Numeracy
Volume 12, Number 2, 2025 pp. 163-178
P-ISSN: 2355-0074 E-ISSN: 2502-6887
Open Access: https://ejournal.bbg.ac.id/numeracy

INTEGRATING SYMPY INTO COLLABORATIVE LEARNING FOR PARTIAL FRACTIONS AND INTEGRATION: A QUALITATIVE STUDY IN CALCULUS

Panyawat Haarsa*[1]
[1]*Department of Mathematics, Faculty of Science, Srinakharinwirot University, Bangkok, Thailand*

* Corresponding Author: *chaiwichith@g.swu.ac.th*

ARTICLE INFO

Article history:
Received : Aug 30, 2025
Revised : Sep 15, 2025
Accepted : Oct 27, 2025
Available online : Oct 31, 2025

Keywords:
Collaborative Learning, Python, SymPy, Partial Fractions, Integration

ABSTRACT

This qualitative classroom study explores how collaborative learning (CL) supported by Python (SymPy) can strengthen university students' understanding of partial fraction decomposition and its application to integration. Thirty-two second-year mathematics majors participated in four weekly sessions that blended manual problem solving with computational checking. Students worked in six small groups with defined roles (problem solver, coder, recorder, presenter), engaged in real-world tasks (e.g., economic growth, fluid flow, and motion analysis), kept reflective journals, and delivered group presentations evaluated with an analytic rubric. Data sources comprised observations, journals, and presentation assessments. Thematic analysis indicates improved participation, clearer conceptual linking between algebraic manipulation and integral calculus, and more systematic error-checking when Python was used to validate manual work. Groups demonstrating stronger within-group communication tended to employ Python more effectively and reached higher rubric scores. The study discusses practical design choices for CL tasks that combine traditional and computational approaches, and reflects on limitations such as heterogeneous prior programming experience and the absence of pre–post achievement testing. Implications for practice include structuring explicit roles, scaffolding Python early, and aligning tasks with students' backgrounds.

INTRODUCTION

Collaborative learning (CL) has revolutionized education, especially in subjects like mathematics that require theoretical and practical understanding. CL, based on social constructivism, emphasizes student interaction in groups. In these environments, students solve problems, share knowledge, and get to know each other. This educational style encourages peer relationships and knowledge collectives, unlike the traditional, individual-centered approach, according to Callo and Ubayubay (2024). CL provides a systematic framework for simplifying mathematical procedures like partial fraction decomposition and integration. Students can solve problems more creatively and effectively. This strategy improves academic performance and develops critical thinking, communication, and teamwork, which are useful in real life, according to Putri et al. (2024). Partial fraction decomposition simplifies complex rational functions into simpler fractions, facilitating faster integration (Zhu & Luo, 2023). Widely applied in physics, engineering, and economics, it connects mathematics to real-world contexts such as fluid dynamics and financial modelling. These problems make abstract concepts more engaging for students. Building on prior research showing that collaborative learning (CL) enhances problem-solving (Li et al., 2023), this study integrates Python to further improve learning outcomes. Unlike traditional methods that separate manual and computational work, combining CL with Python helps students link theory and practice through symbolic computation and iterative feedback (Zhang, 2019; Hsu et al., 2018).

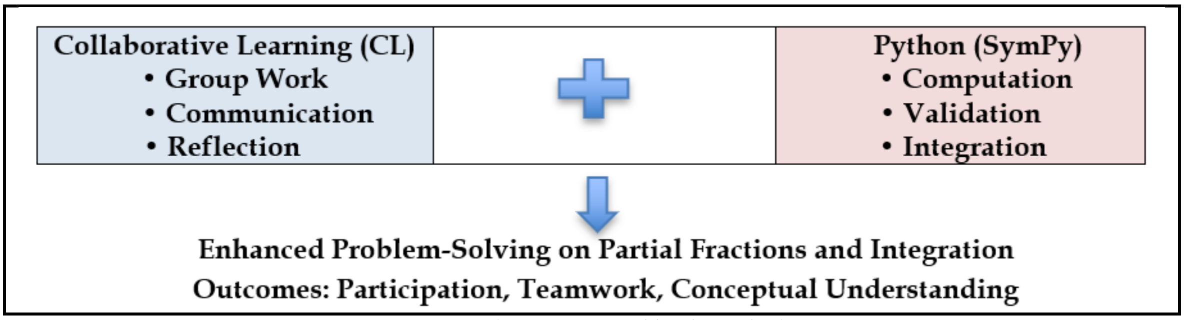
CL helps students develop vital interpersonal skills, which is an essential benefit. confidence Jahara and Daulay (2023) found that collaborative students improve self-confidence, communication, and cooperation. Collaborative problem-solving helps students break down tasks, allocate roles, and overcome hurdles, reflecting real-world interdisciplinary collaboration (He et al., 2023).

When combined with technology, CL exercises allow students to explore Python while improving their technical and analytical skills. Python's partial fraction decomposition and symbolic integration tools allow students to verify manual computations, and enhance their methods (Kamalov et al., 2023). This helps students learn theory and gain computational skills. Due to its dual benefits of improved collaboration and math skills, CL is effective in tackling the complexity of current education.

