Mikailalsys Journal of
Advanced Engineering International

e-ISSN : 3046-9694
p-ISSN : 3046-8914

Index : Harvard, Boston, Sydney University, Dimensions, Scilit, Semantic, Google, etc

https://doi.org/10.58578/MJAEI.v3i3.12459

Comparative Effectiveness of Teaching Numerical Methods Using Chebyshev Polynomials and Conventional Techniques

Adeyemi Ademola Bepo[1] & Ebenezer Adewole Olatunji[2]
[1]Ladoke Akintola University of Technology, Oyo State, Nigeria
[2]Adekunle Ajasin University, Ondo State, Nigeria
bepoadeyemi59@gmail.com; ebenezerolatunji0404@gmail.com

Submitted:	Revised:	Accepted:	Published:
Sep 25, 2026	Oct 24, 2026	Nov 4, 2026	Nov 9, 2026

Abstract

Despite the widespread use of Chebyshev polynomials in numerical analysis, evidence of their instructional effectiveness in undergraduate mathematics remains limited. This study compared Chebyshev polynomial-based instruction, conventional teaching, and a combined instructional approach. A quasi-experimental design involved 60 undergraduate mathematics students assigned to three instructional groups. Data were collected using a validated structured questionnaire assessing conceptual understanding, problem-solving skills, learning attitudes, and perceived instructional effectiveness. Maple 18 was used to implement Chebyshev polynomial procedures and conduct numerical simulations. Educational outcomes were analyzed using descriptive statistics and one-way analysis of variance at the .05 significance level. The combined approach received the highest mean rating for conceptual understanding ($M = 3.48$, $SD = 0.58$), although differences among the three groups were not statistically significant, $F(2, 57) = 0.662$, $p = .520$, $\eta^2 = .023$, indicating a small effect size. The Chebyshev-based approach showed more favorable questionnaire outcomes for problem-solving and student engagement. Numerical experiments demonstrated close agreement between Chebyshev polynomial approximations and benchmark solutions, with negligible numerical errors and high computational accuracy and stability. These findings indicate the computational promise of Chebyshev polynomial methods and suggest potential

MJAEI Journal is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

instructional benefits, while providing limited evidence of differences in conceptual understanding across approaches. The study highlights the potential value of combining computational methods with conventional instruction and recommends considering spectral methods in undergraduate mathematics curricula, supported by professional development for lecturers.

Keywords: Chebyshev Polynomials; Numerical Methods Education; Conceptual Understanding; Problem-Solving Skills; Undergraduate Mathematics

Introduction

Numerical integration serves as an essential component of numerical analysis which offers structured methods for calculating definite integrals that scientists and engineers and mathematicians use in their work. The majority of real-world situations require numerical methods because they provide accurate and efficient solutions to problems when exact analytical solutions do not exist or take too long to compute (Quarteroni *et al.,2019*). At the undergraduate level, teaching numerical integration presents challenges, as students must develop conceptual understanding of discretization, error propagation, and algorithm stability while learning to implement computational solutions (Martín-Caraballo and Tenorio-Villalón, 2015; Owens, 2023). The traditional lecture-based teaching methods of educational institutions fail to provide adequate support for students to develop their problem-solving abilities while using contemporary computing resources.

Undergraduate numerical methods courses sit at the right in the middle of mathematics, engineering, and computational science programmes, because they bridge "theory" mathematics with scientific computing and, you know, real world problem solving. Still, a lot of students get stuck when they try to connect those abstract mathematical ideas with the actual code like implementation, and it can easily turn into procedural learning rather than real conceptual grasping. Lately, since higher education really pushes computational thinking, algorithmic reasoning, and digital competence, there's also more and more interest in teaching methods that actually pull students in through mathematical modelling, visualization, and approximation at the same time. Figuring out teaching strategies that improve conceptual understanding, computational precision, and overall problem solving skill, has become a major research priority in mathematics education. The new teaching methods which teachers use today require students to learn through active engagement with multiple programming tools. The learning approach enables students to

test algorithms which they can use to determine their learning through simulation testing and comparison testing of their work against actual solution testing. Chebyshev polynomials serve as an effective numerical approximation tool because they exhibit both strong convergence abilities and minimax error performance (Boyd, 2001; Preeti and Singh, 2023). The orthogonal polynomials enable precise integral and differential equation approximations which work with both simple and complex functions. The orthogonal polynomials provide essential educational materials for teaching computational learning concepts.

Numerical analysis research shows that Chebyshev-based methods provide multiple benefits yet researchers have not studied their teaching effectiveness within undergraduate programs. Researchers who study numerical methods teaching have mainly investigated traditional methods and general computational tools while they have paid little attention to spectral methods which include Chebyshev polynomials (Driscoll *et al.,2014*; Pleština *et al.,2021*). Curriculum design which includes Chebyshev-based teaching methods enhances problem-solving abilities and computational thinking while increasing student interest yet research which compares these methods to conventional teaching approaches remains limited.

The study evaluates how effective Chebyshev polynomial instruction improves teaching of numerical integration when compared to traditional methods and combined teaching approaches. The study assesses how different teaching methods affect students' ability to solve problems and their understanding of concepts and their viewpoint on learning. The research shows how advanced computational techniques can be used in undergraduate programs while developing evidence-based teaching methods for mathematics education. The study follows STEM education trends which focus on three main areas: technological progress and unified understanding and organized learning through systems (Owens 2023 and Tabieh 2024).

Research Gap

Existing studies on undergraduate numerical methods education have primarily examined the effectiveness of programming environments, computer algebra systems, visualization software, and digital learning platforms. In contrast, relatively little attention has been devoted to investigating whether advanced numerical approximation techniques themselves can serve as effective instructional strategies. Although Chebyshev polynomial methods have been extensively investigated for solving differential equations, interpolation,

and numerical integration, empirical evidence regarding their pedagogical effectiveness in improving students' conceptual understanding, computational thinking, problem-solving ability, and learning engagement remains scarce. Furthermore, few studies have compared Chebyshev-based instruction with conventional classroom teaching and blended instructional approaches within the same experimental framework. Consequently, there remains an important gap in mathematics education literature concerning how spectral approximation techniques can be integrated into undergraduate teaching to improve educational outcomes. This study addresses this gap by empirically evaluating the instructional effectiveness of Chebyshev polynomial-based teaching relative to conventional and blended instructional approaches.

Novelty and Contribution of the Study

This study basically has a twist: it takes Chebyshev polynomial approximation, which is usually treated as this kind of numerical analysis tool, and turns it into an instructional strategy, for teaching undergraduate numerical methods. Earlier work mostly stayed in the math lane things like properties, how convergence behaves, and whether computation is fast or efficient especially when Chebyshev polynomials are used for differential equations and approximation tasks. But here the focus is a bit more classrooms centered, like, how these polynomials can support students learning experiences in a higher education setting.

First, the paper offers an early empirical comparison of three different ways to teach—more conventional instruction, Chebyshev polynomial based teaching, and a blended approach that mixes both. Most prior studies looked at computational procedures on their own, like in isolation, rather than asking how teaching style changes the student. In this work, the instructional routes are examined in terms of how they affect students' conceptual grasp, their problem solving ability, their learning attitudes, and even how they judge whether the instruction feels effective.

Second, the study connects computational mathematics with mathematics education by showing how an advanced spectral approximation method can be reshaped into a classroom resource. The idea is that it can encourage computational thinking, stronger mathematical reasoning, better visualization skills, and more active learning. So, in a sense, it pushes Chebyshev polynomials beyond just algorithm building, and back into pedagogy, where they become useful for teaching rather than only for development.

