means are not equal. These conclusions are subject to qualification by the following observations:

- small sample size:
  - the nature of 0 appearing in the difference data may not be random;
  - the nature of no negatives appearing in the difference data may not be random;
- bootstrapping:
  - end zones of the sampling distributions are unreliable, so this was mitigated by considering 2 standard deviations only;

- comparative study constraint of repeated material.

Conclusions: This discussion takes into account the qualified nature of the data and tentatively suggests that Tutoria is favourably different to MRS style testing. Two main unexpected events impacted the data collection and its analysis. First, the change in infrastructure at a critical moment in the experiment's cycle meant there were delays in collecting and processing data. Secondly, the reduced sample size challenged the original intention of the research design, which had to be modified to accommodate the new circumstances.

Bootstrapping is a widely used technique [74, 113] when dealing with small sample sizes. At the $\alpha = 0.05$ level, our results leads to rejection of the null hypothesis. Nevertheless, although bootstrapping is widely used, in this study the results must be treated with caution due to the particularly small original sample size of the raw data.

7.4 Concluding Remarks

Despite constraints in sample size and setting, the analysis tentatively supports the conclusion that Tutoria may offer a pedagogical advantage over learning environments that do not incorporate ITS-style support for symbolic reasoning and formative feedback. Unlike traditional assessment tools such as MRS-style quizzes—which are primarily diagnostic—Tutoria provides interactive, model-driven scaffolding that supports conceptual understanding during the learning process.

While these findings are promising, they should be interpreted with caution due to the limited scope of the study. A larger-scale trial is recommended to validate these trends and investigate the longer-term effects on student learning outcomes.

Chapter 8

Conclusion

8.1 Overview: Development and Evaluation of the Tutoria Pilot

This project aimed to design, implement, and evaluate an intelligent tutoring system (ITS) to support students transitioning into tertiary mathematics. The following key aims structured the research, with each addressed through targeted system features and evaluated via pilot deployment.

1. **Enhance recognition of symbolic equivalence.** A primary goal was to support students in understanding symbolic equivalence—an often under-emphasised yet critical concept in early university mathematics. To this end, the system incorporated Mathematica as a back-end engine for verifying symbolic equivalence across diverse forms of input. This approach allowed the tutor to accept a range of correct answers, even if algebraically or structurally distinct. Evaluation data suggest this capability improved both the pedagogical flexibility of the system and student engagement with algebraic manipulation.

2. **Extend ITS functionality to tertiary-transition mathematics.** Many existing ITSs focus on procedural skills at the secondary level. This project explored how ITSs might be adapted for conceptual topics encountered in early university mathematics, such as interpreting and transforming between graphical, symbolic, and verbal forms of linear functions. While the mathematical domain remains relatively elementary, the project targeted conceptual complexity and representation fluency, which are known challenges during the transition to university-level study.

3. **Integrate pedagogical theory with symbolic computation.** The design process was guided by constructivist and schema-oriented pedagogical frameworks (e.g., APOS theory), mapped onto system features including feedback types, question scaffolding, and student modelling. These were implemented in concert with symbolic computation tools, aiming to preserve the precision of formal mathematics while offering pedagogically meaningful feedback. This alignment of instructional design with computational affordances was a core design principle throughout.

4. **Improve models of student understanding in early university contexts.** The ITS was designed to model not only student performance but also potential conceptual and epistemological obstacles. For example, diagnostic feedback was sensitive to patterns of error that might reflect misunderstandings of equivalence, graph-symbol translation, or variable interpretation. Pilot results offered preliminary support for the utility of this modelling approach, although further validation is required.

5. **Support institutional aims around readiness and retention.** The final aim was to ensure the system complemented existing institutional programs supporting student preparedness (e.g., bridging or transition courses). By embedding Tutoria into an existing university-run transition initiative, the project was able to explore how interactive, automated tools might be used to scale support, track student misconceptions, and offer targeted interventions. Initial findings indicate that such systems can enhance the capacity of readiness programs to meet diverse learner needs.

8.2 Future Work

During the course of this project, several opportunities for further development and investigation were identified. These directions reflect both the immediate extensions of the current system and broader research pathways emerging from the intersection of pedagogy, AI, and mathematics education. Key avenues for future work include the following:

Hybrid Model: Combining Model Tracing and CBM As discussed in Section 4.2.2, the current implementation employs a computational engine to verify symbolic equivalence and guide the learner through a concept graph. This initial structure lays the groundwork for integrating a full Constraint-Based Model (CBM). Future work could expand this foundation to implement a hybrid model that combines CBM with traditional model tracing, enabling richer representations of student progression and finer-grained adaptation in instructional pathways.

Expansion of Affective Tools Incorporating affective components into the student model presents a promising direction. Parameters such as expressed confidence, persistence, or frustration signals could inform more adaptive instructional decisions. By dynamically interrogating these features, the system could make nuanced pedagogical choices, for example adjusting problem difficulty or offering motivational feedback. Such research would lie at the intersection of AI, educational psychology, and real-time student modelling.

Social Interaction and Activity-Oriented Extensions The integration of social learning frameworks, particularly activity theory, opens up potential for scaffolding student engagement through peer interaction and collaborative inquiry. A future system might include shared workspaces, discussion-linked feedback, or real-time co-construction of mathematical objects. These would be designed in alignment with best practices in accessibility and universal design, ensuring that interactive features also support learners with diverse needs.

Larger and Longitudinal Trials While this pilot provided initial insights into the system's efficacy, further validation would require broader deployment. A future study might involve a larger sample size, potentially across multiple institutions, and ideally track students longitudinally to assess sustained learning gains, retention, and transfer. Such work would necessitate institutional collaboration and substantial resources, but would offer stronger empirical support for the system's effectiveness.

LLM–Symbolic Integration While large language models (LLMs) such as GPT-4 excel in natural language generation, they lack the deterministic symbolic verification critical for mathematical tasks. Recent research has explored hybrid approaches combining LLMs with symbolic solvers—such as Logic-LM and SymbCoT frameworks [141, 201]—which enhance logical reasoning by aligning generative capabilities with formal symbolic engines. Future versions of Tutoria could incorporate such methods, allowing richer natural language interaction while maintaining mathematical precision and rigour.

Qualitative Evaluation and Thematic Analysis In the current study, qualitative methods were not fully implemented. However, future iterations could incorporate structured student interviews, open-ended survey responses, and Likert-based self-efficacy measures [171] to assess user perceptions and attitudes. Using thematic analysis [135], this data would provide insight into how learners experience the system, informing both evaluation and iterative design.

Appendix A

Ethics and Data collection

A.1 Ethics Approval

UTS

WORKFLOW NO. 22283

UTS Internal

UTS MEMORANDUM

UTS FACULTY OF SCIENCE: SCHOOL OF MATHEMATICAL AND PHYSICAL SCIENCES

TO:	PROFESSOR JAMES WALLMAN PROFESSOR STEVEN LANGFORD	DATE:	24/11/2023
ACTION:	FOR APPROVAL		
FROM:	JASON STANLEY		
SUBJECT:	WRITTEN CONSENT FROM DEAN – ETHICS FORM		

Dear Prof James Wallman,

I am seeking approval regarding an ethics application for the completion of my PhD project "*Intelligent Tutoring System in the Mathematical domain*" which will be conducted online on over the Summer session, here at the, University of Technology Sydney. Since the project involves student members from the faculty, I will need your approval.

The project involves the recruitment and participation (voluntary) of Science students undertaking Mathematics 1 and the MRS over the summer session. My PhD supervisor (Professor Christopher Poulton [primary]) is aware and supportive of my undertaking of this research, and we have been in close consultation regarding the administrative requirements to ensure the safety, privacy and wellbeing of those involved. Staff and students who choose to participate will be asked to use a simple web-based assessment application and then an ITS based web-based assessment application and complete a short online feedback questionnaire. All aspects of the study are online and can be timed to suit each participant so as to minimise disruption to their work and study commitments.

Please find attached the ethics application for your information.

Kind regards,

Production Note:
Signature removed prior to publication.

Production Note:
Signature removed prior to publication.

27 November 2023

Endorsed by: …………………………

Jason Stanley

(Steven Langford - Head of School, MaPS)

Production Note:
Signature removed prior to publication.

Approved by Dean: ……………………………………

A.2 Data Collection

Invitation to participate in Pilot evaluation study Student information was fully redacted for this student research activity. No identifying information was kept. The collection material is provided here.

Invitation to participate

Interested in Research?
Jason Stanley is inviting you to join in and experience real research!
Just drop in!

Where: attend CB04.03.331
What time: after 2:00PM on
When: Tuesday, Wednesday and Thursday for the first three weeks of January 2024.
Yes! He is the MRS dude!

Jason is also doing research in Intelligent Tutoring Systems,
ITS for mathematics...

So if you can spare 40-50 minutes and would like to see the difference
between a traditional style LMS and an ITS come along!

Completely voluntary, you are under no obligation whatsoever to participate...

Jason

Opening Tutoria page: by proceeding from this page, submitting below, you agree to participate in the research project:

Intelligent Tutoring System (ITS) for Mathematics Using Computational Engines

being conducted by Jason Stanley, UTS School of Mathematical and Physical Sciences,
telephone number 9514 2273 at the University of Technology Sydney.
I understand that the purpose of this study is to find out how students' opinions about Tutoria, compared with a traditional LMS system while learning mathematics, and undertaking content quizzes. The study is part of the research that Mr Stanley is undertaking towards a PhD at the University of Technology Sydney. His supervisor at UTS is Professor Dr Christopher Poulton. I understand that my participation in this research will involve about two twenty minutes occasions while attending CB04.03.331 to complete a post session survey, and that this participation is entirely voluntary. I am aware that if I agree to participate my assessment results in mathematics will be accessed by the researcher in a form that does not identify me by name, and used as data in the study. I am aware that I can contact Mr Stanley if I have any concerns about the research. I also understand that I am free to withdraw my participation from this research project at any time I wish and without giving a reason. My decision to withdraw will not affect my academic results. I agree that the research data gathered from this project may be published. I agree to this on the condition that the outcomes are expressed in a form that does not identify me in any way.

Submit - continue	Cancel
∘	∘

NOTE: The Ethics approval for this study has been provided by the Dean of Science for completion over the 2023/2024 Summer. If you have any complaints or reservations about any aspect of your participation in this research you may contact the Deans office at the Faculty of Science through the Research Ethics Officer. Any complaint you make will be treated in confidence and investigated fully and you will be informed of the outcome.

A summary sample of the testing used.

A.2.1 Testing

MRS

- Using an MRS style system, users were given five linear function questions, after reading, asking to identify:
  - equivalent forms: an understanding of co-efficients,
    - * Is the following equation a correct example of an application of the general form, Ax+By+C=0, for a linear equation?
      2x+3y-6=0.
      ∘ T, ∘ F
  - re-arrange for particular co-efficients
    - * Choose the correct re-arrangement for the term 3y the subject in 2x+3y-6=0
      ∘ y = 6-2x ∘ 3y = 6-2x ∘ -3y = -6+2x
      ∘ y = -3+2x ∘ I have not seen this before
  - solve for gradient
    - * Choose the correct re-arrangement for the gradient as the subject in 2x+3y-6=0
      ∘ 2x = -3y+6 ∘ $x = -\frac{3}{2}y + 3$ ∘ x = 6-2y
      ∘ x = -3-2y ∘ I have not seen this before
  - solve for y-intercept, and so on and so forth...
    ⋮

ITS Notice MCQ questions no longer being used.

- Is the following equation a correct example of an application of the general form, Ax+By+C=0, for a linear equation?
  2x+3y-6=0 ans. ____________
- Make the term 3y the subject in 2x+3y-6=0,
  ans. ____________
- Find the gradient of 2x+3y-6=0,
  ans. ____________
- solve for y-intercept, and so on and so forth...
  ⋮

A.2.2 Curricula Interface

Figure A.1 depicts the student entry point to the system. The links on the right menu item list, among the entries, 'Nodes' and 'Tutoria'. In actuality, both links are entry points to the Tutoria ITS.

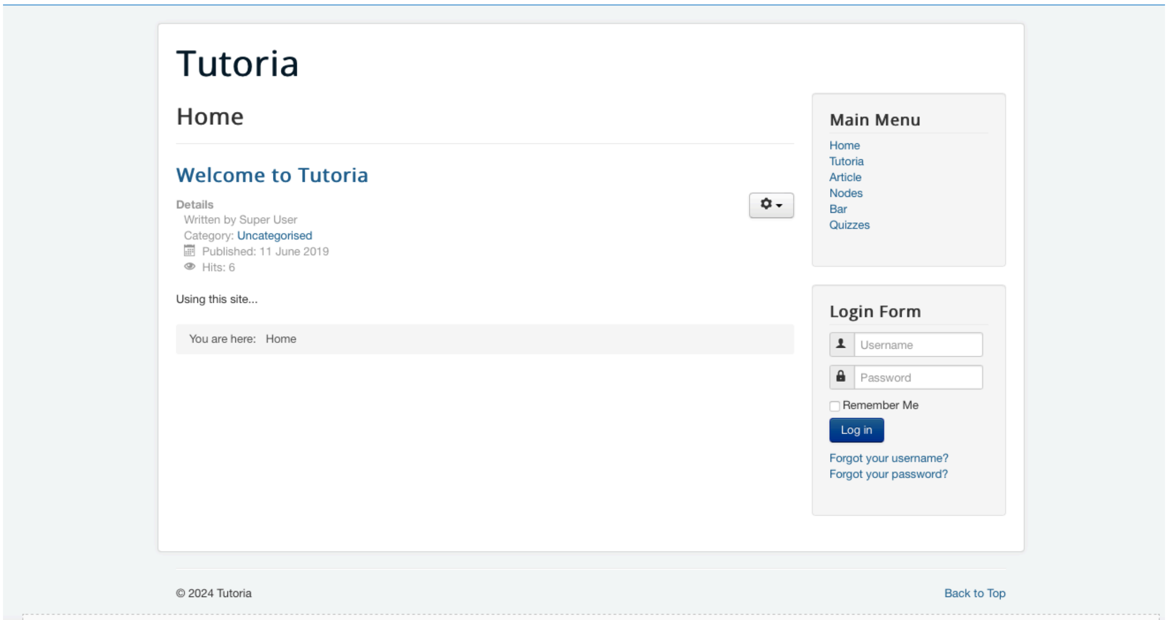


Fig. A.1 User entry points to Tutoria

The 'Tutoria' link follows the National Curriculum, or the New South Wales (N.S.W) curricula, as options, and the other ('Nodes') has access only to the New South Wales

(N.S.W) curricula. These versions are defined, designed and implemented according to the expertise of the content expert, acting as author.

Nodes In the entry point example named Nodes, the navigation of the Domain Model rests with an author's design. Selection of a node by a student loads a new page which has a question from the Domain Model. Figure A.2 shows a list of nodes available for selection, based on questions about linear functions, specifically, rearranging terms.

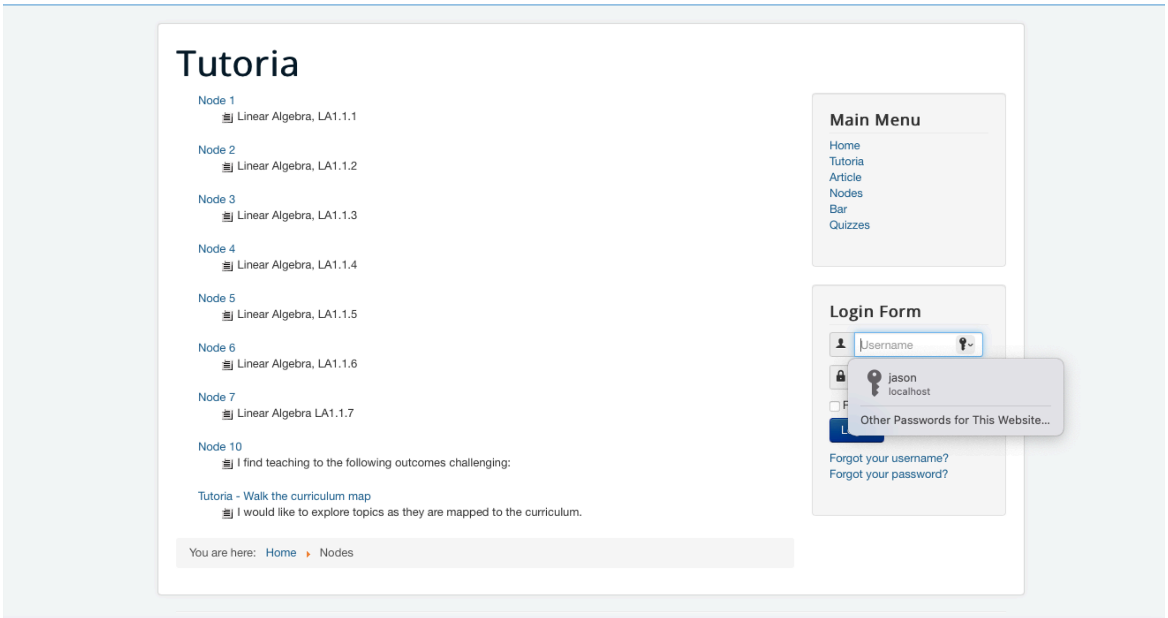


Fig. A.2 Nodes site

Such pages can be developed rapidly using the authoring subsystem described in Chapter 6 — allowing the the incremental development of a flexible and extensible Domain Model.

Tutoria The Tutoria version of Domain Model entry point, represents a new approach to mathematics education. A new approach in that the curriculum as Domain Model becomes front and center of a mathematics education. The introduction page is simple. This design makes the Domain Model visible and central, reinforcing the system's emphasis on transparency and coherence. The introduction page reflects this by presenting a clear and minimal starting point.