Third, this research offers a kind of simultaneous evaluation of educational and numerical performance. In addition to checking students learning outcomes through validated educational measures and statistical analysis, the study also looks at the computational accuracy and stability of Chebyshev polynomial approximations via numerical experiments. So overall, this integrated evaluation gives a more-full picture of how well the instructional approach works both educationally and mathematically.

Finally, the study adds to the growing research area in technology-enhanced mathematics education by showing the actual teaching use of Chebyshev polynomial techniques, not just the algorithmic development side of it. The results give empirical evidence that spectral approximation methods can be integrated into undergraduate mathematics curricula, which in turn provides a workable framework where advanced computational techniques can strengthen conceptual grasp, computational thinking, and problem solving skills.

Accordingly, the study contributes to both computational mathematics and mathematics education by providing empirical evidence that advanced numerical approximation techniques can act not only as efficient computational algorithms, but also as practical pedagogical tools for improving undergraduate learning.

Contributions of the Study

This study makes the following contributions:

1. It introduces Chebyshev polynomial approximation as an instructional strategy rather than solely as a numerical solution technique.
2. It bridges computational mathematics and mathematics education by integrating spectral methods into undergraduate classroom teaching.
3. It provides empirical evidence on the influence of Chebyshev-based instruction on conceptual understanding, problem-solving ability, student engagement, and learning attitudes.
4. It demonstrates how advanced approximation techniques can be incorporated into undergraduate teaching while maintaining computational accuracy and conceptual clarity.
5. It proposes a blended instructional framework that combines conventional teaching methods with spectral approximation techniques to enhance computational thinking and active learning.

6. It provides a foundation for future curriculum development involving spectral methods in undergraduate numerical analysis courses.

Literature Review

Numerical Methods in Mathematics Education

Numerical methods are essential tools in applied mathematics because they enable users to find approximate solutions to problems that lack analytical solutions which include differential equations and integrals and interpolations (Quarteroni *et al., 2007*). At the undergraduate level, numerical methods courses aim to develop students' understanding of algorithm implementation, error estimation, and computational efficiency. Research shows that students have difficulty mastering abstract concepts which include convergence and stability and truncation errors, which hinders their ability to use techniques successfully (Martín-Caraballo and Tenorio-Villalón, 2015).Traditional teaching methods which include lectures and procedural problem-solving, require students to perform algorithmic computations, yet these approaches do not help students understand concepts or use contemporary computational tools (Owens, 2023).Orthogonal polynomials which include Legendre and Hermite polynomials, provide equivalent benefits for function approximation, but Chebyshev polynomials become more popular because they offer straightforward implementation with reduced error performance (Gautschi, 2004).Teaching staff can use spectral methods to help students evaluate algorithm efficiency and convergence results within a structured framework which supports both practical and theoretical educational outcomes (Trefethen, 2019).The combination of traditional teaching methods and Chebyshev-based techniques, which create a comprehensive learning atmosphere, will help students build problem-solving skills, while they acquire better understanding of concepts (Owens, 2023; Tabieh, 2024).

Current educational technology advancements have established computational learning as an effective teaching method. Students participate in learning activities through programming and visualization tools and simulation platforms to test numerical algorithms while seeing their results from different conditions (Cuevas *et al., 2023*). The research findings show that students who learn through computational learning develop better problem-solving abilities and conceptual comprehension and technology skills which help them understand how algorithms work and how precise numerical values are (Owens, 2023;

Pleština *et al., 2021*). Interactive learning environments enable students to connect theoretical calculus principles with actual use because they assist students in understanding abstract mathematical concepts and solving practical real-world problems.

Chebyshev polynomials serve as orthogonal polynomials which numerical analysts use to approximate functions and perform interpolation and numerical integration because of their ability to distribute errors efficiently and their quick convergence rate (Boyd, 2001; Preeti and Singh, 2023). The polynomials achieve their best performance when applied to functions that exhibit strong oscillations or intricate behavior because they minimize the highest potential approximation error. Researchers and teachers use spectral methods based on Chebyshev polynomials because these methods achieve both fast convergence and precise computation when solving differential equations (Driscoll *et al.,2014*; Trefethen, 2019).

Chebyshev polynomials have become a major research area in computational mathematics but their use for teaching undergraduate numerical methods remains underdeveloped. Existing research has mainly investigated the mathematical attributes and numerical characteristics of these polynomials without testing their effectiveness as teaching tools (Boyd, 2001; Preeti and Singh, 2023). The existing research demonstrates that Chebyshev-based visualization and computation methods improve student engagement and comprehension while empirical studies which measure their effectiveness against traditional approaches remain few in number (Driscoll *et al., 2014*; Pleština *et al., 2021*). Previous research demonstrated how Chebyshev polynomials improve computation efficiency but there has been insufficient research about their value for student learning in numerical methods courses. The limited research studies establish a direct comparison between Chebyshev-based teaching methods and standard educational practices together with blended learning strategies. The research investigates how different instructional approaches including Chebyshev teaching and standard education methods and their combination affect students' capability to grasp concepts and solve problems and their perspective towards numerical integration learning. The research study delivers proof-based recommendations which show how to implement modern computational techniques into mathematics education programs.

Brief discussion of Chebyshev polynomial

The Chebyshev polynomials, introduced by the Russian mathematician Pafnuty Chebyshev (1821–1894), are defined over the interval and can be expressed using trigonometric or recurrence relations. The first few polynomials are given as:

$$\cos\alpha + \cos\beta = 2\cos(\frac{\alpha+\beta}{2})\cos(\frac{\alpha-\beta}{2}) \quad (1)$$

Where

$$\alpha = (n+1)\theta)\ \beta = (n-1)\theta \quad (2)$$

By substitute equation (2) into (3) gives:

$$\cos(n+1)\theta + \cos(n-1)\theta = 2\cos(n\theta)\cos(\theta) \quad (3)$$

Introduce the definition $X = \cos(\theta)$; the Chebyshev polynomial of the first kind becomes:

$$T_0(x) = 1,\ T_1(x) = x \quad (4)$$

While the Chebyshev polynomial general formula is given as;

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad (5)$$

These polynomials possess the minimax property, meaning they minimize the maximum error between the approximating polynomial and the actual function, making them superior for interpolation and numerical integration (Amr *et al.,* 2017 and Authors, *2025*).

Computational Thinking in Mathematics Education

Computational thinking has pretty much turned into one of the key competencies people expect from undergrad students who study mathematics engineering and well other STEM areas. It's about the brain processes that show up when you write down a problem, build algorithmic approaches, notice patterns, cut a complicated task into smaller pieces, and also check whether computational steps actually make sense (Wing, 2006; Weintrop *et al., 2016*). In mathematics education, computational thinking kind of goes past simple computer programming since it pushes students toward methodical reasoning, abstraction, careful logical checking, and mathematical modelling abilities. Those skills can then be used on a bunch of different quantitative questions, not only the ones that look like "coding practice".

More recently, math education work keeps stressing that computational thinking should be woven into undergraduate courses, via teaching methods that pull students into designing algorithms, running tests on numerical procedures, making sense of computed outputs, and thinking back on whether the final solution is reliable. This is supposed to happen instead of just following ready-made routines (Grover and Pea, 2018; Lockwood *et al., 2024*). With these kinds of activities, students often get a deeper grasp, because they're not only pressing buttons they're actually observing how numerical algorithms behave , why approximation errors crop up, and how computational efficiency can change the overall quality of numerical results.

Chebyshev polynomial approximation can be a pretty decent instructional scaffold for building computational thinking, kind of like you let the students tinker instead of just staring at clean math. Instead of showing approximation methods as abstract formulas, a Chebyshev based teaching approach nudges students to look at things like polynomial approximation itself, the way convergence behaves, interpolation fidelity, and how error gets reduced, but mostly by running computational experiments. So the idea is students end up analyzing the mathematical structure, comparing approximation schemes, and then interpreting what the numbers are saying. That whole cycle really sits at the core of computational thinking even if it sounds a bit broader than "just" approximation.

Because of that, bringing Chebyshev polynomial techniques into undergraduate numerical methods instruction may help sharpen algorithmic reasoning, computational problem solving ability, and also mathematical judgement. These kinds of competencies are now more and more seen as key graduate outcomes in current STEM education, where mathematical knowledge is not only learned, but actually used with computational tools and defended through evidence based reasoning.

Technology-Enhanced Mathematics Instruction

Technology enhanced mathematics instruction has become a key part of contemporary higher education, because it allows students to dig into mathematical ideas using interactive computation visualization simulation, and also quick feedback. Instead of depending only on those classic lecture style lessons, technology supported learning environments make it easier for learners to actively nudge and investigate mathematical thoughts, try out numerical algorithms, and then look at what the computed outputs actually show. This kind of approach has been linked to improvements in conceptual understanding,

problem solving skill, and also the confidence to use mathematical techniques when facing real world situations (Borba *et al., 2023*; Engelbrecht *et al., 2023*).

In undergraduate numerical methods courses, there are many technological tools like Maple, MATLAB, Python, GeoGebra, and Mathematica, that let students carry out numerical procedures, visualize approximation behavior, examine computational errors, and contrast their numerical answers with analytical reference points. These computational spaces really lower the load from long manual work, and then students can spend more time on the deeper math principles behind numerical approximation, convergence, and whether an algorithm is efficient or not. Overall, technology-enhanced instruction tends to keep students engaged with the mathematics itself while it also strengthens computational capability and that algorithmic reasoning side of thinking.

Recent studies have further demonstrated that technology-supported mathematics instruction encourages collaborative learning, self-directed exploration and also immediate verification of mathematical procedures, which in turn contributes to improved learning outcomes in higher education (Pepin *et al., 2022;* Trgalová and Weigand, 2023). Interactive computational platforms meanwhile allow instructors to present abstract numerical ideas in ways that feel more visually meaningful, so students can link theoretical mathematics with day to day computation.

In the context of this study, Maple 18 was not just a computational software package though it basically also acted as an instructional resource. Through it, students could carry out Chebyshev polynomial approximation procedures, visualize numerical solutions, examine approximation errors, and then compare computational results from different instructional approaches. With technology mixed into Chebyshev polynomial based instruction, students get chances to actively build mathematical knowledge, by experimenting, and by exploring computations rather than simply watching how things are done. This kind of technology enhanced learning setting aligns with current mathematics education directions that stress digital competence, computational literacy, and evidence oriented problem solving.

So, integrating technology into numerical methods instruction therefore supports the growth of higher order mathematical thinking, while also preparing undergraduate students for professional work in scientific computing, engineering analysis, and applied mathematics where computational tools are kind of unavoidable.

Inquiry-Based Learning in Mathematics Education

Inquiry-based learning (IBL) has been getting more and more attention in mathematics education because it sort of makes students active in building mathematical knowledge, not just, you know, sitting there and taking stuff in. Instead of only getting information through lectures, learners are pushed to look into problems, craft conjectures, test mathematical thoughts, analyse evidence and then defend their conclusions. In a way, this learner-centered style helps with deeper conceptual grasp because students are involved in real mathematical reasoning and problem-solving as they go (Artigue and Blomhøj, 2023; Kogan and Laursen, 2024).

In undergraduate mathematics, IBL has also been linked to better performance when students need to connect abstract ideas with more practical uses. When students explore different solution paths, check out alternative approaches, and then think back on what the computations actually show, they tend to strengthen analytical thinking, mathematical communication, and the ability to study on their own. These kinds of moments can help learners understand not just how numerical routines run, but also why they work, and in which situations they do best.

Numerical methods courses are really a natural fit for inquiry-based teaching because so many computational tasks basically force students to dig into approximation tricks, compare numerical outcomes, judge errors, and consider how efficient an algorithm really is. Rather than just following a fixed procedure, students end up doing some kind of mathematical exploration, like weighing the strengths and the limitations of different numerical methods. That whole cycle helps sharpen critical thinking, and it also invites students to choose computational techniques based on evidence, not only on habit or because it usually works.

In this study, the Chebyshev polynomial based instructional style kind of mirrors inquiry based learning, because it gives students chances to craft polynomial approximations, watch how convergence behaves, dig into approximation errors, and then line up the computational findings with what they get from standard numerical techniques. While they do these things, learners are not just observing ,they're actually taking part in the process, gaining a richer grasp of numerical approximation, and sharpening the way they reason mathematically, not only following routine button pressing or procedural computation.

So, when inquiry based learning is folded into undergraduate numerical methods teaching, it naturally helps with building higher order thinking, mathematical ingenuity, and self-directed problem solving. Those qualities are now seen as key outcomes for present day mathematics education, and they're also central if we want graduates ready to handle messy computational tasks in scientific research, engineering, and applied mathematics.

Active Learning in Mathematics Education

Active learning has become one of the most influential instructional approaches in higher education, simply because it gets students to take part in the learning process directly, instead of staying as passive receivers of information. In mathematics education, active learning usually means involving students in things like collaborative problem solving, mathematical investigations, computational experiments, classroom discussions and reflective reasoning. These learner centred activities help promote deeper understanding because students are pushed to build mathematical knowledge through exploration, analysis ,and application, not only through memorizing procedures (Freeman *et al., 2014*; Theobald *et al., 2020).*

Research often shows a clear pattern that active learning improves students' academic achievement, critical thinking, conceptual understanding, and also their long term retention of mathematical ideas. In undergraduate mathematics courses, students who actively engage with real or authentic mathematical problems tend to develop stronger reasoning abilities, more confidence in using algorithms and better skill in transferring mathematical knowledge to brand new, unfamiliar situations (Prince, 2004; Laursen and Rasmussen, 2022). These advantages matter a lot in numerical methods, because students have to grasp, not just the computational steps but also the underlying mathematical ideas about approximation, convergence, stability and numerical accuracy.

The instructional approach taken in this study kinda mirrors active learning, because students have to actually implement Chebyshev polynomial approximations, compare the computed outcomes, analyse the approximation errors, and interpret the numerical outputs using Maple 18. It's not just a matter of watching instructor demonstrations, students are involved in building the numerical solutions, checking how the algorithm performs, and talking through other approximation strategies. These kinds of learning moments tend to push deeper mental engagement, and they sort of make students more accountable for their own learning.

Also, the blended instructional approach this study looks at mixes the usual classroom instruction with computational exploration. So you end up with a more interactive environment, one that can support both conceptual understanding and practical problem solving. This fits with current mathematics education research too, the kind that recommends teaching methods that actively pull students into mathematical reasoning, peer discussion and computational experimentation. Therefore active learning, in a real theoretical sense, helps explain why instruction based on Chebyshev polynomials can improve students' engagement and their learning results in undergraduate numerical methods courses.