The NSW curriculum is organised into a variety of topic strands, themselves comprising a variety of units, each of which is studied over a number of stages within the curriculum.

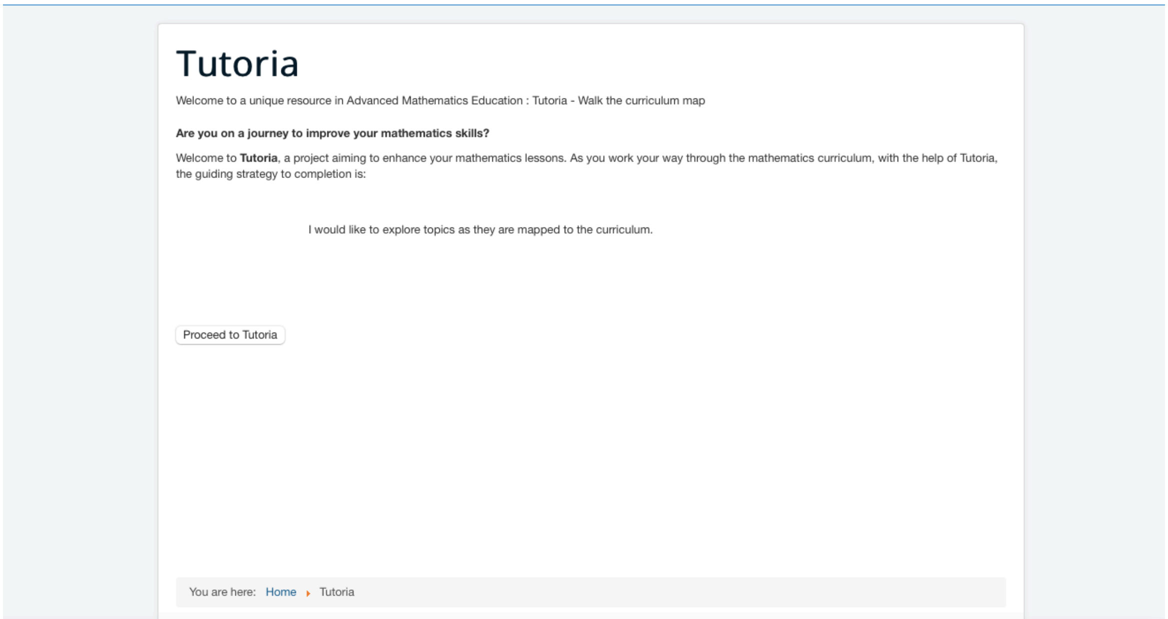


Fig. A.3 Enter Tutoria node map walking

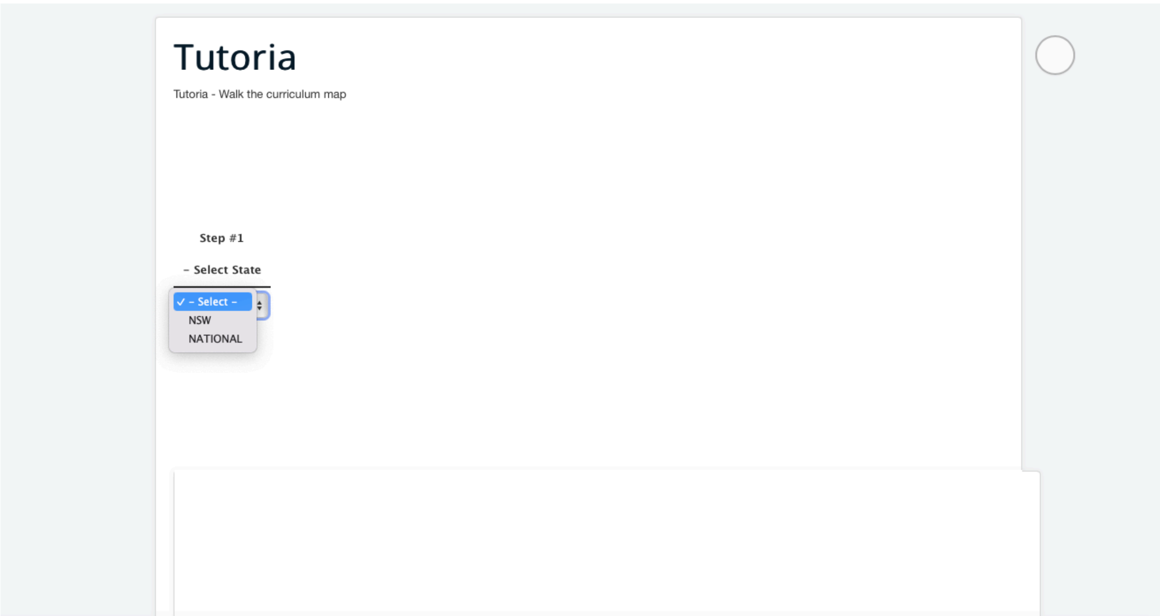


Fig. A.4 Choosing curricula

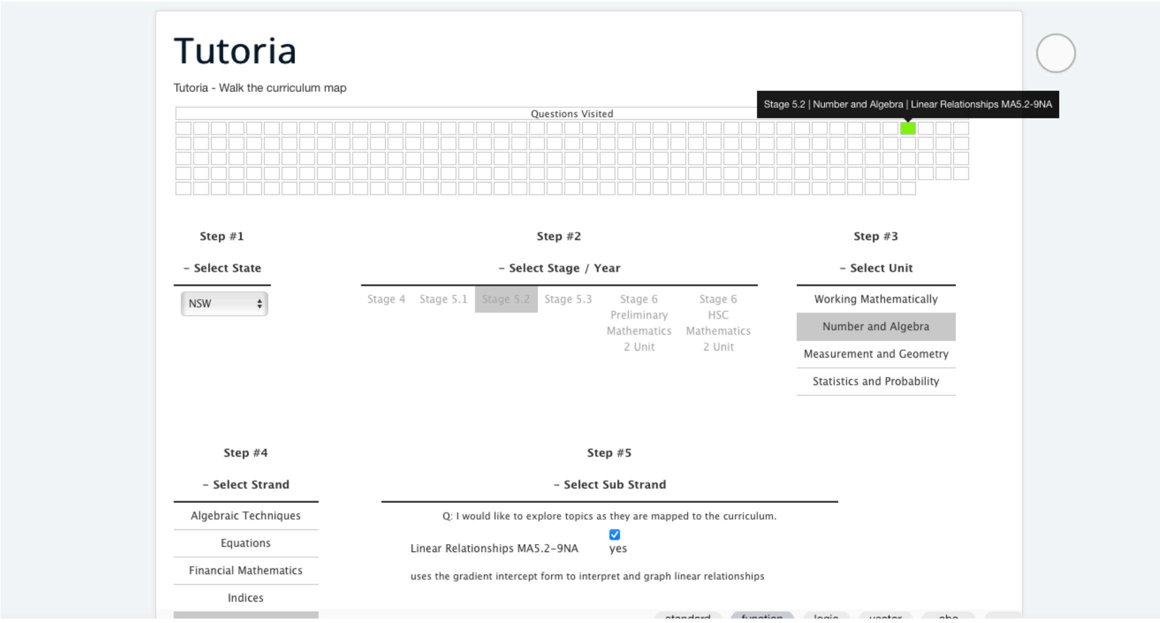


Fig. A.5 Curriculum table

Students can choose to access directly any point of this curriculum structure, as illustrated in Fig. A.5.

Having chosen their entry point, that is, their entry node, students are presented with an exercise on the relevant curriculum element.

Fig. A.9 show a condensed table, which when hovered over gives an indication of the particular curriculum point, it's details including stage. Further after completing a question associated with a stage the corresponding table item is coloured green to show the user has been there.

Choosing the NSW option and then selecting the Stage 5.2 link. By clicking the yes box two main things happen. The table previously mentioned appears, as does a question taken from the topic. The following two figures are actually one page divided for the publication. The tools displayed have since been superseded and were available at the time of experiment, Fig. A.11, as they were Open Source and freely available under a General Public License, GNU.

The novelty of Tutoria is on display. The sum used is commonly known and converges to the number 2. It just so happens the gradient being sort in the question is also the number 2. Mathematica affirms the correctness of this entry demonstrating the power of it's computational engine. Of the many systems of Computer Algebra System, CAS, and vehicles of testing mathematics on a computer, from numerous book publishing companies, none have come close to this level of symbolic use.

Fig. A.6 Choosing stage

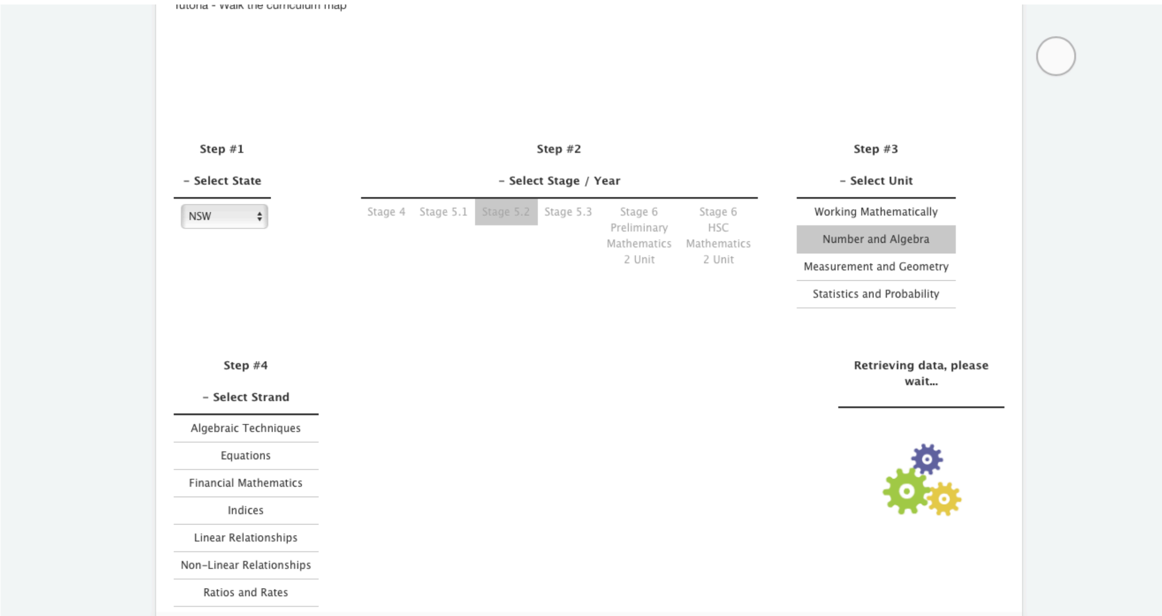


Fig. A.7 Choose curriculum topic

Fig. A.8 Confirming question

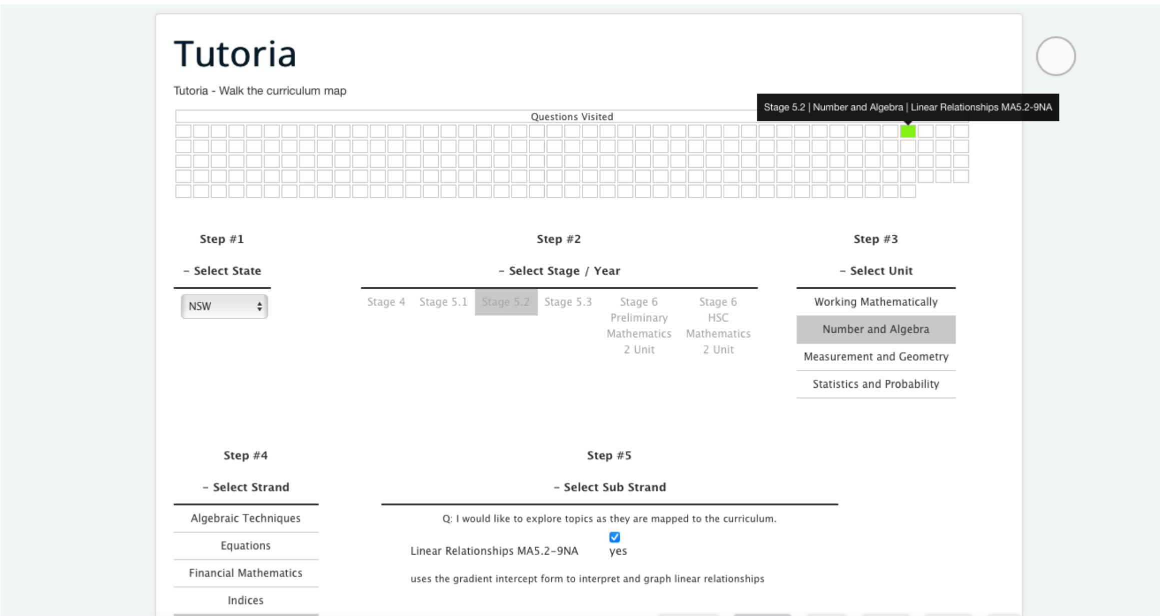


Fig. A.9 Curricula table

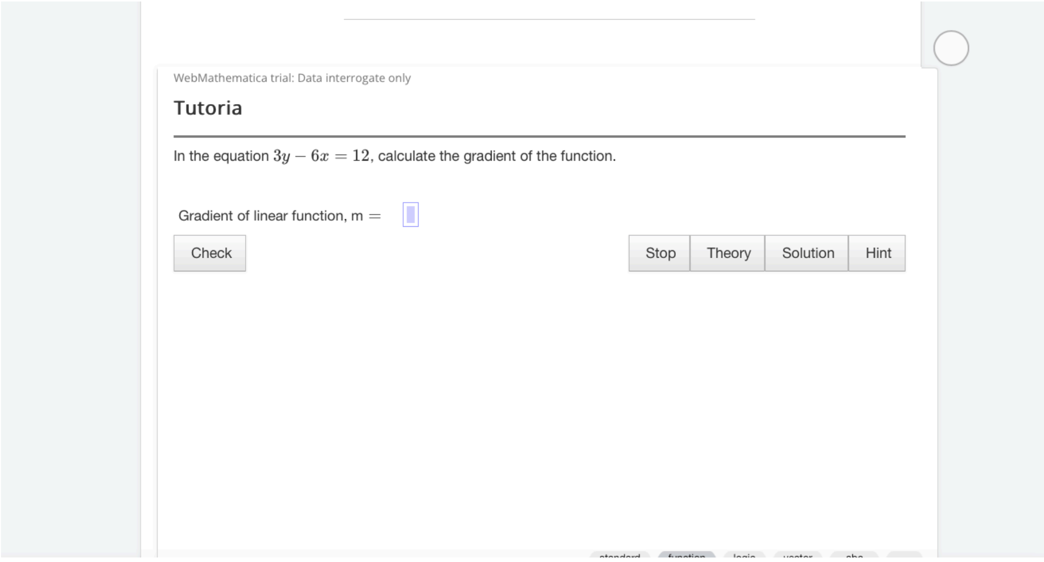


Fig. A.10 Another simple question

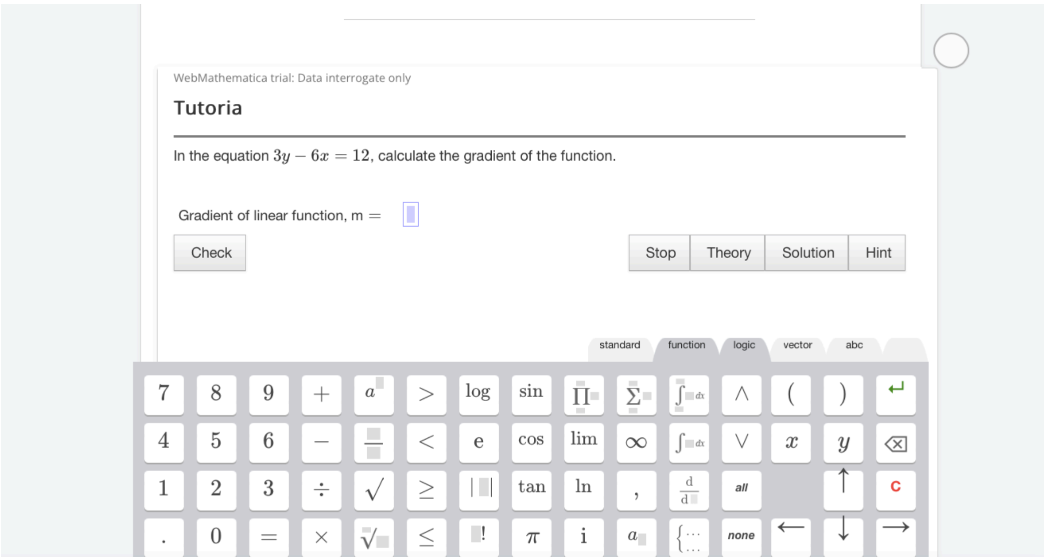


Fig. A.11 Some tools to help

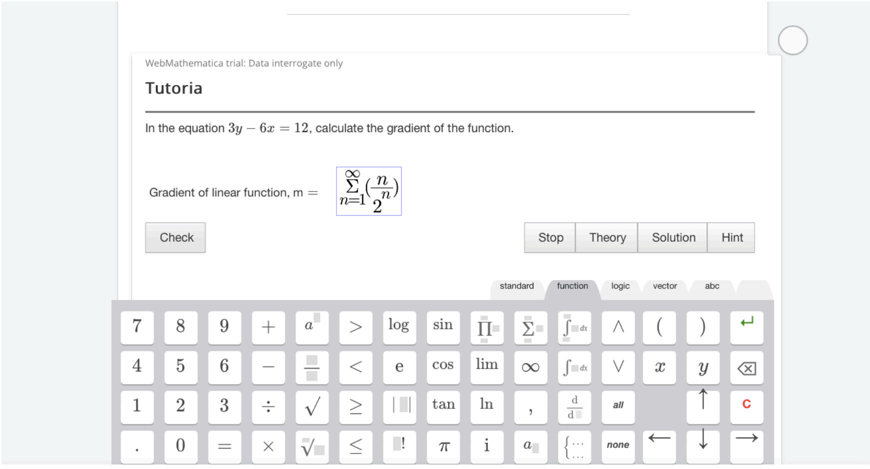


Fig. A.12 A novel system

A.2.3 Apache Server Configuration

Configuration of the Apache server enables interaction with the Apache-Tomcat server using a well known Multi-processing Modules (MPM) library. MPM workers are set or configured in both servers, facilitating silent asynchronous communication between them. That is, calls to the Apache-Tomcat server hosting Mathematica are made as normal procedural calls. Configuration files are located in Appendix D. A further configuration requires defining workers, and workers are also defined in the Apache-Tomcat server configuration files. The addition of the required module can also be found in the Appendix.