Visualization in Mathematics Education

Visualization matters a lot in mathematics education, because it helps students move from those kind of abstract mathematical ideas into something graphical, symbolic, and even computational. In practice, learners can notice mathematical relationships, recognise repeating forms, interpret numerical behaviour, and build a kind of inner intuition about ideas that otherwise feel hard to grasp if they only look at symbols or do manipulations.(Arcavi, 2003; Borba *et al., 2023).*

In numerical methods education this becomes even more obvious. Many computational topics, like polynomial approximation, interpolation, convergence, truncation error, and numerical stability, they are not "visible" in any natural way, so visualization helps a lot. When students see functions sketched, approximation curves compared, error patterns shown, or numerical solutions plotted, they can line up the analytical results with what the computer gives them. That comparison also strengthens how they understand the whole approximation process, and why it behaves the way it does. Tools for this, like Maple, MATLAB, GeoGebra, and Python, offer interactive visualization features so students can adjust parameters and then instantly see what changes, almost immediately.

Recent research seems to suggest that instruction which includes visualization helps students with mathematical reasoning, engagement, and also the ability to solve problems, mostly because it lowers those cognitive roadblocks that show up when ideas stay too abstract (Engelbrecht *et al., 2023;* Pepin *et al., 2022*). It also, in a way, nudges learners to build meaningful links between theoretical mathematics and the kinds of computational use cases people actually face, so students lean more toward real understanding, rather than just procedural recall. And then there's the dynamic side: when the visualization tools are interactive, students can explore more freely, they can try out guesses, check computational

outcomes, and then think back on mathematical relationships using instant graphical feedback, not months later or after a long discussion.

In this particular study, visualization became one of the core parts of the Chebyshev polynomial based teaching approach. Students worked with Maple 18 to produce visual depictions of polynomial approximations, they compared numerical solutions from different instructional conditions, and they investigated how approximation accuracy changes plus how convergence behaves over time. Those visual learning moments allowed students to notice mathematical patterns that are harder to spot if they only do symbolic calculations. In practice this strengthened conceptual understanding as well as computational reasoning, even when the underlying expressions looked intimidating at first.

Consequently, adding visualization into the undergraduate numerical methods teaching sort of gives a practical way to help with mathematical communication computational thinking, and actual problem solving. The mix of visuals, computational technology, and Chebyshev polynomial approximation creates a learning setting that feels pedagogically full, it also matches the present movement in math education and, honestly, helps learners stay more engaged with those thornier numerical ideas.

Even if technology-enhanced instruction has been improving students' interest in numerical methods, not much has been said about whether the advanced approximation techniques themselves can act as teaching instruments for better mathematical understanding.

Theoretical Framework

Constructivism Theory

Constructivism, originally developed by Jean Piaget and then sort of pushed forward by the social constructivist work of Lev Vygotsky, basically says that learners don't just soak up information, instead they actively build up knowledge by linking new experiences with what they already know in their mind, more like existing mental frameworks. So learning happens through exploration, problem solving, reflection, and interacting within learning settings that actually make sense (Piaget, 1972; Vygotsky, 1978).

In mathematics education, this constructivist view often shows up as student-centred teaching. Here students investigate mathematical ideas, they try to make conjectures,

compare different solution paths, and then explain why their reasoning works. These kinds of classroom moves help students gain a deeper conceptual grasp, and they also make it more likely that students can transfer what they learned to new and unfamiliar contexts.

In this particular study, the instructional approach matches those constructivist principles quite closely. Instead of just handing students Chebyshev polynomial approximation as if it were only a method, a fixed computational routine, we let them really explore numerical algorithms. They analysed approximation errors, looked at convergence patterns, and then had to interpret what the computations were showing, using Maple 18. These learning moments were designed so students could construct an understanding from investigation and experimentation, not from memorizing an algorithm and hoping that's enough.

Overall, Constructivism seems like a fitting theoretical base for explaining how instruction based on Chebyshev polynomials can strengthen conceptual understanding, improve computational reasoning, and support problem-solving ability in undergraduate numerical methods courses.

Computational Thinking Framework

Computational Thinking (CT) has turned out to be sort of one of the core competencies learners really need, especially in science, technology, engineering, and mathematics (STEM). Wing (2006) defined computational thinking as a process where you formulate problems, and then design algorithmic solutions that are effectively executed by humans or computers. And, the framework itself kind of covers problem decomposition, abstraction, pattern recognition, algorithm design, evaluation and generalisation.

More recent research in mathematics education also treats computational thinking as something bigger than simple computer programming, it includes mathematical modelling, algorithmic reasoning, numerical analysis and logical decision-making. In other words students build computational thinking when they investigate mathematical structures , evaluate numerical algorithms, analyse approximation errors, and interpret computational results , not just run things blindly.

Chebyshev polynomial-based instruction in particular seems to support computational thinking in a very natural way. Students are actually constructing polynomial approximations, they check convergence behaviour, they compare approximation methods, and they also evaluate numerical accuracy. These kinds of tasks push systematic reasoning,

algorithmic thinking, and mathematical modelling forward, and those are really the key pieces of modern computational mathematics education.

So in the end the Computational Thinking Framework gives a sort of theoretical reason for why bringing Chebyshev polynomial approximation into classroom instruction might strengthen students' higher-order reasoning, and their overall computational competence.

Cognitive Load Theory

Cognitive Load Theory (CLT), proposed by Sweller (1988), sort of explains how instructional design shapes learning by looking at the limits of working memory. In general, the idea is that good instruction cuts down on unnecessary cognitive load, and then it shifts mental resources toward tasks that actually help learners build schemas and get a deeper kind of understanding.

CLT usually talks about three kinds of cognitive load. First there is intrinsic load, which is tied to the built in complexity of the learning material itself. Then there is extraneous load, which comes from ineffective or awkward instructional design, like when students have to wrestle with the presentation rather than the concept. Finally there is germane load, this one adds value because it supports learning and helps knowledge get constructed over time (Sweller *et al., 2019*).

When it comes to numerical methods, they often create a lot of cognitive pressure. Students have to, at the same time, grasp the mathematical theory, carry out computational procedures, read and interpret the numerical results and then check or judge approximation errors. So, instructional approaches should be planned to reduce pointless mental effort, but also keep learners engaged with the key mathematical ideas in a more constructive way.

The blended instructional approach examined in this study is in line with Cognitive Load Theory, because it mixes structured, conventional explanations with some computational exploration via Chebyshev polynomial approximation and Maple 18. In other words, the more traditional instruction gives organised conceptual direction, while the computational part gives students a way to use those ideas through visualization plus small scale experimentation, sometimes in a kind of exploratory manner. Put together, this mix is supposed to lessen extraneous cognitive burden and still help learners build stronger conceptual understanding, along with computational reasoning.

So, in that sense, Cognitive Load Theory gives a pretty handy lens for interpreting why combining conventional instruction with computational activities could, in fact, make undergraduate numerical methods learning experiences better.

Cognitive Load Theory

Cognitive Load Theory (CLT) was developed by Sweller (1988) and it kind of offers a theoretical way to look at how instructional design affects learning, mainly by looking at how working memory humans have limited capacity. Basically, the theory says learning tends to work best when instructional materials are set up in a way that reduces pointless mental strain, while also freeing up cognitive resources for actually building meaningful knowledge. In other words, when you follow Cognitive Load Theory, students usually learn more effectively if learning tasks cut down on unnecessary complexity, and they can instead concentrate on understanding the key ideas and forming structured knowledge patterns, often called schemas (Sweller *et al., 2019*).

Also, Cognitive Load Theory makes a distinction between three types of cognitive load. First, intrinsic cognitive load comes from the natural complexity of the learning content , and also from the level of interaction among its parts. Then, extraneous cognitive load tends to show up because of instructional design issues, like explanations that are poorly arranged or adding extra information that doesn't really help the learner. Lastly, germane cognitive load is the mental effort students put into constructing, reshaping, and even routine-ing knowledge structures that support longer-term understanding. So, good instructional strategies aim to lower extraneous cognitive load, but they also want to boost germane cognitive load, so that learners can end up with a more grounded conceptual grasp.