Apache-Tomcat server Configuration of the Apache-Tomcat server enables the receiving connector to respond and initiate with the Apache workers. A series of XML files achieves configuration, and will not work without correct settings.

Context This configuration monitors resources. Typically these resources are files, once edited Tutoria will cause the web application will be reloaded.

```
<WatchedResource>WEB-INF/web.xml</WatchedResource>
<WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
<WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
```

These particular files are themselves Apache-Tomcat system configuration files. Some are discussed below.

Tomcat-users Security: Apache-Tomcat is a secure system and requires registration of users and their passwords, however, this is at an administrative server level. As administration can be done through both a generic web interface and a function call, details of security must be configured. Calls made to Mathematica Scripts or MSP rountines use these secure user definitions as server access to client calls only. As such individuals are masked as client side requests to the server and so individual users do not need access, as the Apache HTML/PHP server is using these credentials on behalf of clients.

Server Returning to configuration, the server configuration opens ports and establishes pools and monitors threads as well as maintaining the all important AJP connector, which facilitates inter server communication.

```
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
<Resource name="UserDatabase" auth="Container"
    type="org.apache.catalina.UserDatabase" description="User database that can be updated and sa
    factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
    pathname="conf/tomcat-users.xml" />
    <!-- A "Connector" represents an endpoint by which requests are received and responses are re
    <Connector port="8080" protocol="HTTP/1.1"
               URIEncoding="UTF-8"
               connectionTimeout="20000"
               redirectPort="8443" />
```

Monitoring and management to pools is critical when large numbers of clients are using the system and is managed through the management web interface.

Web Configuration files are located in Appendix E.
This file represents the 'servlets' used for the web based server outputs. There are two basic ways to program the java server, one way as mentioned, is to use servlets, the other is to use scripts, script files have a .jsp file suffix. The preferred method is to invoke servlets because there is more control over what happens in runtime. Servlets are considered resources and as such have more features attached to them in the form of attributes that can be edited and maintained. Debugging servlets is aided by the parameters organised in the web.xml file.

Further MIME file types can be registered in this file so the Java server 'knows' what to do with when encountered. A typical format:

```
<servlet>
  <servlet-name>default</servlet-name>
    <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
      <init-param>
        <param-name>debug</param-name>
        <param-value>11</param-value>
      </init-param>
      <init-param>
        <param-name>listings</param-name>
        <param-value>true</param-value>
      </init-param>
      <load-on-startup>1</load-on-startup>
  </servlet>
```

Appendix B

Authoring Administration Scope and Code

node.php list nodes for selection to edit.

```
<?php defined('_JEXEC') or die('Restricted access'); ?>
<script type="text/javascript" language="javascript">
  function submitAssignUsers(id) {
    nodeId = document.getElementsByName('cid[]') ;
^^InodeId = nodeId[nodeId.length - 1];
^^InodeId.value = id ;
    document.adminForm.task.value = 'assignUsers' ;
    document.adminForm.submit() ;
^^I}
  function submitAssignSets(id) {
    nodeId = document.getElementsByName('cid[]') ;
    nodeId = nodeId[nodeId.length - 1];
    nodeId.value = id ;
    document.adminForm.task.value = 'assignSets' ;
    document.adminForm.submit() ;
^^I}
</script>
<?php //JLog::add(JText::_(INconnectHelper::indentjson(json_encode($this->nodes)).
//'= $this->nodes line 29 inconnect elements nodeelement '), JLog::WARNING,
//'com_inconnect'); ?>
<form action="index.php" method="post" name="adminForm" id="adminForm">
  <div id="editcell">
    <table class="adminlist">
    <thead>
^^I   <tr>
```

```
        <th width="20">
<?php echo JText::_( 'NUM' ); ?></th>
^^I^^I<th width="20">
<?php echo JHtml::_('grid.checkall', 'checkall-toggle', count( $this->nodes )); ?></th>
        <th width="20%">
<?php echo JHTML::_( 'grid.sort', 'TITLE', 'TITLE',
$this->lists['order_Dir'],$this->lists['order'] );  ?></th>^^I^^I^^I
        <th>
<?php echo JHTML::_( 'grid.sort', 'DESCRIPTION', 'description', $this->lists['order_Dir'],
$this->lists['order'] );  ?></th>
        <th>
<?php echo JText::_('AFFECTED_SETS') ; ?></th>
        <th width="5%">
<?php echo JHTML::_( 'grid.sort', 'PUBLISHED', 'published', $this->lists['order_Dir'],
$this->lists['order'] );  ?></th>
        <th width="5%">
<?php echo JHTML::_( 'grid.sort', 'ACCESS', 'groupname', $this->lists['order_Dir'],
$this->lists['order'] );  ?></th>
        <th width="5%">
<?php echo JText::_('USER_AFFECTATION') ;  ?></th>
        <th width="5%">
<?php echo JText::_('SETS_ASSIGNATION') ; ?></th>
        <th width="5">
<?php echo JHTML::_( 'grid.sort', 'ID', 'id', $this->lists['order_Dir'],
$this->lists['order'] );  ?></th>
      </tr>
    </thead>^^I
<tbody>
<?php
if (count( $this->nodes )) :
$k = 0;
for ($i = 0, $n = count( $this->nodes ) ; $i < $n ; $i++) :
$row =& $this->nodes[$i];
$checked = JHTML::_( 'grid.id', $i, $row->id );
$link = JRoute::_( 'index.php?option=com_inconnect&controller=node&task=edit&cid[]='.
$row->id ) ;
$linkPublish = JHTML::_('grid.published', $row, $i, 'publish_g.png', 'publish_x.png' );?>
      <tr class="<?php echo "row$k"; ?>">
        <td align="center">
<?php echo $i+1; ?></td>
        <td>
<?php echo $checked; ?></td>^^I^^I^^I
```

```
        <td>
<a href="<?php echo $link ; ?>"><?php echo $row->title; ?></a></td>^^I^^I^^I
        <td>
<?php echo $row->description; ?></td>
        <td><?php if (is_array($this->affectedSets[$row->id]) ||
        is_object($this->affectedSets[$row->id])) {
  foreach ( $this->affectedSets[$row->id] AS $set) :
$linkSet = JRoute::_( 'index.php?option=com_inconnect&controller=setofquestionsnode&
task=edit&cid[]='. $set->id );
echo '<a href=' . $linkSet . ' title=' . $set->title . '>' . $set->title . '</a><br />' ;
  endforeach ; } ?> </td>
        <td align="center">
<?php echo $linkPublish ; ?></td>
        <td> <?php // to improve try using grid.access later^^I
if ( !$row->access_id )  :
$color_access = 'style="color: green;"';
elseif ( $row->access_id == 1 ) :
$color_access = 'style="color: red;"';
else :
$color_access = 'style="color: black;"';^^I^^I^^I^^I^^I^^I
endif ;^^I?>
<span <?php echo $color_access ;?>><?php echo $row->groupname ;?></span>
        </td>
        <td align="center">
<?php if ($row->access_id == 1) : ?>
<a href="javascript:void(0);" onclick="submitAssignUsers(<?php echo $row->id ; ?>)">
<img src="/Joomla_3/images/groups_f2.png" /></a>
<?php else : ?>
<img src="/Joomla_3/images/groups.png" title="<?php JText::_('PUBLIC_QUIZ') ?>"
alt="<?php JText::_('PUBLIC_QUIZ') ?>" />^^I
<?php endif ; ?></td>
        <td align="center">
<a href="javascript:void(0);" onclick="submitAssignSets(<?php echo $row->id ; ?>)">
<img src="/Joomla_3/images/address_f2.png" /></a> </td><?php
//JLog::add(JText::_(INconnectHelper::indentjson(json_encode($row->id)).'
// = $row->id line 128 inconnect nodes default '), JLog::WARNING, 'com_inconnect');?>
        <td>
<?php echo $row->id; ?></td></tr><?php
$k = 1 - $k;
endfor ;
else :
echo '<tr><td colspan="10">' . JText::_('THERE_ARE_NO_QUIZZES') .
```

```
<br/><br />' . JText::_('MIN_TUTO_QUIZZES') . '</td></tr>' ;
 endif ;?></tbody><tfoot>
        <tr>
          <td colspan="10">
<?php echo $this->pageNav->getListFooter() ; ?></td></tr></tfoot></table>^^I
  </div>
<input type="hidden" name="option" value="com_inconnect" />
<input type="hidden" name="task" value="" />
<input type="hidden" name="cid[]" value="" />
<input type="hidden" name="boxchecked" value="0" />
<input type="hidden" name="controller" value="nodes" />
<input type="hidden" name="filter_order" value="<?php echo $this->lists['order']; ?>" />
<input type="hidden" name="filter_order_Dir" value="<?php echo
 $this->lists['order_Dir']; ?>" />
<?php echo JHTML::_( 'form.token' ); ?>
</form
```