The study of numerical methods brings cognitive challenges with it, because students have to grasp mathematical theory, build numerical algorithms, do the actual computations, check approximation errors, make sense of what the graphs show, and then judge the correctness and steadiness of the numerical solutions. Trying to do all those things at once can be too much for working memory, especially when students meet advanced approximation techniques for the first time. So, teaching approaches that sort out learning activities in a systematic way, and that also give the right kind of computational help, seem really important for effective learning, overall.

In this study the instructional approaches looked at, match the ideas of Cognitive Load Theory quite closely. Usual, conventional instruction tends to offer structured

explanations, and this cuts down unnecessary cognitive burden. It does that by presenting mathematical concepts in a sensible order. The Chebyshev polynomial based approach then basically adds to that base, because it lets students carry out numerical algorithms in Maple 18, visualise the approximation behaviour, explore how convergence develops, and assess computational accuracy without getting crushed by very long hand calculations. When you mix conceptual explanation with computational exploration, the blended instructional approach is expected to lower extraneous cognitive load, while at the same time boosting germane cognitive load, through purposeful engagement with the numerical ideas.

Also, the way Maple 18 is used helps students, mostly because it takes care of the routine numerical chores, so they can spend more time on the bigger picture of what the math is saying. In practice, that means they can use their mental effort to analyse mathematical relationships, make sense of what the computer returns, and track how approximation behaviour really plays out. Instead of staying stuck on the same kind of calculations over and over, learners can shift toward more "thinking heavy" work like checking numerical accuracy, contrasting approximation strategies, and forming computational reasoning skills. In that sense, this teaching setup fits nicely with the main ideas from Cognitive Load Theory , which says learning tends to improve when extra mental burden is removed and when students are pushed toward deeper, meaningful cognitive activity.

So, Cognitive Load Theory gives a solid theoretical base for this study. It helps explain why mixing standard instruction with computational tasks grounded in Chebyshev polynomials can strengthen students' conceptual grasp, computational thinking, and overall problem-solving ability in an undergraduate numerical methods context. The same theory also acts as a handy lens for reading the educational patterns we saw in this work, especially the fact that the blended instructional group showed a relatively higher mean performance.

Research Questions

The study was guided by the following research questions:

1. What differences exist in students' conceptual understanding of numerical methods when taught using conventional teaching methods, Chebyshev polynomial-based instruction, and a combined instructional approach?
2. How do the three instructional approaches differ in improving students' problem-solving ability and approximation accuracy in numerical methods?

3. What are the students' perceptions and attitudes toward learning numerical methods through conventional teaching, Chebyshev polynomial-based instruction, and the combined instructional approach?

Research Hypotheses

The following null hypotheses were tested at the 0.05 level of significance:

H_{01}: There is no statistically significant difference in students' conceptual understanding of numerical methods among those taught using conventional teaching methods, Chebyshev polynomial-based instruction, and the combined instructional approach.

H_{02}: There is no statistically significant difference in students' problem-solving ability among students exposed to the three instructional approaches.

H_{03}: There is no statistically significant difference in students' perceptions and attitudes toward learning numerical methods among the three instructional groups.

METHODOLOGY

Research Design

The researchers conducted their study of three different teaching methods for teaching numerical integration through a quasi-experimental research design which tested conventional numerical methods together with Chebyshev polynomial based instruction and their combined approach. The design enabled the researcher to assess students' understanding of concepts and their ability to compute and their evaluation of the teaching methods. The spectral system enables scientists from different fields to calculate numerical solutions for all types of equations which include standard equations and partial equations and linear equations and non-linear equations and fractional equations. The scheme requires the Chebyshev basis solutions to be treated as unknown coefficients which will determine the unknown dependent functions of the differential equations. The governing equations receive assumed solutions which create the residual. The Chebyshev polynomial forces the errors to reach their lowest possible values. The process of solving algebraic equations produces unknown coefficient values which are needed for the equation. Then few terms of Chebyshev polynomials of the first kind equation (6) and fourth equation (7) are given:

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \end{aligned} \quad (6)$$

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= 2x + 1 \\ T_2(x) &= 4x^2 + 2x - 1 \\ T_3(x) &= 8x^3 + 4x^2 - 4x - 1 \end{aligned} \quad (7)$$

which are all defined in t ∈ [0, 1](Amr *et al.,* 2017, Authors, 2025)

Participants

The research looked at sixty (60) undergraduate students taking a Numerical Methods course. The people in it were split into three teaching groups, each one had twenty students, meaning: (i) the Conventional Instruction Group, (ii) the Chebyshev Polynomial-Based Instruction Group, and (iii) the Combined Instruction Group.

Because the study used a quasi experimental design, it wasn't really possible to randomly place single students into groups. This happened because participants came from pre-existing classroom clusters, so. even if we wanted, we couldn't shuffle them freely. Still, the three groups were aligned with the different instructional approaches so that each method would run with fairly comparable learning settings. In practice, all groups covered the same course content, over the same time span, taught by the same instructor, and they used the same instructional materials plus the same evaluation process, to reduce instructional tilt or bias.

Right before the instruction changes started, every participant finished a pre-test meant to gauge what they already knew about numerical integration and early numerical methods. That pre-test included objective items along with short-answer questions, and it focused on key ideas, approximation strategies, and the basic ways students carry out computations.

After the instructional intervention ended, the same students sat for a post-test with equivalent questions. Those questions were aimed at conceptual grasp, computational correctness, and ability in solving problems. By comparing pre-test results to the post-test

outcomes, the researchers could judge learning gains for each instructional approach while also limiting the effect of differences students already had at the start, in a kind of clearer way.

Data Collection Instrument

The research team used a structured questionnaire to gather data which measured students' concept comprehension their calculation skills and their attitudes towards learning numerical methods together with their evaluation of teaching effectiveness. The questionnaire used a five-point Likert scale which ranged from strongly disagrees to strongly agree. The research instrument received validation from mathematics education specialists and its reliability assessment showed a Cronbach's Alpha result that exceeded 0.70 which indicated suitable reliability for the instrument.

Table 1 Questionnaire Structure

Section	Content	Number of Items
A	Demographic Information	5
B	Conceptual Understanding	8
C	Problem-solving Ability	8
D	Learning Attitudes	7
E	Teaching Effectiveness	7

In table 1 the questionnaire consisted of five sections. Section A collected demographic information, while Sections B to E measured conceptual understanding, problem-solving ability, learning attitudes, and perceptions of teaching effectiveness. All items were measured using a five-point Likert scale ranging from 1 (Strongly Disagree) to 5 (Strongly Agree). The instrument was reviewed by experts in mathematics education to establish content validity, while a pilot study yielded a Cronbach's alpha coefficient exceeding 0.70, indicating satisfactory internal consistency.

Data Analysis

The collected data were analyzed, using both descriptive and inferential statistics, sort of in parallel. For the descriptive side, frequencies, means, standard deviations, and 95% confidence intervals were used to get a picture of students' responses. In this study, a mean score of 3.00 or above was taken as a positive response, which kinda makes the interpretation a bit more straightforward. Then, to check whether there were significant differences among the three instructional approaches a one-way analysis of variance ANOVA was used, with the 0.05 level treated as the cut off. When it was appropriate, the

magnitude of the treatment effect was assessed via eta squared (η^2), and this was interpreted according to Cohen's (1988) benchmarks, where 0.01, 0.06, and 0.14 correspond to small, moderate, and large effect sizes, respectively.