Once a node has been selected it presents for editing.

```
<?php defined('_JEXEC') or die('Restricted access');
//dump($this->has_failure_child, '$has_failure_child: ');
//dump($this->has_success_child, '$has_success_child: ');
//dump($this->node, '$this->node: ');
//dump($this->nodeedges, '$this->nodeedges: ');
//dump($this->node, '$this->node: ');?>
<script language="javascript" type="text/javascript">
  window.onload = function () {
    function checkDate(publish) {
      dateStr = new String() ; dateStr = publish.value ;
      year = dateStr.slice(0,4) ;  month = dateStr.slice(5,7) ;
      month -= 1 ;  day = dateStr.slice(8,10) ;
      time = '' ;   time = publish.value.slice(10) ;
      if (!year || !month || !day) { date = new Date() ; }
      else {   date = new Date(year, month, day) ; }
      // building the date
      month = date.getMonth() + 1 ; day = date.getDate() ;
      if (month < 10) { month = '0' + month ; }
      if (day < 10) {   day = '0' + day ;      }
      publish.value = date.getFullYear() + '-' + month + '-' + day ;
      // adding time
      if (time) { hours = time.slice(1,3) ; minutes = time.slice(4,6) ;
        seconds = time.slice(7,9) ;
        if (hours < 0 || hours > 23) { hours = minutes = seconds = '00' ; }
        if (minutes < 0 || minutes > 59) { hours = minutes = seconds = '00' ; }
```

```
      if (seconds < 0 || seconds > 59) { hours = minutes = seconds = '00' ; }
      t = hours + ':' + minutes + ':' + seconds ; publish.value += ' ' + t ;
    } else {  publish.value += ' 00:00:00' ; } }
  use_global = document.getElementById('detailspaginate2') ;
  no_paginate = document.getElementById('detailspaginate0') ;
  paginate = document.getElementById('detailspaginate1') ;
  question_page = document.getElementById('detailsquestionPage') ;
  slide = document.getElementById('detailsslide1') ;
  no_slide = document.getElementById('detailsslide0') ;
  publish_down = document.getElementById('detailspublish_down') ;
  publish_up = document.getElementById('detailspublish_up') ;

  function noPagination() { slide.checked = false ;
    no_slide.checked = false ; slide.disabled = true ;
    no_slide.disabled = true ; question_page.value = 0 ;
    question_page.disabled = true ;
    return true ;  }
  if (use_global.checked || no_paginate.checked ) { noPagination() ; }
    use_global.onclick = function () { noPagination() ; return true ;  }
    no_paginate.onclick = function () { noPagination() ; return true ;  }
    paginate.onclick = function () { question_page.disabled = false ;
      question_page.value = 5 ;  slide.disabled = false ;
      no_slide.disabled = false ; no_slide.checked = true ;
      return true;    }
    publish_up.onchange = function() {  checkDate(publish_up) ;}
    publish_down.onchange = function() {
  if (publish_down.value != "" && publish_down.value != "<?php echo JText::_('Never')
      ; ?>")   {  checkDate(publish_down) ; }
  if (publish_down.value == "") {
        publish_down.value = '<?php echo JText::_('Never') ; ?>' ; } }
document.getElementById('chain').value = document.getElementById( 'id_id' ).value;
document.getElementById('chain_title').value = document.getElementById( 'id_name' ).value;
document.getElementById('chain1').value = document.getElementById( 'id1_id' ).value;
document.getElementById('chain1_title').value = document.getElementById(
'id1_name').value; }
function buildDate(object) { dateStr = new String() ; dateStr = object.value ;
  year = dateStr.slice(0,4) ; month = dateStr.slice(5,7) ; month -= 1 ;
  day = dateStr.slice(8,10) ; hours = dateStr.slice(11,13) ;
  minutes = dateStr.slice(14,16) ;  seconds = dateStr.slice(17,19) ;
  date = new Date(year, month, day) ;  date.setHours(hours);
  date.setMinutes(minutes);   date.setSeconds(seconds);
  return date ;  }
```

```
// show or hide the session control according to the value ok access of the node
// (public or registred)
function sessionControl() {
  accessPublic     = document.getElementById('accessp') ;
  accessRegistred   = document.getElementById('accessr') ;
  session_control   =  document.getElementById('session_control') ;
  uniqueSession     =  document.getElementById('unique_session') ;
  if (accessPublic.checked) {  session_control.style.display = 'none' ;
    uniqueSession.checked = false ; }
  if (accessRegistred.checked) { session_control.style.display = '' ;  }  }
function clearMessages()  {  // remove joomla messages
  statusMessage = document.getElementsByTagName('dl') ;
  if (statusMessage.length == 2)  { JoomlaStatusMessage = statusMessage[0] ;
    JoomlaStatusMessage.parentNode.removeChild(JoomlaStatusMessage) ;   }
 // clear INconnect messages error
  var errorList = document.getElementById('errorList') ;
  errorList.innerHTML = '' ;  }
function addError(error)  { messageDiv = document.getElementById('message') ;
  messageDiv.style.display = "block" ;  var  errorList =
  document.getElementById('errorList') ;
  var errorUL = document.createElement("UL") ;
  var errorLI = document.createElement("LI") ;
  errorLI.innerHTML = error ;  errorUL.appendChild(errorLI) ;
  errorList.appendChild(errorUL) ; }
function submitbutton(pressbutton) { clearMessages();
  var form = document.adminForm;
//document.getElementById('chain1').value = document.getElementById( 'id_id' ).value;
//document.getElementById('chain1_title').value = document.getElementById( '[id]' ).value;
  if (pressbutton == 'cancel') { submitform( pressbutton ); return ;  }
  detailspaginate1 = document.getElementById('detailspaginate1') ;
    if ( (isNaN(question_page.value) || question_page.value < 1) &&
    detailspaginate1.checked )   {
      question_page = document.getElementById('detailsquestionPage') ;
    addError("\"" + question_page.value + "\" " + "<?php echo
    JText::_("INVALID_NUMBER_OF_QUESTION_PER_PAGE") ; ?>") ;
     question_page.focus() ;  question_page.select() ;
     return false ;  }
  if (form.title.value == "")  {
    addError("<?php echo JText::_( 'PLEASE_PROVIDE_A_TITLE_FOR_THE_QUIZ' ); ?>") ;
    return false ;  }   // getting the time limit
  timeLimit = document.getElementById('time_limit') ;
  if (timeLimit.value != "" && (isNaN(timeLimit.value) || timeLimit.value <= 0) ) {
```

```
      addError("<?php echo JText::_( 'TIME_LIMIT_NOT_VALID') ?>" ) ;
      timeLimit.focus() ;   timeLimit.select() ;
      return false ;    }  // getting the date of publishing
    publishUp = document.getElementById('detailspublish_up') ;
    publishUpDate = buildDate(publishUp) ;
    // getting the date of end of publishing
    publishDown = document.getElementById('detailspublish_down') ;
    publishDownDate = buildDate(publishDown) ;
    if (publishUpDate >= publishDownDate)   {
      addError("<?php echo JText::_( 'PUBLISH_DATE_INTERVAL_WRONG') ?>" ) ;
      publishDown.focus() ;
      return false ; }
    submitform( pressbutton ); } </script>
<script type="text/javascript" language="javascript">
  function proceed()  { check = document.getElementById('show_chain') ;
      proceedButton = document.getElementById('proceedButton') ;
      if (check.checked) { proceedButton.disabled = false ;
      } else { proceedButton.disabled = true ; } } </script>
<script type="text/javascript" language="javascript">
  /**
  *  Debug Function, that works like print_r for Objects in Javascript
  */
  function var_dump(obj) { var vartext = "";   //  alert('vartext: ' + vartext);
    for (var prop in obj) { if( isNaN( prop.toString() )) {
      if(eval( "obj."+prop.toString())) {
        vartext += "\t->"+prop+" = "+ eval( "obj."+prop.toString()) +"\n"; }}}
  //  alert('vartext1: ' + vartext);
      if(typeof obj == "object") {
        return "Type: "+typeof(obj)+((obj.constructor) ? "\nConstructor: "+obj.constructor :
        "") + "\n" + vartext;
       } else {  return "Type: "+typeof(obj)+"\n" + vartext; }
  }//end function var_dump
  function print_r(arr,level) { var dumped_text = ""; if(!level) level = 0;
  //The padding given at the beginning of the line.
  var level_padding = "";
  for(var j=0;j<level+1;j++) level_padding += "    ";
  if(typeof(arr) == 'object') { //Array/Hashes/Objects
  //alert('level: ' + level + ' arr: ' + arr);
      for(var item in arr) {  var value = arr[item];
          if(typeof(value) == 'object') { //If it is an array,
              dumped_text += level_padding + "'" + item + "' ...\n";
  //alert(' dumped_text: ' + dumped_text);
```

```
            dumped_text += print_r(value,level+1); }
        else { dumped_text += level_padding + "'" + item + "' => \"" + value + "\"\n";}}
  //  alert(' dumped_text: ' + dumped_text);
  } else { //Stings/Chars/Numbers etc. dumped_text = "===>"+arr+"<===("+typeof(arr)+")";}
  return dumped_text; } </script>
<?php //JLog::add(JText::_(INconnectHelper::indentjson(json_encode($this->nodes)).' =
//$this->nodes line 29 inconnect elements nodeelement '), JLog::WARNING, 'com_inconnect');?>
<form action="index.php" method="post" name="adminForm" id="adminForm">
  <div id="message" style="display:none;">
    <dl id="system-message">
    <dt class="error">Error</dt>
    <dd id="errorList" class="error message fade"></dd>
  </dl>
  </div>
  <div class="col100">
    <table>
    <tr valign="top">
      <td></td>
    <fieldset class="adminform">
      <legend><?php echo JText::_( 'Node' ); ?></legend>
    <table class="admintable">
      <tr>  <td width="100" align="right" class="key">
          <label for="title"><?php echo JText::_( 'TITLE' ) ; ?>:</label></td>
        <td>  <input class="text_area" name="title" id="title" size=40 value="<?php echo
        $this->node->title ; ?>" /></td></tr>
      <tr>  <td width="20" align="right" class="key"><label for="description"><?php echo
        JText::_( 'DESCRIPTION' ) ; ?>:</label></td>
        <td><textarea class="text_area" name="description" id="description" cols=40 ><?php
        echo $this->node->description ; ?></textarea></td></tr>
      <tr>  <td width="20" align="right" class="key"><label for="description">
      <?php echo JText::_( 'ACCESS' ) ; ?>:</label></td>
        <td>  <input class="radio" type="radio" name="access" id="accessp" value="0"
        onchange="sessionControl()" <?php if ($this->node->access_id == 0) echo 'checked'
        ;?> />
        <label for="accessp"><?php echo JText::_('PUBLIC') ?></label>
        <input class="radio" type="radio" name="access" id="accessr" value="1"
        onchange="sessionControl()" <?php if ($this->node->access_id == 1 ) echo 'checked'
        ;?> />
        <label for="accessr"><?php echo JText::_('REGISTRED') ?></label></td></tr>
      <tr id="session_control" style="<?php if ( 1 != $this->node->access_id ) : echo
      'display:none;' ; endif ; ?>">
        <td width="20" align="right" class="key">
```

```
              <span class="editlinktip hasTip" title="<?php echo JText::_( 'UNIQUE_SESSION'
              );?>::<?php echo JText::_("UNIQUE_SESSION_MAY_BE_PASSED_ONLY_ONCE"); ?>">
              <label for="unique_session"><?php echo JText::_('UNIQUE_SESSION') ; ?></label>
        </span></td><td>
        <input type="checkbox" name="unique_session" id="unique_session" <?php if
        ($this->node->unique_session == true) echo 'checked' ?>/></td></tr>
      <tr> <td width="20" align="right" class="key">
          <label for="description"><?php echo JText::_( 'TIME_LIMIT' ) ; ?>:</label></td>
        <td>  <input type="text" name="time_limit" id="time_limit" value="<?php echo
        ($this->node->time_limit) ; ?>" size="4" /><?php echo ' ' . JText::_('MINUTES') ; ?>
        </td></tr>
      <tr>  <td width="20" align="right" class="key">
          <label for="show_results"><?php echo JText::_('SHOW_RESULTS') ; ?></label></td>
        <td>  <input type="checkbox" name="show_results" id="show_results" <?php if
          ($this->node->show_results == true) echo 'checked' ?>/></td></tr>
      <tr>  <td width="20" align="right" class="key">
              <label for="show_agree"><?php echo "Show agreement filter" ; ?></label></td>
            <td>  <input type="checkbox" name="show_agree" id="show_agree" <?php if
              ($this->node->show_agree == true) echo 'checked' ?>/></td></tr>
      <tr>  <td width="20" align="right" class="key">
              <label for="show_chain"><?php echo "Show nodes" ; ?></label></td>
            <td>  <input type="checkbox" name="show_chain" id="show_chain" <?php if
              ($this->node->show_chain == true) echo 'checked' ?>/></td></tr>
<?php $_name = ''; $name = 'success';
    $nodesuccess = $this->nodeedges[0]->success_node_child_id;
    $_name1 = '';      $name1 = 'failure';
    $nodefailure = $this->nodeedges[0]->node_child_id;
    $node = $this->node->id; //dump($node, '$node: ');
    $doc =& JFactory::getDocument();
      if ($this->node->show_chain == true) {
      $html_stuff = JElementNode::fetchElementSuccess($name, $nodesuccess, $node,
      $nodesuccess, $control_name);
      $html_stuff .= "\n".'<tr>';
    $html_stuff .=  '<td>';
    $html_stuff .= JElementNode::fetchElement($name1, $nodefailure, $node, $nodefailure,
    $control_name);
    $html_stuff .= "\n".'<tr>';
    $html_stuff .= '<td>';
    echo $html_stuff;
/******************************/
//JLog::add(JText::_(INconnectHelper::indentjson(json_encode($this->edgeslists["edges"])).'
//= $this->edgeslists line 449 inconnect view node default'), Log::WARNING,'com_inconnect');
```

```
    if (count( $this->edgeslists )) :
//        echo $this->edgeslists["edges"].'</td>';
    endif ;     if (count( $this->nodeedges )) :
      $k = 0;$i=0;
      foreach ($this->nodeedges as $nodeedge) {
//JLog::add(JText::_(INconnectHelper::indentjson(json_encode($edgelist)).' = $edgelist line
//454 inconnect view node default '), JLog::WARNING, 'com_inconnect');
          $k = 1 - $k;      }
       else :
          echo '<tr><td colspan="10">' . JText::_('THERE_ARE_NO_NODES') . '<br /><br />' .
          JText::_('MIN_TUTO_NODES') . '</td></tr>' ;
        endif ; echo '</tr>'; }   ?> </table> </fieldset> </td>
    <td width="150"> </td>
    <td><?php   /****** other column with tabs *************/
    $options = array(
          'onActive' => 'function(title, description){
          description.setStyle("display", "block");
          title.addClass("open").removeClass("closed");}',
        'onBackground' => 'function(title, description){
          description.setStyle("display", "none");
          title.addClass("closed").removeClass("open");
        }',
        'startOffset' => 0,  // 0 starts on the first tab, 1 starts the second, etc...
        'useCookie' => true // this must not be a string. Don't use quotes.
        ); ///dump($options, 'options: ');
        $html = array();
        $html[] = JHtmlTabs::start('content-pane', $options);
        $html[] = JHtmlTabs::panel('Parameters', 'details');
        $html[] = '<div class="width-60 fltlft">'; ///dump($html);
  foreach ($this->form->getFieldsets('details') as $fieldsets => $fieldset):
    $html[] = '<fieldset class="adminform">';
    $html[] = '<dl>';
    foreach($this->form->getFieldset($fieldset->name) as $field):
      if ($field->hidden):
        $html[] = $field->input;
      else:
        $html[] = '<dt>';
        $html[] = $field->label;
        $html[] = '</dt>';
        $html[] = '<dd'.(($field->type == 'Editor' || $field->type == 'Textarea') ? '
        style="clear: both; margin: 0;"' : '').'>';
        $html[] = $field->input;
```

```
        $html[] = '</dd>';
      endif; ///dump($html);
    endforeach;
          $html[] = '</dl>';
          $html[] = '</fieldset>';
        endforeach;
        $html[] = '</div>';
        $html[] = JHtmlTabs::panel(JText::_('INconnectHelper'),'panel-id-2'); //You can use
        any custom text
        $html[] = '$this->params: '.INconnectHelper::dump($this->params).'<br >';
        $html[] = '$this->form: '.INconnectHelper::dump($this->form).'<br >';
        $html[] = JHtmlTabs::end();
        echo implode("\n", $html); ?>  </td></tr></table></div>
  <div class="clr"></div>
   <input type="hidden" name="notify_message" value="<?php echo
   htmlspecialchars($this->node->notify_message) ; ?>"/>
  <input type="hidden" name="option" value="com_inconnect"/>
  <input type="hidden" name="cid[]" value="<?php echo $this->node->id; ?>" />
  <input type="hidden" name="published" value="<?php echo $this->node->published ; ?>" />
  <input type="hidden" name="task" value="" />
  <input type="hidden" name="controller" value="node" />
  <input type="hidden" name="filter_order" value="<?php echo $this->lists['order']; ?>" />
  <input type="hidden" name="filter_order_Dir" value="<?php echo $this->lists['order_Dir'];
  ?>" />
  <input type="hidden" name="node_edge_id" value="<?php echo $this->nodeedges[0]->id ?>" />
  <?php echo JHTML::_( 'form.token' ); ?>
</form>
```

Appendix C

Apache

Configuration information as maintained in the configuration file for the Apache HTTP server, known commonly as the server deamon, HTTPD.

C.1 httpd_PhD/conf

```
#
# This is the main Apache HTTP server configuration file.
Define SRVROOT "/local/Apache24"

ServerRoot "${SRVROOT}"
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80

# Dynamic Shared Object (DSO) Support
#
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule allowmethods_module modules/mod_allowmethods.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule authn_core_module modules/mod_authn_core.so
LoadModule authn_file_module modules/mod_authn_file.so
LoadModule authz_core_module modules/mod_authz_core.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_host_module modules/mod_authz_host.so
```

```
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule cgi_module modules/mod_cgi.so
LoadModule dir_module modules/mod_dir.so
LoadModule env_module modules/mod_env.so
LoadModule include_module modules/mod_include.so
LoadModule isapi_module modules/mod_isapi.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
LoadModule negotiation_module modules/mod_negotiation.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
LoadModule ssl_module modules/mod_ssl.so
LoadModule jk_module modules/mod_jk.so
LoadModule php7_module "/local/php/php7apache2_4.so"
^^I PHPIniDir "/local/php"
       AddType text/html .php
       AddHandler application/x-httpd-php .php
User daemon
Group daemon

# 'Main' server configuration
#
ServerAdmin jason.stanley@uts.edu.au
#

ServerName localhost

<Directory />
    AllowOverride none
    Require all denied
</Directory>

DocumentRoot "${SRVROOT}/htdocs"
<Directory "${SRVROOT}/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
<IfModule dir_module>
    DirectoryIndex index.php index.html
</IfModule>
```

```
<Files ".ht*">
    Require all denied
</Files>
ErrorLog "logs/error.log"
LogLevel warn

<IfModule log_config_module>
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    CustomLog "logs/access.log" common
</IfModule>

<IfModule alias_module>
    #
    # Redirect: Allows you to tell clients about documents that used to
    ScriptAlias /cgi-bin/ "${SRVROOT}/cgi-bin/"
</IfModule>


#
# "${SRVROOT}/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "${SRVROOT}/cgi-bin">
    AllowOverride None
    Options None
    Require all granted
</Directory>

<IfModule headers_module>
    #
    # Avoid passing HTTP_PROXY environment to CGI's on this or any proxied
    RequestHeader unset Proxy early
</IfModule>

<IfModule mime_module>
    #
    # TypesConfig points to the file containing the list of mappings from
    # filename extension to MIME-type.
    #
    TypesConfig conf/mime.types
```

```
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz
</IfModule>

# Configure mod_proxy_html to understand HTML4/XHTML1
<IfModule proxy_html_module>
Include conf/extra/proxy-html.conf
</IfModule>

# Secure (SSL/TLS) connections
Include conf/extra/httpd-ssl.conf
#
# Note: The following must must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>
Include conf/other/*.conf
AddDefaultCharset utf-8
```

Appendix D

Apache-Tomcat

Configuration information as maintained in the configuration file for the Apache-Tomcat Java/servlets server.

D.1 mod_jk_PhD/conf

```
# Replace jsp-hostname with the properties of your JSP server, as
# specified in workers.properties
#

<IfModule mod_jk_module>
    JKWorkersFile conf/workers.properties
^^IJKLogFile /local/Apache24/logs/mod_jk.log
^^IJKShmFile /local/Apache24/logs/mod_jk.shm
^^IJKLogLevel error
^^I
^^IJKMount /*.jsp tutoria-jsp
^^IJKMount /examples/* tutoria-jsp
^^IJKMount /tmintra/* tutoria-jsp
^^IJKMount /webMathematica/* tutoria-jsp
^^I
</IfModule>
^^I
```

D.2 context_PhD/xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
```

```
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<!-- The contents of this file will be loaded for each web application -->
<Context>

    <!-- Default set of monitored resources. If one of these changes, the    -->
    <!-- web application will be reloaded.                                   -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat restarts -->
    <!--
    <Manager pathname="" />
    -->
</Context>

```

D.3 tomcat-users_PhD/xml

```
<?xml version="1.0" encoding="UTF-8"?>
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">

<role rolename="manager-gui"/>
<role rolename="admin-gui"/>
<user
    username="admin"
    password="admin"
```

```
    roles="manager-gui,admin-gui,tmadmin"/>
</tomcat-users>

```

D.4 server_PhD/xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Note:  A "Server" is not itself a "Container", so you may not
     define subcomponents such as "Valves" at this level.
     Documentation at /docs/config/server.html
 -->
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
  <GlobalNamingResources>
    <!-- Editable user database that can also be used by
         UserDatabaseRealm to authenticate users
    -->
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
              description="User database that can be updated and saved"
              factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
              pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>
  <!-- A "Service" is a collection of one or more "Connectors" that share
       a single "Container" Note:  A "Service" is not itself a "Container",
       so you may not define subcomponents such as "Valves" at this level.
       Documentation at /docs/config/service.html
   -->
  <Service name="Catalina">

    <!--The connectors can use a shared executor, you can define one or more named thread pools--
    <!--<Executor name="tomcatThreadPool" namePrefix="catalina-exec-"maxThreads="150" minSpareThr

    <!-- A "Connector" represents an endpoint by which requests are received
         and responses are returned. Documentation at :
         Java HTTP Connector: /docs/config/http.html
         Java AJP  Connector: /docs/config/ajp.html
```

```
         APR (HTTP/AJP) Connector: /docs/apr.html
         Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
    -->
    <Connector port="8080" protocol="HTTP/1.1"
               URIEncoding="UTF-8"
               connectionTimeout="20000"
               redirectPort="8443" />

    <!-- Define an AJP 1.3 Connector on port 8009
^^I     This connector joins Apache and Tomcat using workers -->
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" URIEncoding="UTF-8"/>


    <!-- An Engine represents the entry point (within Catalina) that processes
         every request.   -->
    <Engine name="Catalina" defaultHost="localhost">
      <!-- Use the LockOutRealm to prevent attempts to guess user passwords
           via a brute-force attack -->
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <!-- This Realm uses the UserDatabase configured in the global JNDI
             resources under the key "UserDatabase".  Any edits
             that are performed against this UserDatabase are immediately
             available for use by the Realm.  -->
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase"/>
      </Realm>

      <Host name="localhost"  appBase="webapps"
            unpackWARs="true" autoDeploy="true"
            xmlValidation="false" xmlNamespaceAware="false">

        <!-- Access log processes all example.
             Documentation at: /docs/config/valve.html
             Note: The pattern used is equivalent to using pattern="common" -->
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
               prefix="localhost_access_log" suffix=".txt"
               pattern="%h %l %u %t &quot;%r&quot; %s %b" />

      </Host>
    </Engine>
  </Service>
```

```
</Server>

```

D.5 web_PhD/xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
                      http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0">
    <servlet>
        <servlet-name>default</servlet-name>
        <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
        <init-param>
            <param-name>debug</param-name>
            <param-value>11</param-value>
        </init-param>
        <init-param>
            <param-name>listings</param-name>
            <param-value>true</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet>
        <servlet-name>jsp</servlet-name>
        <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
        <init-param>
            <param-name>fork</param-name>
            <param-value>false</param-value>
        </init-param>
        <init-param>
            <param-name>xpoweredBy</param-name>
            <param-value>false</param-value>
        </init-param>
        <load-on-startup>3</load-on-startup>
    </servlet>

    <!-- The mapping for the default servlet -->
    <servlet-mapping>
        <servlet-name>default</servlet-name>
```

```
        <url-pattern>/</url-pattern>
    </servlet-mapping>
    <!-- The mappings for the JSP servlet -->
    <servlet-mapping>
        <servlet-name>jsp</servlet-name>
        <url-pattern>*.jsp</url-pattern>
        <url-pattern>*.jspx</url-pattern>
    </servlet-mapping>
  <!-- ==================== Default Session Configuration ================= -->
  <!-- You can set the default session timeout (in minutes) for all newly   -->
  <!-- created sessions by modifying the value below.                       -->

    <session-config>
        <session-timeout>30</session-timeout>
    </session-config>

  <!-- ===================== Default MIME Type Mappings =================== -->
  <!-- When serving static resources, Tomcat will automatically generate    -->
  <!-- a "Content-Type" header based on the resource's filename extension,  -->
  <!-- based on these mappings.  Additional mappings can be added here (to  -->
  <!-- apply to all web applications), or in your own application's web.xml -->
  <!-- deployment descriptor.                                               -->
  <!-- Note: Extensions are always matched in a case-insensitive manner.    -->