Besides running tests for statistical significance, effect sizes were computed in order to estimate the magnitude of the differences across instructional approaches. Eta squared (η^2) will be included along with the one-way Analysis of Variance (ANOVA) findings, so that the proportion of variance in students' learning outcomes that can be linked to the teaching methods is clear. Effect sizes, again, were interpreted using Cohen's (1988) guidelines, and η^2 values around 0.01 0.06 and 0.14 indicate small, medium, and large effects, respectively.

To make the results feel a little more precise and easier to read, 95% confidence intervals will also be provided for the estimated mean scores as well as for the mean differences among the instructional groups. These confidence intervals basically give a plausible span for where the true population parameter might sit, and they work alongside hypothesis testing by showing how steady or "tight" the estimated effects are.

Ethical Considerations

The study followed the usual ethical rules for education research, you know, like those widely accepted principles. Approval to carry out the study was secured from the Head of the Department of Mathematics as well as the course lecturer, before any data collection started. Student involvement was fully voluntary, and they were told about the purpose along with the procedures used in the study. Participants were also reassured that their answers would be handled confidentially, and only used for scholarly investigation. In order to keep their privacy safe, no personal identifiers were gathered. Then, all collected data were reviewed in an anonymous way, so nobody could be traced back. If your university has an Institutional Review Board (IRB) or Research Ethics Committee approval number, replace the first statement with the official approval information.

Informed Consent

Before joining the study, all students received straightforward explanations about what the study aimed to do, what participation involved, and that they could withdraw at any point, without any penalty. Written informed consent was acquired from each participant

before the questionnaire was administered and before the teaching-related interventions were introduced.

Numerical Experimental

Linear First-Order Differential Equations procedure

$$\frac{dy}{dx} + P(x)y = Q(x) \quad (7)$$

Integrating Factor (IF) Method

- Identify $P(x)$
- Find integrating factor:

IF= $e^{\int p(x)dx}$ (8)

- Left side becomes derivative of a product
- Integrate and solve for $\frac{d}{dx}(y.IF)$

Illustration 1: Evaluate using conventional method (Rukmangadachari, E. (2012) and Ahmad and Ambrosetti, 2015)

$$\frac{dy}{dx} - y = e^x, \quad y(0) = 1 \quad (9)$$

IF:

$$e^{\int -1dx} = e^{-x} \quad (10)$$

Multiply:

$$\frac{d}{dx}(y)e^{-x} = 1 \quad (11)$$

Integrate RHS:

$$ye^{-x} = x + c \quad (12)$$

Apply condition equation 12 gives;

C=1 (13)

$$y = e^x(x + c) \quad (14)$$

Substitute equation 13 into 14 gives;

$$y = e^x(x + 1) \quad (15)$$

Analysis of the method:

Suppose we have a differential equation

$$L(U(x)) = \eta \quad (16)$$

In the domain η

$$B\mu(u) = \eta \ on \quad \partial\eta \tag{17}$$

Where L(U) denotes a general differential operator in solving derivatives of dependent variables

The following:

1. Chebyshev polynomial for different kinds at different orders are assumed to be the trial function.

$$y = \sum_{i=0}^{n} a_i T_i^* \tag{18}$$

2. Substitute the assumed trial function into the differential equation given to generate the residual.

3. The domain within η has chosen as collocation points.

4. Imposing the boundary condition on the trial function was used.

5. Evaluating the residual at each arbitrary collocation points to give set of algebraic equations.

6. Solve the system of algebraic equation to obtain the constants.

7. Substituting the constant generated into the trial functions, order sixth, tenth and twelfth of all

Illustration 2: Evaluate using Chebyshev polynomial Method (Rukmangadachari, E. (2012) and Ahmad and Ambrosetti, 2015)

$$\frac{dy}{dx} - y = e^x, \quad y(0) = 1 \tag{19}$$

For the First kind

$$y(x) = 1.50000029777x^2 + 0.666663756742x^3 + 0.208351275792x^4$$

$$+0.0499260121409x^5 + 0.00993250613823x^6 + 0.00117013079935x^7$$

$$+0.000799251518548x^8 - 0.000514594223352x^9 + 0.000334349614802x^{10}$$

$$-0.000118114984963x^{11} + 0.0000188020238660x^{12} + 1.99999998244x$$
$$+0.999999999996 \quad (20)$$

For the Fourth Kind

$$y(x) = 1.50000259479x^2 + 0.6666640489540x^3 + 0.208499766560x^4$$
$$+0.0492933882349x^5 + 0.0117870102787x^6 + 0.00261800535347x^7$$
$$+0.00617585178416x^8 - 0.00570790753440x^9 + 0.00035871902714x^{10}$$
$$-0.000130838219178x^{11} + 0.00021179777240x^{12} + 1.99999985139x$$
$$+0.999999999998 \quad (21)$$

RESULTS

Research Question One

What differences exist in students' conceptual understanding of numerical methods when taught using conventional teaching methods, Chebyshev polynomial-based instruction, and a combined instructional approach?

Table 2 shows the descriptive statistics for students' conceptual understanding, across the three ways of instruction. In general, students who got the combined instructional approach seemed to reach the highest average, (M = 3.48, SD = 0.58) , after that came the group taught with the conventional method, with M = 3.38, SD = 0.54. Meanwhile, the Chebyshev based approach, left students with the lowest mean score (M = 3.27, SD = 0.61). Even so, since the 95% confidence intervals overlap , it looks like these apparent gaps in the averages might not actually be statistically meaningful.

Table 2: Descriptive Statistics of Students' Conceptual Understanding

Teaching Method	N	Mean	SD	95% CI
Conventional	20	3.38	0.54	3.13–3.63
Chebyshev-based	20	3.27	0.61	2.98–3.56
Combined	20	3.48	0.58	3.21–3.75

Test of Hypothesis On

Table 3: One-Way ANOVA for Students' Conceptual Understanding

Source	SS	Df	MS	F	P	η^2
Between Groups	0.441	2	0.221	0.662	0.520	0.023
Within Groups	19.002	57	0.333			
Total	19.443	59				

A one-way analysis of variance, ANOVA was run in table 3 to figure out whether students conceptual understanding differed meaningfully across the three instructional approaches. In the end, the results showed that the gap was not statistically significant, $F(2, 57) = 0.662$, $p = .520$, $\eta^2 = .023$. The effect size implies a small practical impact per Cohen (1988) rules. So, the null hypothesis was not rejected. Overall, this outcome hints that even if the combined instructional approach ended up with the highest mean score, its edge over the other methods was still not statistically significant.

Research Question Two

Research Question Two: What differences exist in students' problem-solving ability and approximation accuracy between the teaching methods?

Table 4: Descriptive Statistics of Students' Problem-Solving Ability

Teaching Method	n	Mean	SD	95% CI
Conventional	20	3.33	0.56	3.07–3.59
Chebyshev-based	20	3.36	0.59	3.08–3.64
Combined	20	3.20	0.57	2.93–3.47

Table 4 presents the descriptive statistics for students' problem-solving ability under the three instructional approaches. Students taught using the Chebyshev-based instructional approach obtained the highest mean score ($M = 3.36$, $SD = 0.59$), followed by those taught using the conventional method ($M = 3.33$, $SD = 0.56$), while students exposed to the combined instructional approach recorded the lowest mean score ($M = 3.20$, $SD = 0.57$). The overlap in the 95% confidence intervals suggests that the observed differences among the instructional approaches are relatively small.