    <mime-mapping>
        <extension>123</extension>
        <mime-type>application/vnd.lotus-1-2-3</mime-type>
    </mime-mapping>
      .
      .
      .
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

```
end of configuration files
```

Appendix E

Site Scope, Directories and Files

```
/c/users/115396/desktop/tutoria_holding/joomla_3/components/com_inconnect
├── controller.php
├── controllers
│   ├── index.html
│   ├── nextstepnode.php
│   ├── node.php
│   ├── nodes.php
│   ├── quiz.php
│   ├── quiz.php.old
│   ├── quizzes.php
│   ├── session.php
│   ├── sessionnode.php
│   ├── tutoria.php
│   ├── tutoriasessionwho.php
│   └── who.php
├── evaluate_ajax
│   └── answer
│       └── index.php
├── helpers
│   ├── helper.php
│   ├── html.helper.php
│   └── query.php
├── inconnect.php
├── index.html
├── models
│   ├── categories.php
│   ├── index.html
│   ├── nextstepnode.php
│   ├── node.php
│   ├── nodes.php
│   ├── organizations.php
│   ├── quiz.php
│   ├── quizzes.php
│   ├── session.php
│   ├── sessionnode.php
│   ├── tutoria.php
│   └── who.php
├── router.php
├── tree_admin_latex.txt
└── views
    ├── _mathsinsidesessionwho
    ├── _nextstep
    │   └── tmpl
    ├── _nextstepnode
    │   └── tmpl
    ├── _session
    │   ├── index.html
    │   ├── tmpl
    │   │   ├── default.php
    │   │   ├── dynamic.php
    │   │   └── index.html
    │   └── view.html.php
    ├── _sessionnode
    │   ├── index.html
    │   ├── tmpl
    │   │   ├── default.php
```

```
│   │   ├── dynamic.php
│   │   └── index.html
│   └── view.html.php
├── _tutoriasessionwho
│   ├── index.html
│   ├── tmpl
│   │   ├── default.php
│   │   └── index.html
│   └── view.html.php
├── _who
│   ├── index.html
│   ├── tmpl
│   │   ├── _form.php
│   │   └── index.html
│   └── view.html.php
├── helloworld
│   ├── index.html
│   ├── tmpl
│   │   ├── default.php
│   │   ├── default.xml
│   │   └── index.html
│   └── view.html.php
├── index.html
├── mathsinside
│   └── tmpl
│       ├── default.php
│       ├── dynamic.php
│       └── index.html
├── nextstepnode
│   ├── index.html
│   ├── tmpl
│   │   ├── default.php
│   │   ├── dynamic.php
│   │   └── index.html
│   └── view.html.php
├── node
│   ├── tmpl
│   │   ├── _nodeinfo.php
│   │   ├── calcmath_2.php
│   │   ├── calcmath_menu_palettes.php
│   │   ├── default.php
│   │   ├── default.xml
│   │   ├── index.html
│   │   └── store
│   │       ├── calcmath.php
│   │       ├── calcmath_menu_palettes_.php
│   │       └── calcmath_original_works.php
│   └── view.html.php
├── nodes
│   ├── index.html
│   ├── tmpl
│   │   ├── default.php
│   │   ├── default.xml
│   │   └── index.html
│   └── view.html.php
├── quiz
│   ├── index.html
```

```
│   ├── tmpl
│   │   ├── _quizinfo.php
│   │   ├── default.php
│   │   ├── default.xml
│   │   └── index.html
│   └── view.html.php
├── quizzes
│   ├── index.html
│   ├── tmpl
│   │   ├── default.php
│   │   ├── default.xml
│   │   └── index.html
│   └── view.html.php
└── tutoria
    ├── index.html
    ├── tmpl
    │   ├── _quizinfo.php
    │   ├── default.php
    │   ├── default.php.bak
    │   ├── default.xml
    │   ├── default_May_Save.php
    │   ├── default_works_so_use.php
    │   ├── dyn_catts.php
    │   ├── dyn_save.php
    │   ├── dyn_select.php
    │   ├── dyn_select_1.php
    │   ├── dyn_stgyrs.php
    │   ├── dyn_strand.php
    │   ├── dyn_update.php
    │   └── index.html
    └── view.html.php

37 directories, 107 files
```

Appendix F

Data Analysis

Cautionary Note

This analysis is based on a small, illustrative dataset (`n = 17` per group) designed primarily to demonstrate satisfaction of the aims of the project. The statistical inferences drawn should therefore be interpreted with caution.

ANOVA

ANOVA was applied to the bootstrapped data. One-way ANOVA facilitated the examination of between-group effects, while repeated measures ANOVA enabled the detection of within-subject changes and interaction effects between time and condition [102].

Effect Size (raw scores). In addition to statistical significance testing, we calculated the effect size to evaluate the practical impact of the observed difference. Using Cohen's d for paired samples, the mean difference between Tutoria and MRS scores was estimated as $\bar{X}_D = 0.471$, with a standard deviation of the paired differences approximately $s_D = 0.635$, yielding an effect size of:

$$d = \frac{\bar{X}_D}{s_D} = \frac{0.471}{0.635} \approx 0.74$$

According to Cohen's widely used benchmarks [51], this corresponds to a *medium to large* effect size. This suggests that the observed improvement in student performance was not only statistically significant, but also meaningful in an educational context.

Effect Size (Bootstrapped). To evaluate the practical impact of the observed improvement, Cohen's d was calculated as a descriptive effect size using the paired differences between MRS and Tutoria scores. The raw value was $d = -0.47$, reflecting an increase in scores under the Tutoria condition.

To account for non-normality and small sample size, a bootstrap resampling procedure (10,000 replicates) was applied to the paired differences. This yielded a 95% confidence interval of approximately $[-0.81, -0.22]$.

This interval indicates a medium effect size [51], and supports the interpretation that the improvement was not only statistically significant, but also meaningful in practical terms.

F.1 R_Comprehensive

The following activity is included to demonstrate the use of R as a stand-alone statistical analysis tool.
R code can run on a command-line or it can be used with notebook style files, in an interactive environment. The notebook and its outputs appear below.

Statistical Output and Analysis Script

Summary of Statistical Analysis

The R code provided below performs a comparative statistical analysis between two groups, `MRS` and `Tutoria`, consisting of 17 samples each. The steps executed include:

- Descriptive statistics: mean, standard deviation, and boxplots for both groups.
- Bootstrap resampling (5000 iterations) to estimate the sampling distribution of the mean and compute confidence intervals.
- Levene's test for homogeneity of variance (both raw and transformed data).
- ANOVA on both square-root transformed and Box-Cox transformed data to test for significant group differences.
- Residual diagnostic plots and multiple normality tests (Shapiro-Wilk, Kolmogorov-Smirnov, Cramér–von Mises, Anderson–Darling).
- Non-parametric Wilcoxon rank-sum test and paired t-tests.
- ggplot visualisations: group boxplots and paired profile plots.

This thorough sequence ensures robustness in testing assumptions and in evaluating group differences from both parametric and non-parametric perspectives.

R Code Used for Statistical Analysis - Initial

```
# Load required libraries
library(ggplot2)
library(ggfortify)
library("dplyr")
library(ggpubr)
library(PairedData)
library(nortest)
library(MASS)
library(DescTools)
```

```
library(lawstat)

rm(list=ls())
sink("outputfile.txt")
# Define score vectors
mrs <- c(4,3,3,0,4,2,1,1,3,2,4,5,3,2,0,3,4)
tutoria <- c(5,4,3,1,4,2,2,1,3,3,4,5,4,2,1,3,5)
# Summary statistics
summary(mrs)
summary(tutoria)
mean(mrs)
mean(tutoria)
sd(mrs)
sd(tutoria)
boxplot(mrs,tutoria)
# Wilcoxon and t-tests
wilcox.test(mrs, tutoria, alternative="two.sided")
############################################################
# Bootstrap sampling
nboot <- 5000
data_mrs <- matrix(0, ncol=17, nrow=nboot)
for(i in 1:nboot) data_mrs[i,] <- sample(mrs, replace=TRUE)
data_row_mean_mrs <- rowMeans(data_mrs)
quantile(data_row_mean_mrs, prob = c(0.025, 0.975))
hist(data_row_mean_mrs, freq = FALSE)
curve(dnorm(x, mean=mean(data_row_mean_mrs), sd=sd(data_row_mean_mrs)),
col="grey", lwd=2, add=TRUE)
# Repeat for Tutoria
data_tutoria <- matrix(0, ncol=17, nrow=nboot)
for(i in 1:nboot) data_tutoria[i,] <- sample(tutoria, replace=TRUE)
data_row_mean_tutoria <- rowMeans(data_tutoria)
quantile(data_row_mean_tutoria, prob = c(0.025, 0.975))
hist(data_row_mean_tutoria, freq = FALSE)
curve(dnorm(x, mean=mean(data_row_mean_tutoria),
sd=sd(data_row_mean_tutoria)), col="grey", lwd=2, add=TRUE)
# Difference analysis
diff <- mrs - tutoria
data_diff <- matrix(0, ncol=17, nrow=nboot)
for(i in 1:nboot) data_diff[i,] <- sample(diff, replace=TRUE)
data_row_mean_diff <- rowMeans(data_diff)
quantile(data_row_mean_diff, prob = c(0.025, 0.975))
hist(data_row_mean_diff, freq = FALSE)
```

```
curve(dnorm(x, mean=mean(data_row_mean_diff),
sd=sd(data_row_mean_diff)), col="grey", lwd=2, add=TRUE)
# Build data frame
my_data <- data.frame(data_row_mean_mrs, data_row_mean_tutoria)
# Wilcoxon and t-tests
wilcox.test(data_row_mean_mrs, data_row_mean_tutoria, alternative="two.sided")
# Grouped data and summary
my_data_ <- data.frame(group = rep(c("mrs", "tutoria"),
each=length(data_row_mean_mrs)), weight = c(data_row_mean_mrs,
data_row_mean_tutoria))
group_by(my_data_, group) %>% summarise(count = n(), mean = mean(weight), sd = sd(weight))
ggboxplot(my_data_, x = "group", y = "weight", color = "group",
palette = c("#00AFBB", "#E7B800"), order = c("mrs", "tutoria"),
ylab = "Weight", xlab = "Groups")
mrs <- subset(my_data_, group == "mrs", weight, drop = TRUE)
tutoria <- subset(my_data_, group == "tutoria", weight, drop = TRUE)
pd <- paired(mrs, tutoria)
plot(pd, type = "profile") + theme_bw()
d <- with(my_data_, weight[group == "mrs"] - weight[group == "tutoria"])
shapiro.test(d)
t.test(mrs, tutoria, paired = TRUE)
t.test(weight ~ group, data = my_data_)
t.test(weight ~ group, data = my_data_, alternative = "less")
t.test(weight ~ group, data = my_data_, alternative = "greater")
```

R Code Used for Statistical Analysis - 2nd Iteration

```
rm(list=ls()) # remove all objects
mrs     <- c(4,3,3,0,4,2,1,1,3,2,4,5,3,2,0,3,4)
tutoria <- c(5,4,3,1,4,2,2,1,3,3,4,5,4,2,1,3,5)
diff <- mrs - tutoria
nboot <- 5000
data_diff <- matrix(0,ncol =17,nrow = nboot)
for(i in 1:nboot) {
  data_diff[i,] <- sample(x= diff, size = length(diff), replace = T)
}
diff_means <- rowMeans(data_diff)
hist(diff_means)
diff_mean <- mean(diff_means)
diff_mean
diff_sd <- sd(diff_means)
diff_sd
t975 <- qt(0.975, df=4999)
t975
lb <- diff_mean - t975*diff_sd / sqrt(nboot)
ub <- diff_mean + t975*diff_sd / sqrt(nboot)
c(lb,ub)
t.test(diff_means, mu=0, alternative="two.sided")
######################################
Diff Mean     -0.4713882
Diff Std Dev   0.1214028
t975           1.960439
CI            -0.4747541 -0.4680224
One Sample t-test
data:  diff_means
t = -274.56, df = 4999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.4747541 -0.4680224
sample estimates:
 mean of x
-0.4713882
```

Statistical Output Summary - 2nd Iteration

Descriptive Statistics

Table F.1 Descriptive statistics

Group	Min	1st Qu.	Median	Mean	3rd Qu.	Max
MRS	0.000	2.000	3.000	2.588	4.000	5.000
Tutoria	1.000	2.000	3.000	3.059	4.000	5.000

Group Summary Statistics

Table F.2 Group means and standard deviations

Group	Count	Mean	SD
MRS	5000	2.59	0.346
Tutoria	5000	3.06	0.330

Inferential Statistics

Table F.3 ANOVA results

Source	Df	Sum Sq	Mean Sq	F value	Pr(>F)
data_row_mean_tutoria	1	0.00	0.000013	0.001	0.973
Residuals	4998	58.95	0.011795		

Normality Tests

- **Shapiro-Wilk:** W = 0.99304, p-value = 6.502×10^{-15}
- **Kolmogorov-Smirnov:** D = 0.048956, p-value = 7.803×10^{-11}
- **Cramér–von Mises:** W = 1.6047, p-value = 7.37×10^{-10}
- **Anderson–Darling:** A = 9.2863, p-value $< 2.2 \times 10^{-16}$

Wilcoxon Rank-Sum Test

- W = 4155086, p-value $< 2.2 \times 10^{-16}$
- Alternative hypothesis: true location shift is not equal to 0

Welch Two-Sample t-tests

- **t** = -68.926, df = 9975.6, p-value $< 2.2 \times 10^{-16}$
- 95% CI: $(-\infty, -0.4552)$
- **Means:**
  - MRS = 2.5942
  - Tutoria = 3.0606

F.2 Results Used (Final Analysis)

CI used for final conclusions The standard calculation for Confidence Interval (CI) with paired data is:

$$(1-\alpha)\,\%CI = \bar{x} \pm t_{n-1,\frac{\alpha}{2}} \times \frac{s_d}{\sqrt{n}}$$

- **Raw Means:**
  - MRS = 2.5942
  - Tutoria = 3.0606
- **Bootstrapped Difference in Group Means:**
  - Group Difference Mean = -0.4692824
  - Group Difference Std Dev = 0.1222535
  - Group t975 = 1.960439
  - 95% CI: = (-0.4726718, -0.4658929)
-
```
t.test(diff_means, mu=0, alternative="two.sided")
  One Sample t-test
  data:  diff_means
  t = -271.43, df = 4999, p-value < 2.2e-16
  alternative hypothesis: true mean is not equal to 0
  95 percent confidence interval:
     -0.4726718 -0.4658929
  sample estimates:
  mean of x
   -0.4692824
```

Summary The Confidence Interval (CI) may be used instead of a ‘t’ test. The CI on the ‘t’ test is the same as the manually calculated CI, in the item above (“**Bootstrapped Difference in Group Means:**”). As 0 is not in the CI then it can be concluded there is a difference in the group means. The use of the t975 at 1.96 can be changed to two (2) Standard Deviations. Two (2) Standard Deviations is commonly used with bootstrapping see Appendix H and the exercise was repeated, the results follow:

- Group Difference Mean = -0.4692824
- Group Difference Std Dev = 0.1222535
- Group t = 2
- 95% CI: = (-0.4769451 -0.4701372)

The CI is similar to the ’t’ test and the previous manual calculation. At two standard deviations the salient point is that zero is *not* in the confidence interval, suggesting statistical significance about the difference of the group means and as a result the null hypothesis can be rejected.

```
rm(list=ls()) # remove all objects
#sink("Tutoria_final.txt")
mrs     <- c(4,3,3,0,4,2,1,1,3,2,4,5,3,2,0,3,4)
tutoria <- c(5,4,3,1,4,2,2,1,3,3,4,5,4,2,1,3,5)
diff <- mrs - tutoria
nboot <- 5000
data_diff <- matrix(0,ncol =17,nrow = nboot)
for(i in 1:nboot) {
   data_diff[i,] <- sample(x= diff, size = length(diff), replace = T)
 }
diff_means <- rowMeans(data_diff)
hist(diff_means)
diff_mean <- mean(diff_means)
diff_mean
Mean -0.4692824
diff_sd <- sd(diff_means)
diff_sd
Std Dev 0.1222535
t975 <- qt(0.975, df=4999)
t975
[1] 1.960439
lb <- diff_mean - t975*diff_sd / sqrt(nboot)
ub <- diff_mean + t975*diff_sd / sqrt(nboot)
c(lb,ub)
CI -0.4726718 -0.4658929
lb_ <- diff_mean - 2*diff_sd / sqrt(nboot)
ub_ <- diff_mean + 2*diff_sd / sqrt(nboot)
c(lb_,ub_)
```

```
CI -0.4769451 -0.4701372
t.test(diff_means, mu=0, alternative="two.sided")
One Sample t-test
data:  diff_means
t = -271.43, df = 4999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.4726718 -0.4658929
sample estimates:
 mean of x
-0.4692824
```

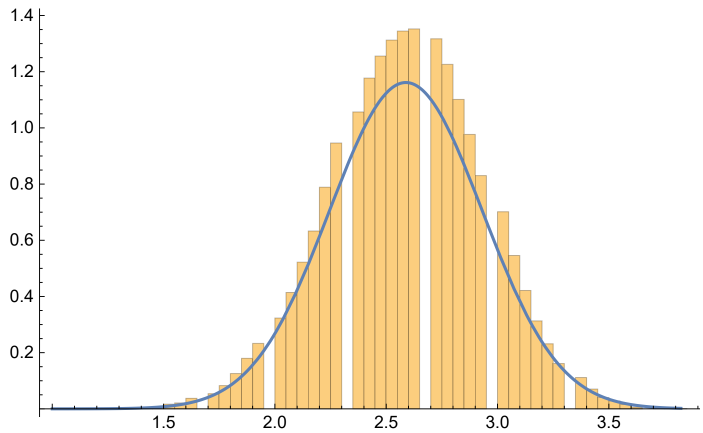
Appendix G

Mathematica Notebook Code

Only used in preparation and testing of initial bootstrapping paradigm The following activity is included to demonstrate the use of Mathematica as a stand-alone tool. The notebook and its outputs appear below. Many thanks to Scott Alexander for his patience and meaningful discussion during this part of the process.

```
In[24]:= mrs = {4, 3, 3, 0, 4, 2, 1, 1, 3, 2, 4, 5, 3, 2, 0, 3, 4} // N;
         tutoria = {5, 4, 3, 1, 4, 2, 2, 1, 3, 3, 4, 5, 4, 2, 1, 3, 5} // N;

In[65]:= sample = mrs;
         nSample = Length[sample];

         nBoot = 100000;
         boot = Table[RandomChoice[sample, nSample], nBoot];

         bootParams = Mean[Transpose[boot]];
         bootParamMin = Min[bootParams];
         bootParamMax = Max[bootParams];
         bootParamMean = Mean[bootParams];
         bootParamStdDev = StandardDeviation[bootParams];
         bootParamCI = Quantile[bootParams, {0.025, 0.975}];
         Print["Bootstrap X̄ = " <> ToString[bootParamMean] <>
            " with 95% CI " <> ToString[{bootParamCI[[1]], bootParamCI[[2]]}]];
         Show[Histogram[bootParams, 50, "PDF"],
          Plot[PDF[NormalDistribution[bootParamMean, bootParamStdDev], x],
           {x, bootParamMin, bootParamMax}]]
         Clear[nSample, α, β, sample, nBoot, bootParams, bootParamMin,
          bootParamMax, bootParamMean, bootParamStdDev, bootParamCI]

Bootstrap X̄ = 2.58833 with 95% CI {1.88235, 3.23529}
```

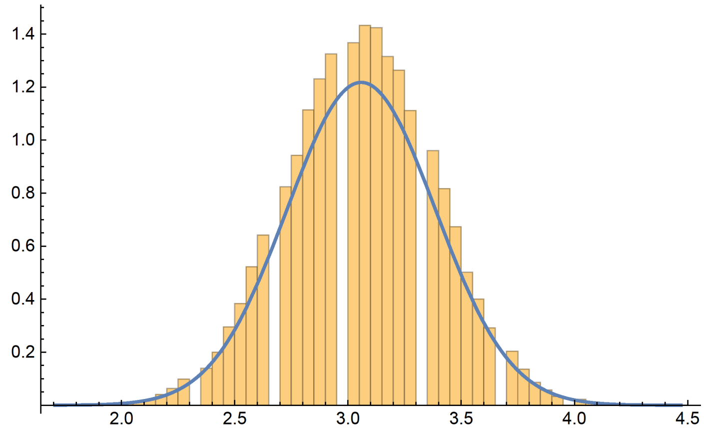
Out[76]=



In[78]:=
```
sample = tutoria;
nSample = Length[sample];

nBoot = 100 000;
boot = Table[RandomChoice[sample, nSample], nBoot];

bootParams = Mean[Transpose[boot]];
bootParamMin = Min[bootParams];
bootParamMax = Max[bootParams];
bootParamMean = Mean[bootParams];
bootParamStdDev = StandardDeviation[bootParams];
bootParamCI = Quantile[bootParams, {0.025, 0.975}];
Print["Bootstrap X̄ = " <> ToString[bootParamMean] <>
   " with 95% CI " <> ToString[{bootParamCI[[1]], bootParamCI[[2]]}]];
Show[Histogram[bootParams, 50, "PDF"],
 Plot[PDF[NormalDistribution[bootParamMean, bootParamStdDev], x],
  {x, bootParamMin, bootParamMax}]]
Clear[nSample, α, β, sample, nBoot, bootParams, bootParamMin,
 bootParamMax, bootParamMean, bootParamStdDev, bootParamCI]
```

```
Bootstrap X̄ = 3.05847 with 95% CI {2.41176, 3.70588}
```

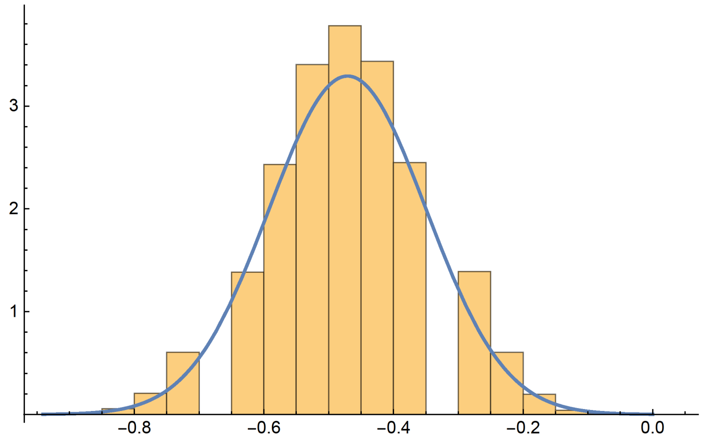
Out[89]=



In[91]:=
```
sample = mrs - tutoria;
nSample = Length[sample];

nBoot = 100 000;
boot = Table[RandomChoice[sample, nSample], nBoot];

bootParams = Mean[Transpose[boot]];
bootParamMin = Min[bootParams];
bootParamMax = Max[bootParams];
bootParamMean = Mean[bootParams];
bootParamStdDev = StandardDeviation[bootParams];
bootParamCI = Quantile[bootParams, {0.025, 0.975}];
Print["Bootstrap X̄ = " <> ToString[bootParamMean] <>
   " with 95% CI " <> ToString[{bootParamCI[[1]], bootParamCI[[2]]}]];
Show[Histogram[bootParams, 50, "PDF"],
 Plot[PDF[NormalDistribution[bootParamMean, bootParamStdDev], x],
  {x, bootParamMin, bootParamMax}]]
Clear[nSample, α, β, sample, nBoot, bootParams, bootParamMin,
 bootParamMax, bootParamMean, bootParamStdDev, bootParamCI]
```

```
Bootstrap X̄ = -0.470841 with 95% CI {-0.705882, -0.235294}
```

Out[102]=



Appendix H

References informally recorded

Personal Communication

Jason Stanley
University of Technology Sydney

On the advice of Examiner 3 references recorded as Personal Communication have been removed from the Thesis References and placed in this appendix as a record of their informal contribution to my thesis.

For their insightful but sometimes unexpected conversations, I would like to acknowledge...

Tim Langtry	A master of scientific narrative.
Mary Coupland	Mary's insight into UTS history and operations are impressive.
Danitza Solina	An academic with passion when it comes to educational testing and data design used in studies.
Scott Alexander	Scott is a very powerful computer scientist although this is not his professional specialty, truly impressive.
James Brown	Professor of Official Statistics is a very powerful statistician with deep knowledge of his field.

Informal Personal Communication — BibTex entries used but relocated after examination:


```
@misc{Alex2024,
     author      = {Scott Alexander},
     title       = {{Data in \emph{Mathematica}}},
     year        = 2024,
     note        = {Conversation and demonstration on the use of \emph{Mathematica}
                         to employ bootstrapping, January 2024},
     howpublished = {Personal communication}
}
@misc{Brown2024,
        title       = {{Statistics in bootstrapping}},
        author      = {{Brown, J.}},
        year        = {2024},
        note        = {Conversation on the use of 2 standard deviations when using bootstrapping, January 2024},
        howpublished = {Personal communication with Prof. Official Statistics}
}
@misc{Coup2014,
     author      = {Mary Coupland},
     title       = {{Pathways to UTS mathematics}},
     year        = 2014,
     note        = {Conversation on Pathways to UTS mathematics, January 2014},
     howpublished = {Personal communication}
}
@misc{Coup2015,
     author      = {Mary Coupland},
     title       = {{Pathways to University mathematics}},
     year        = 2015,
     note        = {Conversation on Pathways to University mathematics, October 2015},
     howpublished = {Personal communication}
}
@misc{Lang2018,
        title       = {{Education research discussion through a mathematicians' lens}},
        author      = {Langtry, T.},
        year        = {2018},
        note        = {Conversation on Pathways to University mathematics, September 11},
        howpublished = {Personal communication}
}
@misc{Solina2019,
     author      = {Danitza Solina},
     title       = {{Education research discussion through a physists' lens}},
     year        = {2019},
     note        = {Conversation on blind studies in education, May 8},
     howpublished = {Personal communication}
}
```

References

[1] Ahuja, N. J. and Sille, R. (2013). A Critical Review of Development of Intelligent Tutoring Systems: Retrospect, Present and Prospect. *International Journal of Computer Science Issues (IJCSI)*, 10(4):39–48. General online https://api.semanticscholar.org/CorpusID:18120762, however, Downloaded via UTS Library http://ezproxy.lib.uts.edu.au/login?url=https://www.proquest.com/scholarly-journals/critical-review-development-intelligent-tutoring/docview/1470806230/se-2, must have UTS staff access.

[2] Ainley, J., Kos, J., and Nicholas, M. (2008). Participation in science, mathematics and technology in australian education. https://research.acer.edu.au/acer_monographs/4. https://research.acer.edu.au/acer_monographs/4/.

[3] Ainsworth, S. and Grimshaw, S. (2004). Evaluating the REDEEM authoring tool: can teachers create effective learning environments? *International Journal of Artificial Intelligence in Education*, 14:279–312. https://dl.acm.org/doi/abs/10.5555/1434913.1434916.

[4] Aleven, V., McLaren, B. M., Sewall, J., and Koedinger, K. R. (2006). The Cognitive Tutor Authoring Tools (CTAT): Preliminary evaluation of efficiency gains. In Ikeda, M., Ashley, K. D., and Chan, T.-W., editors, *Intelligent Tutoring Systems: 8th International Conference, ITS 2006, Jhongli, Taiwan, June 26–30, 2006, Proceedings*, pages 61–70, Berlin, Heidelberg. Springer. Downloaded via UTS Library https://link.springer.com/chapter/10.1007/11774303_7, must have UTS staff access.

[5] Aleven, V., Ogan, A., Popescu, O., Matthew, R., Rosé, C., and Koedinger, K. R. (2016). Integrating authoring tools and tutors in CTAT: Towards an open learning platform. In Conati, C. *et al.*, editor, *Intelligent Tutoring Systems (ITS 2016)*, volume 9684 of *Lecture Notes in Computer Science*, pages 292–298. Springer. Downloaded via UTS Library https://link.springer.com/book/10.1007/978-3-319-39583-8, must have UTS staff access.

[6] Alexander, P. A., Schallert, D. L., and Reynolds, R. E. (2009). What is learning anyway? a topographical perspective considered. *Educational Psychologist*, 44(3):176–192. https://www.tandfonline.com/doi/abs/10.1080/00461520903029006.

[7] Alexander, R. J. (2008). *Towards Dialogic Teaching: Rethinking Classroom Talk*. Dialogos, 4th edition. https://pure.york.ac.uk/portal/en/publications/towards-dialogic-teaching-rethinking-classroom-talk-4th-edition/.

[8] Anderman, E. M., Anderman, L. H., and Ormrod, J. E. (2024). *Educational Psychology: Developing Learners*. Pearson, Boston, 11th edition. https://www.pearson.com/en-us/subject-catalog/p/educational-psychology-developing-learners/

P200000009802/9780137871384?srsltid=AfmBOoobGHsPWM2Y_s4YACng8LaD1VQSJm3c36dVR0rCUJyAzcDfhXnP.

[9] Anderson, J. R., Boyle, C. F., and Reiser, B. J. (1985). Intelligent tutoring systems. *Science*, 228(4698):456–462. https://psycnet.apa.org/record/1986-07584-001.

[10] Anderson, J. R. and Corbett, A. T. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278. https://link.springer.com/article/10.1007/BF01099821.

[11] Anderson, J. R., Corbett, A. T., Koedinger, K. R., and Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *Journal of the Learning Sciences*, 4(2):167–207. https://www.tandfonline.com/doi/abs/10.1207/s15327809jls0402_2.

[12] Anderson, T. and Shattuck, J. (2012). Design-Based Research: A Decade of Progress in Education Research? *Educational Researcher*, 41(1):16–25.

[13] Anderson, J. R. (1983). *The Architecture of Cognition*. Harvard University Press, Cambridge, Massachusetts. https://www.sciencedirect.com/science/article/pii/0004370286900846.

[14] Apache Software Foundation (2024). Apache tomcat. https://tomcat.apache.org/. Accessed: 2025-06-12.

[15] Artigue, M. (2009a). Didactical design in mathematics education. *Theories of mathematics education*, pages 467–489. https://doi.org/10.1163/9789087907839_003.

[16] Artigue, M. (2009b). Didactical design in mathematics education. In Winslow, C., editor, *Nordic Research in Mathematics Education: Proceedings from NORMA08 in Copenhaguen, April 21-April 25, 2008*, pages 7–16, Copenhaguen, Denmark. Sense Publishers. Paper available on Sense Publishers' website. https://hal.science/hal-00699775.

[17] Artigue, M (1994). Didactic engineering as a framework for the conception of teaching products. In Biehler, R., Scholz, R. W., Strässer, R., and Winkelmann, B., editors, *Didactics of Mathematics as a Scientific Discipline*, pages 27–39. Kluwer Academic Publishers, Dordrecht. https://link.springer.com/book/10.1007/0-306-47204-X and ebook: https://www.researchgate.net/profile/Roland-Scholz-3/publication/257927023_Didactics_of_Mathematics_as_a_Scientific_Discipline/links/5c22006a92851c22a3445347/Didactics-of-Mathematics-as-a-Scientific-Discipline.pdf.

[18] Asiala, M. and Brown, A. and DeVries, D. and Dubinsky, E. and Mathews, D. and Thomas, K. (1996). A framework for research and curriculum development in undergraduate mathematics education. *Research in Collegiate Mathematics Education II, CBMS Issues in Mathematics Eduction*, 6:1–32.

[19] Australian Education Research Organisation (2023). Benchmarking Performance: Future Directions for Australia's National Assessment Program. https://www.edresearch.edu.au/research/research-reports/benchmarking-performance-future-directions-nap. Accessed: 2025-04-17.

[20] Bain, M. and Swan, G. (2011). Technology-enhanced support for the transition to university mathematics. *International Journal of Mathematical Education in Science and Technology*, 42(7):843–853.

[21] Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., and Koedinger, K. (2008). Why Students Engage in “Gaming the System” Behavior in Interactive Learning Environments. *The Journal of Interactive Learning Research*, 19(2):185–224. https://www.learntechlib.org/p/24328.

[22] Barab, S. A. and Squire, K. D. (2004). Design-based research: Putting a stake in the ground. *The Journal of the Learning Sciences*, 13(1):1–14.

[23] Barnes, T., Stamper, J., Croy, M., and Lehman, L. (2008). Toward automatic hint generation for logic proof tutoring using historical student data. *International Journal of Artificial Intelligence in Education*, 18(1):39–71.

[24] Barrington, F. and Evans, M. (2014). Participation in Year 12 Mathematics 2008–2017. *Australian Mathematical Sciences Institute*. Available at https://amsi.org.au/?publications=year-12-mathematics-participation-in-australia-2008-2017.

[25] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). *Manifesto for Agile Software Development*. Agile Alliance. https://agilemanifesto.org/.

[26] Berliner, D. C. (1993). The 100 year journey of educational psychology from interest, to disdain, to respect for practice. In Fagan, T. K. and VandenBos, G. R., editors, *Exploring applied psychology: Origins and critical analyses*, pages 37–78. American Psychological Association, Washington, D.C.

[27] Berliner, D. C. (2006). Educational psychology: Searching for essence throughout a century of influence. In Alexander, P. A. and Winne, P. H., editors, *Handbook of Educational Psychology*, pages 3–27. Routledge, 2nd edition.

[28] Black, P. and Wiliam, D. (1998). Inside the black box: Raising standards through classroom assessment. *Phi Delta Kappan*, 80(2):139–148.

[29] Borwein, J. M. (1999). Talk given at Technology in Mathematics Education at the Secondary and Tertiary Level conference, Brock University, Canada, June 2-3,. In *Proceedings of the Annual Meeting*, volume 23, St. Catherine’s. https://eric.ed.gov/?id=ED445894.

[30] Borwein, J. M. (2016). The life of modern homo habilis mathematicus: Experimental computation and visual theorems. In Monaghan, J., Trouche, L., and Borwein, J. M., editors, *Tools and Mathematics: Instruments for Learning*, pages 23–90. Springer. online pdf https://content.e-bookshelf.de/media/reading/L-7605753-7c1a00820a.pdf.

[31] Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101.

[32] Brock, R. (2020). Connectionism—Edward Thorndike. In Akpan, B. and Kennedy, T. J., editors, *Science Education in Theory and Practice: An Introductory Guide to Learning Theory*, Springer Texts in Education, pages 101–112. Springer.

[33] Brousseau, G. (2002). Theory of Didactical Situations in Mathematics: Didactique des Mathématiques, 1970–1990. In Balacheff, N., Cooper, M., Sutherland, R., and Warfield, V., editors, *Encyclopedia of Mathematics Education*. Springer.

[34] Brousseau, G. and Warfield, V. (2020). Didactic Situations in Mathematics Education. In Lerman, S., editor, *Encyclopedia of Mathematics Education*, pages 206–213. Springer International Publishing, Cham.

[35] Brown, J. S. and Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2(2):155–192. https://doi.org/10.1207/s15516709cog0202_4.

[36] Brown, J. S., Burton, R. R., and de Kleer, J. (1978). Diagnosing bugs in student algebra programs. In *Cognitive Science Society Conference*.