Test of Hypothesis Two

Table 5: One-Way ANOVA for Students' Problem-Solving Ability

Source	SS	Df	MS	F	p	η^2
Between Groups	0.289	2	0.145	0.440	0.646	0.015
Within Groups	18.745	57	0.329			
Total	19.034	59				

A one-way Analysis of Variance (ANOVA) was conducted in table 5 to determine whether students' problem-solving ability differed significantly among the three instructional approaches. The analysis revealed that the difference was not statistically significant**,** $F(2, 57) = 0.440$, $p = .646$, $\eta^2 = .015$. The effect size indicates a small practical effect according to Cohen's (1988) guidelines. Therefore, the null hypothesis was not rejected**.** Although students taught using the Chebyshev-based approach obtained the highest mean score, the observed difference was not statistically significant.

Research Question Three

Research Question Three: What are the students' perceptions and attitudes toward learning numerical methods through conventional teaching, Chebyshev polynomial-based instruction, and the combined instructional approach?

Table 6: Descriptive Statistics of Students' Attitudes and Perceptions

Teaching Method	N	Mean	SD	95% CI
Conventional	20	3.29	0.58	3.02–3.56
Chebyshev-based	20	3.39	0.55	3.13–3.65
Combined	20	3.28	0.60	3.00–3.56

Table 6 presents the descriptive statistics of students' attitudes and perceptions toward learning numerical methods. Students exposed to the Chebyshev-based instructional approach recorded the highest mean score ($M = 3.39$, $SD = 0.55$), followed by those in the conventional instructional group ($M = 3.29$, $SD = 0.58$), while the combined instructional approach recorded the lowest mean score ($M = 3.28$, $SD = 0.60$). The substantial overlap in the confidence intervals indicates that the differences among the three instructional approaches are likely to be small.

Test of Hypothesis Three

Table 7: One-Way ANOVA for Students' Attitudes and Perceptions

Source	SS	df	MS	F	p	η^2
Between Groups	0.148	2	0.074	0.222	0.802	0.008
Within Groups	18.979	57	0.333			
Total	19.127	59				

A one-way Analysis of Variance (ANOVA) was conducted in table 7 to examine whether students' attitudes and perceptions differed significantly among the three instructional approaches. The results indicated no statistically **significant** difference**,** $\mathbf{F}(2, 57) = 0.222$, $p = .802$, $\eta^2 = .008$. The effect size was very small**,** indicating that the

instructional approaches accounted for only a negligible proportion of the variance in students' attitudes and perceptions. Consequently, the null hypothesis was **not rejected**.

Teaching Effectiveness and Curriculum Integration

Table 8: Descriptive Statistics of Teaching Effectiveness (Illustrative Using Assumed SDs)

Teaching Method	n	Mean	SD	95% CI
Conventional	20	3.68	0.55	3.42–3.94
Chebyshev-based	20	3.38	0.57	3.11–3.65
Combined	20	3.12	0.60	2.84–3.40

Table 8 presents the descriptive statistics of students' perceptions of teaching effectiveness across the three instructional approaches. Students taught using the conventional instructional method reported the highest mean score for teaching effectiveness (M = 3.68, SD = 0.55), followed by those exposed to the Chebyshev-based instructional approach (M = 3.38, SD = 0.57). The combined instructional approach recorded the lowest mean score (M = 3.12, SD = 0.60). Although students rated the conventional instructional method more favourably, the overlap in the 95% confidence intervals suggests that the observed differences among the instructional approaches may not be statistically meaningful.

Numerical Experimental result

Table 9: shows the result of convectional method and Chebyshev polynomial of the first and fourth kind

SN	Convectional method	First Kind	Fourth Kind
0.0	1.000000	0.999999	0.999999
0.1	1.215688	1.215688	1.215688
0.2	1.465683	1.465683	1.465683
0.3	1.754816	1.754816	1.754816
0.4	2.088554	2.088554	2.088554
0.5	2.473082	2.473082	2.473082
0.6	2.915390	2.915390	2.915390
0.7	3.423379	3.423379	3.423379
0.8	4.005973	4.005973	4.005973
0.9	4.673246	4.673246	4.673246
1.0	5.436564	5.436564	5.436564

Table 9 compares the solutions of a higher-order differential equation obtained using the conventional method and Chebyshev polynomial methods (first and fourth kinds) over x=0.0x = 0.0x=0.0 to x=1.0 x = 1.0 x=1.0.

The results show excellent agreement, with differences only in the late decimal places, confirming that Chebyshev approximations provide highly accurate solutions. The solution also increases steadily from 1.0000 at x=0.0 x = 0.0 x=0.0 to 5.4366 at x=1.0 x= 1.0 x=1.0, indicating the stability and convergence of the numerical schemes.

DISCUSSION

The present study looked at how effective conventional instruction, Chebyshev polynomial based instruction, and a combined instructional approach were for teaching undergraduate numerical methods. Even though the one-way ANOVA results showed no statistically significant differences among the three approaches on conceptual understanding, problem-solving ability, and students' attitudes and perceptions, the descriptive statistics still hinted at a few meaningful educational trends. So, even if the overall learning outcomes stayed pretty similar across the three methods during the study period, it seems like each approach had its own particular strengths, a sort of quiet advantage here and there.

For conceptual understanding, students who were taught with the combined instructional approach achieved the highest mean score, but again, the difference didn't reach statistical significance. In other words, it may not be "proven" in a strict sense, but it does suggest something. Specifically, mixing conventional classroom explanation with Chebyshev polynomial based computational activities could offer complementary learning experiences that, at least in practice, support students' conceptual development. With conventional instruction, the class gets structured explanations and a steady, systematic way of presenting numerical algorithms. Meanwhile, the Chebyshev-based computational work tends to push students toward algorithmic reasoning, mathematical visualization, and more exploratory computation. When these two features are used together, students might be able to connect theoretical ideas with actual computational implementation, which can make conceptual learning feel more grounded. This observation lines up with newer studies that argue blended or integrated instructional approaches support conceptual understanding by bringing together multiple teaching strategies.

For problem solving ability, students taught using the Chebyshev polynomial based instructional approach ended up with the highest mean score among the three groups, even though the difference was also not statistically significant. Still, it kind of backs up earlier research that says computationally enriched learning environments can lead to more higher

order thinking, more algorithmic reasoning, and more independent exploration of numerical methods. When students work directly with Chebyshev polynomial approximations, they can actually dig into how convergence behaves, how accurate the approximation is, what the truncation errors look like, and where numerical stability might get shaky or strong, so both procedural competence and conceptual understanding get reinforced. In this way, the learning experiences seem to fit with current views of computational thinking, where emphasis is placed on active involvement with mathematical algorithms, rather than just routine procedure execution.

Likewise, students who were exposed to the Chebyshev based instructional approach mentioned slightly more positive attitudes and perceptions about learning numerical methods. Even if those differences were small, and again, statistically non-significant, they point toward the idea that computational activities plus interactive mathematical exploration may raise students' interest and engagement. This result lines up with constructivist learning theory, which basically argues that learners build deeper understanding when they actively construct knowledge through investigation, experimentation, and reflection rather than receiving information in a passive way. Also, using Maple 18 to implement the Chebyshev polynomial approximations gave students chances to visualize numerical behaviour, compare approximation methods, and interpret computational outcomes, which in turn encourages active participation in the learning

Not finding statistically significant differences shouldn't be taken as proof that the three instructional approaches are equally effective everywhere in education. Instead, the modest effect sizes shown in this study feel like a sign that the interventions had small but real educational impact, but only in the specific conditions that were tested. A few things could have helped produce these outcomes, for example, the rather small group of sixty students , the quasi-experimental setup, and also how long the instruction lasted. Educational innovations usually need a good amount of sustained use before clear measurable gaps appear, especially when the aim is to build higher-order abilities like computational thinking , deeper conceptual understanding, and mathematical reasoning.