[37] Brown, T. and McNamara, O. (2011). *How Teachers Learn: A Review of Research*, pages 31–48. Springer Netherlands, Dordrecht. https://doi.org/10.1007/978-94-007-0554-8_3.

[38] Bruner, J. S. (1966). *Toward a Theory of Instruction*. Harvard University Press.

[39] Bruner, J. S. (1976). *The Process of Education: Revised Edition*. Harvard University Press, Cambridge, MA.

[40] Brusilovsky, P. and Millán, E. (2007). User models for adaptive hypermedia and adaptive educational systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, pages 3–53. Springer Berlin Heidelberg, Berlin, Heidelberg.

[41] Burton, R. R. and Brown, J. S. (1982). An Investigation of Computer Coaching for Informal Learning Activities. In Sleeman, D. and Brown, J. S., editors, *Intelligent Tutoring Systems*, pages 79–98. Academic Press, London.

[42] Buteau, C., €, N., Jarvis, D., and Lavicza, Z. (2010). Integrating computer algebra systems in post-secondary mathematics education: Preliminary results of a literature review. *International Journal for Technology in Mathematics Education*, 16:475–486.

[43] Carr, B. and Goldstein, I. (1977). Overlays: A theory of modelling for computer aided instruction. 2004 https://www.researchgate.net/publication/37596637_Overlays_A_Theory_of_Modelling_for_Computer_Aided_Instruction and 1977 https://dspace.mit.edu/handle/1721.1/5772.

[44] Cherry, K. (2023). Life, Theories, and Influence of Lev Vygotsky. url: https://www.verywellmind.com/lev-vygotsky-biography-2795533. Accessed: 2020-04-16.

[45] Chi, M. T., Bassok, M., Lewis, M., Reimann, P., and Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13(2):145–182.

[46] Chipman, S. (1993). Gazing Once More Into the Silicon Chip: Who's Revolutionary Now? In Lajoie, S. and Derry, S., editors, *Computers as Cognitive Tools*, pages 341–367. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[47] Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

[48] Clark, M. and Lovric, M. (2009). Understanding secondary–tertiary transition in mathematics. *International Journal of Mathematical Education in Science and Technology*, 40(6):755–766.

[49] Clark, V. L. P., Creswell, J. W., Green, D. O., and Shope, R. J. (2008). Mixing quantitative and qualitative approaches: An introduction to emergent mixed methods research. In Hesse-Biber, S. N. and Leavy, P., editors, *Handbook of emergent methods*, pages 363–387. The Guildford Press.

[50] Cockcroft, W. (1982). *Mathematics Counts: Report of the Committee of Inquiry into the Teaching of Mathematics in Schools*. HMSO, London.

[51] Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. Routledge, 2nd edition.

[52] Collins, J., Greer, J., and Huang, S. (1996). Adaptive assesment using granularity hierarchies and Bayesian nets. In Frasson, C., Gauthier, G., and Lesgold, A., editors, *Proceedings of the Third International Conference: ITS*, pages 569–577. Springer.

[53] Conati, C., Gertner, A., and Vanlehn, K. (2002). Using bayesian networks to manage uncertainty in student modeling. *User Modelling and User-Adapted Interaction*, 12(4):371–417.

[54] Copeland, B. J. (2004). *The Essential Turing*. Oxford University Press, Oxford. Includes a selection of Alan Turing's most important writings.

[55] Corbett, A. and Anderson, J. (1992). LISP intelligent tutoring system: Research in skill acquisition. In Larkin, J. and Chabay, R., editors, *Computer-Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches*, pages 73–109, Hillsdale, New Jersey. Lawrence Erlbaum Associates.

[56] Corbett, A. T. and Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278.

[57] Corbett, A. T., Koedinger, K., and Hadley, W. S. (2001). Cognitive tutors: From the research classroom to all classrooms. In Goodman, P. S., editor, *Technology enhanced learning: Opportunities for change*, page 235–263. Lawrence Erlbaum Associates Publishers.

[58] Corbett, A. T., Koedinger, K. R., and Anderson, J. R. (1997). Chapter 37 - intelligent tutoring systems. In Helander, M. G., Landauer, T. K., and Prabhu, P. V., editors, *Handbook of Human-Computer Interaction*, pages 849–874. North-Holland, Amsterdam, second edition.

[59] Cornu, B. (1991). Limits. In Tall, D., editor, *Advanced Mathematical Thinking*, pages 153–166. Springer Netherlands, Dordrecht.

[60] Cosyn, E., Doble, C., Falmagne, J.-C., Lenoble, A., Thiéry, N., and Uzun, H. (2013). Assessing mathematical knowledge in a learning space. In Falmagne, J.-C., Albert, D., Doble, C., Eppstein, D., and Hu, X., editors, *Knowledge Spaces: Applications in Education*, pages 27–50. Springer Berlin Heidelberg, Berlin, Heidelberg.

[61] Coupland, M. (2004). *Learning with new tools*. PhD thesis, University of Wollongong.

[62] Creswell, J.W. (2012). *Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research.* Pearson, 4th edition.

[63] Dabbagh, N. (2005). Pedagogical Models for E-Learning: A Theory-Based Design Framework. *International Journal of Technology in Teaching and Learning*, 1(1):25–44.

[64] De Guzmán, M., Hodgson, B. R., Robert, A., and Villani, V. (1998). Difficulties in the passage from secondary to tertiary education. *Documenta Mathematica*, pages 747–762.

[65] Design-Based Research Collective (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher*, 32(1):5–8.

[66] Desmarais, M. C. and Baker, R. S. (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1–2):9–38.

[67] Dewey, J. (1966). *Democracy and Education: An Introduction to the Philosophy of Education.* The Free Press, New York, London. First published by Macmillan, New York, in 1916.

[68] Doignon, J.-P. and Falmagne, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, 23(2):175–196.

[69] Dorier, J.-L. and Sierpinska, A. (2001). Research into the Teaching and Learning of Linear Algebra. In Holton, D., Artigue, M., Kirchgräber, U., Hillel, J., Niss, M., and Schoenfeld, A., editors, *The Teaching and Learning of Mathematics at University Level: An ICMI Study*, pages 255–273. Springer Netherlands, Dordrecht.

[70] Driscoll, M. P. (2005). *Psychology of Learning for Instruction*. Pearson/Merrill Prentice Hall, Boston, 3rd edition.

[71] Dubinsky, E. and Leron, U. (1994). Groups. In Dubinsky, E. and Leron, U., editors, *Learning Abstract Algebra with ISETL*, pages 39–82. Springer New York, New York, NY.

[72] Dubinsky, E. and Mcdonald, M. A. (2001). APOS: A Constructivist Theory of Learning in Undergraduate Mathematics Education Research. In Holton, D., Artigue, M., Kirchgräber, U., Hillel, J., Niss, M., and Schoenfeld, A., editors, *The Teaching and Learning of Mathematics at University Level: An ICMI Study*, pages 275–282. Springer Netherlands, Dordrecht.

[73] Edwards, D. and Mercer, N. (2013). *Common knowledge*. Routledge Revivals. Routledge, London, England. https://doi.org/10.4324/9780203095287.

[74] Efron, B. and Tibshirani, R. J. (1994). *An Introduction to the Bootstrap (1st ed.)*. Chapman and Hall/CRC.

[75] Engelbrecht, J. and Harding, A. (2005). Teaching undergraduate mathematics on the internet: Part 1—technologies and taxonomy. *Educational Studies in Mathematics*, 58(2):235–252.

[76] Ernest, P. (1991). *The Philosophy of Mathematics Education*. Falmer Press, London.

[77] Falmagne, J.-C., Cosyn, E., Doignon, J.-P., and Thiery, N. (2003). The assessment of knowledge, in theory and in practice. In *IEMC '03 Proceedings. Managing Technologically Driven Organizations: The Human Side of Innovation and Change (IEEE Cat. No.03CH37502)*, pages 609–615. https://ieeexplore.ieee.org/abstract/document/1245109.

[78] Field, A. (2013). *Discovering Statistics Using IBM SPSS Statistics*. SAGE Publications, London, 4th edition.

[79] Fischbein, E. (1987). *Intuition in Science and Mathematics: An Educational Approach*. Reidel.

[80] Gertner, A. S. and VanLehn, K. (2000). Andes: A coached problem solving environment for physics. In Gauthier, G., Frasson, C., and VanLehn, K., editors, *Intelligent Tutoring Systems*, pages 133–142, Berlin, Heidelberg. Springer.

[81] Gordon, S. and Nicholas, J. (2012). Students' conceptions of mathematics bridging courses. *Journal of Further and Higher Education*, 37:1–17.

[82] Greeno, J. G. (1988). Situativity and cognition: Learning as situated activity. In Schoenfeld, A. H., editor, *Cognitive Science and Mathematics Education*, pages 11–42. Lawrence Erlbaum Associates, Hillsdale, NJ. https://www.routledge.com/Cognitive-Science-and-Mathematics-Education/Schoenfeld/p/book/9780805800579.

[83] Greeno, J. G., Collins, A., Beranek, B., and Resnick, L. B. (1996). Cognition and learning. In Berliner, D. and Calfee, R., editors, *Handbook of educational psychology*, pages 15–46. Macmillan. https://www.researchgate.net/publication/233896256_Cognition_and_learning in Berliner, D.C., Calfee, R.C. (Eds.). (2004). Handbook of Educational Psychology (1st ed.). Routledge. https://doi.org/10.4324/9780203053874.

[84] Greer, J. and McCalla, G. (1994). Student Modelling: The Key to Individualised Knowledge-Based Instruction. In Greer, J. E. and McCalla, G. I., editors, *NATO ASI Series F: Computer and Systems Sciences*, volume 125 of *NATO ASI Ser., Ser. F, Comput. Syst. Sci.*, pages 3–35. Springer-Verlag. Also availale at Google Books: https://books.google.com.au/books?id=xNWoCAAAQBAJ.

[85] Hagerty, G. and Smith, S. (2005). Using the web-based interactive software aleks to enhance college algebra. *Mathematics and Computer Education*, 39(3):183–194. https://www.learntechlib.org/p/67912/.

[86] Hammad, R., Khan, Z., Safieddine, F., and Ahmed, A. (2020). A review of learning theories and models underpinning technology-enhanced learning artefacts. *World Journal of Science, Technology and Sustainable Development*, 17(4):341–354.

[87] Harel, G. and Trgalova, J. (1996). Higher Mathematics Education. In Bishop, A., Clements, K., Keitel, C., Kilpatrick, C., and Laborde, C., editors, *International Handbook of Mathematics Education*, pages pp. 675–700. Kluwer, Dordrecht, The Netherlands.

[88] Heffernan, N. T., Botelho, A., Baral, S., Erickson, J. A., and Benachamardi, P. (2023). Leveraging natural language processing to support automated assessment and feedback for student open responses in mathematics. *Journal of Computer Assisted Learning*, 39(3):823–840. https://doi.org/10.1111/jcal.12793.

[89] Highsmith, J. (2001). *Agile Software Development Ecosystems*. Addison-Wesley. must requeust access https://www.researchgate.net/publication/228090862_Agile_Software_Development_Ecosystem.

[90] Hillel, J. (2001). Trends in curriculum. In Holton, D., Artigue, M., Kirchgräber, U., Hillel, J., Niss, M., and Schoenfeld, A., editors, *The Teaching and Learning of Mathematics at University Level: An ICMI Study*, pages 59–69. Springer Netherlands, Dordrecht. https://doi.org/10.1007/0-306-47231-7_7 and online pdf https://link.springer.com/content/pdf/10.1007/0-306-47231-7.pdf.

[91] Hochmuth, R., Broley, L., and Nardi, E. (2021). Transitions to, across and beyond university. In Hochmuth, R., Broley, L., and Nardi, E., editors, *Research and Development in University Mathematics Education: Overview Produced by the International Network for Didactic Research in University Mathematics*, pages 250–271. Routledge, London.

[92] Hofer, B. K. and Pintrich, P. R. (1997). The development of epistemological theories: Beliefs about knowledge and knowing and their relation to learning. *Review of Educational Research*, 67(1):88–140.

[93] Holt, P., Dubs, S., Jones, M., and Greer, J. (1994). The State of Student Modeling. In Greer, J. and McCalla, G., editors, *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, pages 3–35. Springer-Verlag, Berlin. https://link.springer.com/book/10.1007/978-3-662-03037-0.

[94] Holton, D., Artigue, M., Kirchgräber, U., Hillel, J., Niss, M., and Schoenfeld, A., editors (2001). *The teaching and learning of mathematics at university level*. New ICMI Study Series. Springer, Dordrecht, Netherlands, 2001 edition. https://doi.org/10.1007/0-306-47231-7 online pdf https://link.springer.com/content/pdf/10.1007/0-306-47231-7.pdf.

[95] Hourigan, M. and O'Donoghue, J. (2007). Mathematical under-preparedness: the influence of the pre-tertiary mathematics experience on students' ability to make a successful transition to tertiary level mathematics courses in ireland. *International Journal of Mathematical Education in Science and Technology*, 38(4):461–476. https://doi.org/10.1080/00207390601129279.

[96] Hoyles, C., Newman, K., and Noss, R. (2001). Changing patterns of transition from school to university mathematics. *International Journal of Mathematical Education in Science and Technology*, 32(6):829–845.

[97] International Commission on Mathematical Instruction (2025). ICMI – International Commission on Mathematical Instruction. https://www.mathunion.org/icmi. Accessed: 2025-04-14.

[98] Jaworski, B. and Potari, D. (2009). Bridging the macro- and micro-divide: Using an activity theory model to capture sociocultural complexity in mathematics teaching and its development. *Educational Studies in Mathematics*, 72(2):219–236. https://www.jstor.org/stable/40284619.

[99] Jonassen, D. H. (1991). Evaluating Constructivist Learning. *Educational Technology*, 31(9):28–33.

[100] Joomla Project (2024). Joomla! documentation. https://docs.joomla.org/Main_Page. Accessed: 2025-06-12.

[101] Kaput, J. J. (1992). Technology and mathematics education. In Grouws, D. A., editor, *Handbook of Research on Mathematics Teaching and Learning*, pages 515–556. National Council of Teachers of Mathematics.

[102] Keppel, G. and Wickens, T. D. (2004). *Design and Analysis: A Researcher's Handbook*. Pearson Education, Upper Saddle River, NJ, 4th edition.

[103] Keynes, H. B. and Olson, A. M. (2001). Professional Development for Changing Undergraduate Mathematics Instruction. In Holton, D., Artigue, M., Kirchgräber, U., Hillel, J., Niss, M., and Schoenfeld, A., editors, *The Teaching and Learning of Mathematics at University Level: An ICMI Study*, pages 113–126. Springer Netherlands, Dordrecht.

[104] Kieran, C. (1981). Concepts associated with the equality symbol. *Educational Studies in Mathematics*, 12(3):317–326.

[105] Kimmel, S., Kimmel, H., and Deek, F. (2003). The common skills of problem solving: From program development to engineering design. *International Journal of Engineering Education*, 19(6):810–817.

[106] Koedinger, K. and Anderson, J. (1993). Reifying Implicit Planning in Geometry: Guidelines for Model-Based Intelligent Tutoring System Design. In Lajoie, S. and Derry, S., editors, *Computers as Cognitive Tools*, pages 15–45. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[107] Koedinger, K., Anderson, J., Hadley, W., and Mark, M. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8:30–43.

[108] Koedinger, K. R. and Corbett, A. T. (2006). Cognitive Tutors: Technology bringing learning sciences to the classroom. In *The Cambridge Handbook of the Learning Sciences*, pages 61–78. Cambridge University Press.

[109] Koedinger, K. R., D'Mello, S., McLaughlin, E. A., Pardos, Z. A., and Rosé, C. P. (2015). Data mining and education. *Wiley Interdisciplinary Reviews: Cognitive Science*, 6(4):333–353.

[110] Koutsojannis, C., Prentzas, J., and Hatzilygeroudis, I. (2001). A web-based intelligent tutoring system teaching nursing students fundamental aspects of biomedical technology. In *Proceedings of the 23rd Annual International Conference of the IEEE*, volume 4, pages 4024–4027. https://www.researchgate.net/publication/224055935_

A_web-based_intelligent_tutoring_system_teaching_nursing_students_fundamental_aspects_of_biomedical_technology or https://ieeexplore.ieee.org/document/1019728.

[111] Krogh, L. (2006). New challenges for Teaching at Universities – A Professional Strategy for Teaching in the Light of Current Social Demands. *Danish Journal of University Education*, 2. As noted in thesis this article was translated in Google: https://tidsskrift.dk/dut/article/view/5658.

[112] Langtry, T., Nicorovici, N., and Zinder, Y. (2011). Interactive self paced learning using Mathematica. In Joubert, M., Clark-Wilson, A., and McCabe, M., editors, *Enhancing Mathematics Education Through Technology: Proceedings of the International Conference on Technology in Mathematics Teaching 10*, pages 203–208. University of Portsmouth.

[113] Lock, R. H., Lock, P. F., Morgan, K. L., Lock, E. F., and Lock, D. F. (2021). *Statistics: Unlocking the power of Data*. Hoboken, NJ: Wiley.

[114] Mason, J. (2002). *Researching your own practice: The discipline of noticing*. Routledge.

[115] Maxima (2021). Maxima, a computer algebra system. version 5.45.1. https://sourceforge.net/p/maxima/wiki/FAQ/#.

[116] McCalla, G. I. (1992). The central importance of student modelling to intelligent tutoring. In Costa, E., editor, *New Directions for Intelligent Tutoring Systems*, pages 107–131, Berlin, Heidelberg. Springer.

[117] McCalla, G. I. (1996). Student Modelling: A Crucible for Research. In Liao, T. T., editor, *Advanced Educational Technology: Research Issues and Future Potential*, pages 149–164, Berlin, Heidelberg. Springer.

[118] Melis, E., Andres, E., Budenbender, J., Frischauf, A., Goguadze, G., Libbrecht, P., Pollet, M., and Ullrich, C. (2001). Activemath: A generic and adaptive web-based learning environment. *Artificial Intelligence in Education*, 12. https://www.researchgate.net/publication/32229452_ActiveMath_A_Generic_and_Adaptive_Web-Based_Learning_Environment.

[119] Mercer, N. (1995). *The Guided Construction of Knowledge: Talk Among Teachers and Learners*. Multilingual Matters.

[120] Millán, E., Loboda, T., and Pérez-De-La-Cruz, J. L. (2010). Bayesian networks for student model engineering. *Computers & Education*, 55(4):1663–1683. https://www.sciencedirect.com/science/article/pii/S0360131510002022.

[121] Mills, N. J. D. (2021). ALEKS constructs as predictors of high school mathematics achievement for struggling students. *Heliyon*, 7(6):e07345. https://pmc.ncbi.nlm.nih.gov/articles/PMC8239733/.

[122] Mitchell, P. D. and Grogono, P. D. (1993). Modelling techniques for tutoring systems. *Computers & Education*, 20(1):55–61. Special Issue Selected Papers of the Sixth International Annual Conference of the PEG Group.

[123] Mitrovic, A. (1998). Learning SQL with a computerized tutor. *SIGCSE Bull.*, 30(1):307–311. https://dl.acm.org/doi/10.1145/273133.274318.

[124] Mitrovic, A. (2003). An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education*, 13(2-4):171–195. https://telearn.hal.science/hal-00197316/file/mitrovic03.pdf.

[125] Mitrovic, A. (2010). Modeling domains and students with constraint-based modeling. In Nkambou, R., Bourdeau, J., and Mizoguchi, R., editors, *Advances in Intelligent Tutoring Systems*, pages 63–80. Springer Berlin Heidelberg, Berlin, Heidelberg. https://link.springer.com/chapter/10.1007/978-3-642-14363-2_4.

[126] Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J., and McGuigan, N. (2009). Aspire: An authoring system and deployment environment for constraint-based tutors. *International Journal of Artificial Intelligence in Education*, 19(2):155–188. https://journals.sagepub.com/doi/10.3233/IRG-2009-19%282%2903.

[127] Mogens, N. (1997). ICMI Bulletin of the International Commission on Mathematical Instruction No. 43 December 1997. https://www.mathunion.org/fileadmin/ICMI/files/Digital_Library/Bulletin2/ICMI_43_12_1997.pdf. Accessed: (Wednesday, 11th, May, 2016).

[128] Murray, T. (2003). An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art. In Murray, T., Blessing, S. B., and Ainsworth, S., editors, *Authoring Tools for Advanced Technology Learning Environments: Toward Cost-Effective Adaptive, Interactive and Intelligent Educational Software*, pages 491–544. Springer Netherlands, Dordrecht. https://link.springer.com/book/10.1007/978-94-017-0819-7.

[129] Murray, T. and Arroyo, I. (2002). Toward Measuring and Maintaining the Zone of Proximal Development in Adaptive Instructional Systems. In Cerri, S. A., Gouardères, G., and Paraguaçu, F., editors, *Intelligent Tutoring Systems*, pages 749–758, Berlin, Heidelberg. Springer Berlin Heidelberg.

[130] Nair, C. S. and Fisher, D. L. (2000). Transition from senior secondary to higher education: A learning environment perspective. *Research in Science Education*, 30(4):435–450.

[131] National Council of Teachers of Mathematics (1989). *Curriculum and Evaluation Standards for School Mathematics*. National Council of Teachers of Mathematics, Reston, VA.

[132] Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA.

[133] Newell, A. and Simon, H. A. (1972). *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, NJ.

[134] Nicol, D. and Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2):199–218.

[135] Nowell, L. S., Norris, J. M., White, D. E., and Moules, N. J. (2017). Thematic Analysis: Striving to Meet the Trustworthiness Criteria. *International Journal of Qualitative Methods*, 16(1):1–13.

[136] NSW Education Standards Authority (2017). Mathematics extension 1 stage 6 syllabus. https://educationstandards.nsw.edu.au/wps/portal/nesa/11-12/stage-6-learning-areas/stage-6-mathematics/mathematics-extension-1-2017. Accessed: 2025-06-02.

[137] NSW Education Standards Authority (2022). Mathematics K–10 Syllabus (2022). https://curriculum.nsw.edu.au/learning-areas/mathematics/mathematics-k-10-2022/overview. Stages 4 and 5 (Years 7–10) implementation from 2024.

[138] Ohlsson, S. (1994). Constraint-based student modeling. In Greer, J. E. and McCalla, G. I., editors, *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, pages 167–189, Berlin, Heidelberg. Springer.

[139] Ohlsson, S. (1996.). Learning from performance errors. *Psychological Review*, 103:241–262.

[140] Page, S. and Clark, J. (2010). Experiences of learning and teaching mathematics: Using activity theory to understand tensions in practice. In *Proceedings of the 33rd Annual Conference of the Mathematics Education Research Group of Australasia*, pages 462–469, Fremantle, Australia. MERGA.

[141] Pan, L., Albalak, A., Wang, X., and Wang, W. Y. (2023). Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. *arXiv preprint arXiv:2305.12295*.

[142] Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.

[143] Phillips, D. C. (1995). The good, the bad, and the ugly: The many faces of constructivism. *Educational Researcher*, 24(7):5–12.

[144] Piaget, J. (1968). *Six Psychological Studies. Anita Tenzer (Trans.)*. New York: Vintage Books.

[145] Piaget, J. (2001). *Studies in Reflecting Abstraction*. Psychology Press, London.

[146] Piaget, J. and Inhelder, B. (1972). *The Psychology of the Child*. Basic Books.

[147] Pierce, R. and Stacey, K. (2004). A framework for monitoring progress and planning teaching towards numeracy goals. *International Journal of Computers for Mathematical Learning*, 9:59–93.

[148] Plano Clark, V. L. (2017). Mixed methods research. *The Journal of Positive Psychology*, 12(3):305–306.

[149] Rasmussen, C., Giraldo, V., and Khakbaz, A. S. (2016). *Research on teaching and learning mathematics at the tertiary level*. ICME-13 Topical Surveys. Springer International Publishing, Cham, Switzerland, 1 edition. https://link.springer.com/content/pdf/10.1007/978-3-319-41814-8.pdf.

[150] Razzaq, L., Feng, M., Nuzzo-Jones, G., Heffernan, N. T., Koedinger, K. R., Junker, B., Ritter, S., Knight, A., Aniszczyk, C., Choksey, S., Livak, T., Mercado, E., Turner, T. E., Upalekar, R., Walonoski, J. A., Macasek, M. A., and Rasmussen, K. P. (2005). The assistment project: Blending assessment and assisting. In Looi, C. K., McCalla, G., Bredeweg, B., and Breuker, J., editors, *Proceedings of the 12th International Conference on Artificial Intelligence In Education*, pages 555–562. IOS Press, Amsterdam. http://pact.cs.cmu.edu/koedinger/pubs/Razzag,%20Feng,%20Nuzzo-Jones,%20Heffernan,%20Koedinger,%20Junker,%20Ritter,%20Knight,%20Aniszczyk,Choksey,Livak,Mercado,Turner,Upalekar,Walonoski,Mac.pdf.

[151] Reeves, T. C. (2006). Design research from the technology perspective. In *Educational Design Research*, pages 52–66. Routledge.

[152] RMIT University (2023). NAPLAN scores have dropped – and they look different this year. https://www.rmit.edu.au/news/media-releases-and-expert-comments/2023/aug/naplan. Accessed: 2025-04-17.

[153] Rosenberg, R. (1987). A Critical Analysis of Research on Intelligent Tutoring Systems. *Educational Technology*, 27(11):7–13.

[154] Rowland, T. (2009). *Developing primary mathematics teaching : reflecting on practice with the knowledge quartet*. SAGE, Los Angeles. https://search.worldcat.org/title/228362991.

[155] Sack, W., Soloway, E., and Weingrad, P. (1994). Re-Writing Cartesian Student Models. In Greer, J. and McCalla, G., editors, *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, pages 355–376, Berlin. Springer-Verlag. Also available at Google Books: https://books.google.com.au/books?id=xNWoCAAAQBAJ.

[156] Sadler, D. R. (1989). Formative assessment and the design of instructional systems. *Instructional Science*, 18(2):119–144.

[157] Sani, S. M. and Aris, T. N. M. (2014). Modeling Student/Tutor Components of Intelligent Tutoring Systems: A Review. In *Proceedings of the International Conference on Advances in Computer Science and Electronics Engineering CSEE 2014*, pages 98–102.

[158] Schneider, S. C. and De Meyer, A. (1991). Interpreting and responding to strategic issues: The impact of national culture. *Strategic Management Journal*, 12(4):307–320.

[159] Schoenfeld, A. and Sloane, A. H., editors (1994). *Mathematical thinking and problem solving*. Studies in mathematical thinking and learning. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, US.

[160] Schoenfeld, A. H. (2002). Purposes and methods of research in mathematics education. *Notices of the AMS*, 47:221–236.

[161] Sfard, A. (1991). On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin. *Educational Studies in Mathematics*, 22(1):1–36.

[162] Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1):153–189.

[163] Sierpinska, A. (1992). On understanding the notion of function. *MAA Notes:The Concept of Function: Aspects of Epistemology and Pedagogy*, 25.

[164] Skemp, R. (1976). Relational understanding and instrumental understanding. *Mathematics Teaching*, 77:20–26.

[165] Sleeman, D. and Brown, J.S. (1982). *Introduction: Intelligent Tutoring Systems*. New York Academic Press, London.

[166] Soloway, E., Frye, D., and Littman, D. C. (1988). The next wave of problems in its: Controllong the "user issues" of interface design and system evaluation. In Psotka, J., Massey, L. D., and Mutter, S. A., editors, *Intelligent tutoring systems: Lessons learned*. Psychology Press. https://books.google.com.au/books?id=sFtYUgHeEqUC.

[167] Soloway, E., Letovsky, S., Pinto, J., and Lampert, R. (1987). A cognitive analysis of a code inspection. In *Empirical Studies of Programmers: Second Workshop*, page 231–247. Ablex Publishing Corp., USA.

[168] Son, T. (2024). Intelligent tutoring systems in mathematics education: A systematic literature review using the substitution, augmentation, modification, redefinition model. *Computers*, 13(10):270. https://www.mdpi.com/2073-431X/13/10/270.

[169] SOWISO (2023). SOWISO - Digital Math Learning. https://www.sowiso.com.

[170] Steen, L. (1992). Steen: Twenty Questions That Deans Should Ask Their Mathematics Departments. *Bulletin of the American Association of Higher Education*, 44:3–6.

[171] Sullivan, G. M. and Artino, A. R. (2013). Analyzing and interpreting data from likert-type scales. *Journal of Graduate Medical Education*, 5(4):541—542.

[172] Suppes, P. (1967). Some theoretical models for mathematics learning. *Journal of Research & Development in Education*, 1(1):5–22.

[173] Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4(4):295–312.

[174] Swinburne University of Technology (2024). New analysis reveals the brain drought putting Australia's STEM future at risk. Accessed: 2025-05-15.

[175] Swinyard, C. and Larsen, S. (2012). Coming to Understand the Formal Definition of Limit: Insights Gained From Engaging Students in Reinvention. *Journal for Research in Mathematics Education*, 43(4):465–493.

[176] Tall, D. (1991). The psychology of advanced mathematical thinking. In Tall, D., editor, *Advanced Mathematical Thinking*, pages 3–21. Springer.

[177] Tall, D. (1992). The Transition to Advanced Mathematical Thinking: Functions, Limits, Infinity, and Proof. *Handbook of Research on Mathematics Teaching and Learning*, 20:495–511.

[178] Tall, D. (2004). Thinking through three worlds of mathematics. *Proceedings of the 28th Conference of the International Group for the Psychology of Mathematics Education*, 4:281–288.

[179] Teach for Australia (2024). 167 new teachers added to address chronic shortages, with 40 STEM-qualified to empower future innovators. https://teachforaustralia.org/167-new-teachers-added-to-address-chronic-shortages-with-40-stem-qualified-to-empower-future-inn Accessed: 2025-05-15.

[180] The Australian (2024). Physics students in 'catastrophic' decline in senior high school. https://www.theaustralian.com.au/higher-education/physics-students-in-catastrophic-decline-in-senior-high-school/news-story/3d9960b5047b9419057ab99c3e28b7df. Accessed: 2025-05-15.

[181] The Educator (2023). New report sounds the alarm on maths enrolments. urlhttps://www.theeducatoronline.com/k12/news/new-report-sounds-the-alarm-on-maths-enrolments/284598. Accessed: 2025-05-15.

[182] The Educator (2024). Australia faces decline in domestic STEM enrolments. https://www.theeducatoronline.com/k12/news/australia-faces-decline-in-domestic-stem-enrolments/286332. Accessed: 2025-05-15.

[183] Thomas, M. O. J. (2008). The transition from school to university and beyond. *Mathematics Education Research Journal*, 20(2):1–4. https://link.springer.com/article/10.1007/BF03217473.

[184] Timms, M. J. (2023). Reversing Australia's downward spiral in STEM. url: https://deved.org/gb/discover/article/reversing-australias-downward-spiral-in-stem. Accessed: 2025-05-15.

[185] Uhr, L. (1969). Teaching machine programs that generate problems as a function of interaction with students. In *Proceedings of the 1969 24th National Conference*, ACM '69, page 125–134, New York, NY, USA. Association for Computing Machinery.

[186] University of Technology Sydney (2025). Records and archives hub. https://studentutsedu.sharepoint.com/sites/Records-and-archives-hub. Access restricted to UTS students and staff.

[187] Valero, P., Christensen, O. R., and Skovsmose, O. (2009). Introduction: The multi-layered transitions of knowledge production and university education in science and mathematics. In Skovsmose, O., Valero, P., and Christensen, O. R., editors, *University Science and Mathematics Education in Transition*, pages 1–13. Springer US, Boston, MA.

[188] VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3):227–265.

[189] von Glasersfeld, E. (1995). *Radical Constructivism: A Way of Knowing and Learning*. Falmer Press.

[190] Wang, F. and Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational Technology Research and Development*, 53(4):5–23.

[191] Watson, J. B. (1924). *Behaviorism*. New York: People's Institute Publishing Company.

[192] Wenger, E. (1987.). *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge.* Morgan Kaufmann Publishers, Inc, Los Altos, CA.

[193] Wilson, B. G., editor (1996). *Constructivist Learning Environments: Case Studies in Instructional Design*. Educational Technology Publications, Englewood Cliffs, NJ.

[194] Winne, P. H. and Perry, N. E. (2000). Chapter 16 - measuring self-regulated learning. In Boekaerts, M., Pintrich, P. R., and Zeidner, M., editors, *Handbook of Self-Regulation*, pages 531–566. Academic Press, San Diego.

[195] Wolfram, S. (2020). *An Elementary Introduction to the Wolfram Language*. Wolfram Media, third edition.

[196] Wolfram Research (2025). Wolfram Educational Technology: Quezzio. https://www.wolfram.com/educational-technology/quezzio/. Accessed: 2025-04-17.

[197] Wood, L. N. (2001). The Secondary-Tertiary Interface. In Holton, D., Artigue, M., Kirchgräber, U., Hillel, J., Niss, M., and Schoenfeld, A., editors, *The Teaching and Learning of Mathematics at University Level: An ICMI Study*, pages 87–98. Springer Netherlands, Dordrecht.

[198] Wood, L. N. and Breyer, Y. A., editors (2017). *Success in Higher Education: Transitions to, within and from University*. Springer Singapore, Singapore, 1 edition. web https://doi.org/10.1007/978-981-10-2791-8 and pdf https://link.springer.com/content/pdf/10.1007/978-981-10-2791-8.pdf.

[199] Wood, L. N., Petocz, P., and Reid, A. (2012). *Becoming a Mathematician: An International Perspective*. Mathematics Education Library. Springer Dordrecht, Dordrecht, 1 edition. web https://doi.org/10.1007/978-94-007-2984-1 and pdf https://link.springer.com/content/pdf/10.1007/978-94-007-2984-1.pdf.

[200] Woolf, B.P. (2007). *Building Intelligent Interactive Tutors*. Morgan Kaufmann.

[201] Xu, J., Fei, H., Pan, L., Liu, Q., Lee, M.-L., and Hsu, W. (2024). Faithful logical reasoning via symbolic chain-of-thought. In Ku, L.-W., Martins, A., and Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13326–13336. Association for Computational Linguistics.

[202] Yaron, D., Karabinos, M., Lange, D., Greeno, J. G., and Leinhardt, G. (2010). The chemcollective—virtual labs for introductory chemistry courses. *Science*, 328(5978):584–585.

[203] Yerushalmy, M. (2009). Epistemological discontinuities and cognitive hierarchies in technology-based algebra learning. In Dreyfus, T., Schwarz, B., and Hershkowitz, R., editors, *Transformation of Knowledge through Classroom Interaction*. Routledge.

[204] Zakariya, Y. F., Nilsen, H. K., Goodchild, S., and BjÃžrkestÃžl, K. (2022). Self-efficacy and approaches to learning mathematics among engineering students: empirical

evidence for potential causal relations. *International Journal of Mathematical Education in Science and Technology*, 53(4):827–841. https://doi.org/10.1080/0020739X.2020.1783006.

[205] Zimmermann, H. (1980). Osi reference model–the iso model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4):425–432.

[206] Zinder, Y., Nicorovici, N., and Langtry, T. (2010). Mathematica based platform for self-paced learning. In *EDULEARN10 Proceedings*, 2nd International Conference on Education and New Learning Technologies, pages 323–330. IATED.