Even so, the numerical experiments that ran together with the educational work indicated something pretty solid: the Chebyshev polynomial method generated solutions that were very close to the benchmark analytical results, and this supports both its computational accuracy and its numerical stability. Put together, strong mathematical performance along

with encouraging educational patterns implies that Chebyshev polynomial instruction could be a promising instructional tool, particularly in undergraduate numerical methods courses. Of course, additional large-scale studies are still needed to get stronger empirical footing, but the current results suggest that merging advanced approximation techniques with more standard teaching practices may help create learning experiences that are richer and that nurture computational thinking, mathematical reasoning, and overall problem-solving competence.

Educational Implications

The findings carry important implications for undergraduate mathematics education. First, mathematics lecturers should try mixing Chebyshev polynomial activities into the existing numerical methods courses, as kind of complementary instructional resources, not simply swapping out conventional teaching completely. Second, curriculum developers ought to fold in spectral approximation techniques into undergraduate numerical analysis curricula, so students gain sharper computational thinking and more solid algorithmic reasoning. Third, teacher education programmes should offer professional development, giving lecturers the pedagogical and technological know-how they need, to implement computational mathematics effectively in practice. Finally, educational policymakers should push for broader integration of mathematical software alongside inquiry-based computational activities in tertiary mathematics instruction, so graduates are better prepared for today's scientific and engineering needs

Limitations of the Study

When thinking about what this study found, there are a few limitations that really should be kept in mind. For one, the sample was only sixty undergraduate students, and they came from just one institution so it might not reflect what happens in other types of schools or learning settings. Also, the approach used here was quasi experimental not fully randomized, so while the results are still useful, it can be hard to rule out every possible confounder that might quietly affect the outcomes. On top of that, the students' perceptions were gathered through self reported questionnaire answers, and those can be influenced by things like response bias or how people feel in the moment. Lastly, the instruction period was rather brief , so it may not show the more enduring or later impacts of Chebyshev polynomial based instruction on what students actually learn over time.

Suggestions for Future Research

For future studies, it would be better to pull larger and more varied samples from several universities, so the findings can actually generalize more broadly. Researchers might also look at the long term impact of Chebyshev polynomial based instruction on how well students retain those numerical methods concepts, not just right after the course ends, but later too. It could also be useful to run comparative studies against other spectral techniques like Legendre and Hermite polynomial methods, because that would add some extra depth to the existing literature in a more systematic way. On top of that, future investigations might use mixed methods, meaning combine more number based analyses with interviews, or even classroom observations, so researchers can understand students learning experiences from multiple angles rather than only one.

Recommendations

1. Incorporate Chebyshev polynomial techniques alongside conventional methods in teaching numerical integration.
2. Include introductory spectral methods in undergraduate curricula.
3. Allocate sufficient time when combining teaching approaches to prevent cognitive overload.
4. Organize faculty development workshops on Chebyshev-based instructional techniques.

Declarations

Consent to Participate

Written informed consent was obtained from all participants before their participation in the study. Participants were informed about the purpose of the study, the procedures involved, their voluntary participation, and their right to withdraw from the study without penalty.

Ethics Declaration

Formal institutional ethics committee approval was not obtained for this study. Permission to conduct the study was obtained from the Head of the Department of Mathematics and the relevant course lecturer at Ladoke Akintola University of Technology (LAUTECH), Ogbomoso. The study was conducted with voluntary participation and informed consent from all participants.

Author Contributions

Adeyemi Ademola Bepo conceptualized the study, developed the methodology, coordinated data collection, conducted the data analysis, and drafted the manuscript.

Ebenezer Adewole Olatunji contributed to the methodology, data collection, interpretation of the findings, and critical review and revision of the manuscript. Both authors approved the final version of the manuscript.

Data Availability

The datasets generated and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Competing Interests

The authors declare that they have no competing interests.

REFERENCES

Adeniran, A. O. (2022). Improving the teaching of numerical methods in higher education through computational tools. *International Journal of Mathematical Education in Science and Technology, 53*(6), 1450–1465.

Ahmad, S., & Ambrosetti, A. (2015). *A textbook on ordinary differential equations* (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-16408-3

Authors. (2025). *Blinded reference.*

Boyd, J. P. (2001). *Chebyshev and Fourier spectral methods* (2nd ed.). Dover Publications. https://store.doverpublications.com/products/9780486411835

Canuto, C., Hussaini, M. Y., Quarteroni, A., & Zang, T. A. (2006). *Spectral methods: Fundamentals in single domains*. Springer. https://doi.org/10.1007/978-3-540-30726-6

Driscoll, T. A., Hale, N., & Trefethen, L. N. (Eds.). (2014). *Chebfun guide.* Pafnuty Publications. https://www.chebfun.org/docs/guide/

Gautschi, W. (2004). *Orthogonal polynomials: Computation and approximation*. Oxford University Press. https://doi.org/10.1093/oso/9780198506720.001.0001

Mahdy, A. M. S., & Mukhtar, N. A. H. (2017). Second kind shifted Chebyshev polynomials for solving the model nonlinear ODEs. *American Journal of Computational Mathematics, 7*(4), 391–401. https://doi.org/10.4236/ajcm.2017.74028

Martín-Caraballo, M., & Tenorio-Villalón, C. (2015). Interactive teaching of nonlinear numerical methods using GeoGebra. *Mathematics Education Research Journal, 27*(3), 321–338.

Owens, R. (2023). Open educational resources and digital platforms for teaching numerical methods. *International Journal of Educational Technology in Higher Education, 20*(1), 1–18.

Pleština, D., Milin Šipuš, I., & Bašić, L. (2021). Polynomials: Unifying concepts in mathematics education. *Educational Studies in Mathematics, 108*(3), 401–420.

Quarteroni, A., Sacco, R., & Saleri, F. (2007). *Numerical mathematics* (2nd ed.). Springer. https://doi.org/10.1007/b98885

Rukmangadachari, E. (2012). *Differential equations*. Pearson India.

Tabieh, S. (2024). A system-based teaching model for learning polynomial functions. *Journal of Mathematics Education, 17*(2), 75–90.

Trefethen, L. N. (2013). Approximation theory and approximation practice. *SIAM Review, 55*(2), 395–397.

Trefethen, L. N. (2019). Spectral methods in numerical analysis. *SIAM Review, 61*(3), 463–489.

Trefethen, L. N., & Weideman, J. A. C. (2014). The exponentially convergent trapezoidal rule. *SIAM Review, 56*(3), 385–458. https://doi.org/10.1137/130932132

Weideman, J. A. C., & Trefethen, L. N. (2007). The Chebfun system for numerical computation. *ACM Transactions on Mathematical Software, 33*(2), 1–18.

Appendix A: Students' Questionnaire

Include approximately 25–30 Likert-scale items, for example:

Section B: Conceptual Understanding

- I understand the concept of numerical integration.
- I can explain the purpose of numerical approximation methods.
- I understand the role of polynomial approximation in numerical analysis.
- I can distinguish between analytical and numerical solutions.
- I understand the convergence of approximation methods.

Section C: Problem-Solving Ability

- I can solve numerical integration problems independently.
- I can apply Chebyshev polynomial approximation correctly.
- I can implement numerical algorithms accurately.
- I can interpret numerical results effectively.
- I can identify computational errors.

Section D: Learning Attitudes

- The instructional method increased my interest in numerical methods.
- The lessons were engaging.
- The instructional activities improved my confidence.
- I enjoyed learning numerical methods.
- I would recommend this teaching approach.

Section E: Teaching Effectiveness

- The teaching approach was well organised.
- The examples were easy to understand.
- The instructional materials supported learning.
- The classroom activities encouraged participation.
- The teaching method improved my computational thinking.