Reflective techniques enhance CL's impact on learning outcomes. Reflective activities help individuals and groups thrive (Rania et al., 2021). These tasks help students evaluate their performance and discover comprehension gaps. After solving difficulties together, students can review their approaches to learn why some worked and how to improve others. This cyclical method helps students develop lifelong learning and problem-solving skills, increasing self-awareness and adaptability. The combination of reflective techniques and computational technologies allows students to obtain instant feedback on their efforts and guarantees that theoretical knowledge is applied (de Oliveira Santos, 2018). This study extends prior work by applying a collaborative learning (CL) approach to teaching partial fraction decomposition through both manual and computational methods. Integrating Python into group problem-solving, it examines how CL enhances students' mathematical understanding, collaboration, and problem-solving skills. The findings aim to inform mathematics education by demonstrating how modern instructional strategies prepare students for complex interdisciplinary challenges. While prior studies support CL and computational tools, few have explored their combined impact on conceptual understanding and teamwork. For instance, prior studies emphasized programming skills or computational thinking without linking them to mathematical reasoning (Zhang, 2019; Hsu et al., 2018; Li et al., 2023; Boye, 2019). Furthermore, few studies have explored students' reflective engagement and communication processes when using Python as a learning scaffold in calculus contexts. This study fills these gaps by investigating how the integration of Python (SymPy) within a collaborative learning framework can simultaneously foster conceptual understanding, teamwork, and reflective practice in solving partial fraction and integration problems.

RESEARCH METHODS

This study examined how collaborative learning (CL) affected students' partial fraction decomposition and integration comprehension. We accomplished this using Python and manual problem-solving. Figure 1 provides an overview of the intervention, highlighting how CL and Python (SymPy) were integrated to scaffold problem-solving on partial fractions and integration.

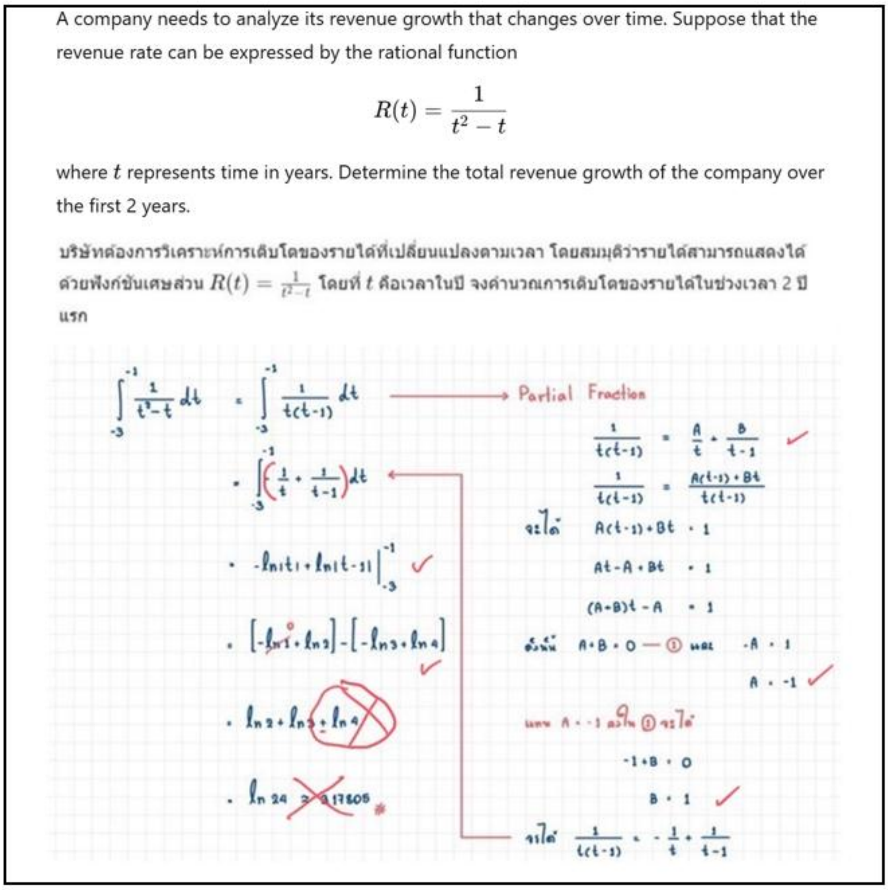


(Source: Created by the author)

Figure 1. Integration of Collaborative Learning (CL) and Python (SymPy) in Partial Fractions and Integration

The methodological approach investigated how computer technologies to enhance group cooperation could improve educational outcomes and interpersonal skills. We used a qualitative approach to study how collaborative activities promote mathematical comprehension and problem-solving. We assessed student involvement, teamwork, and computing skills using observational data, reflective diaries, and group presentations. The study assessed students' abilities to use Python programming and manual computations to solve real-world problems, as well as their ability to collaborate and communicate in multiple groups. Thirty-two Srinakharinwirot University second-year mathematics students in a basic mathematics course participated. We divided students into six groups of five to six, intentionally mixing mathematics and programming proficiency to balance strengths and promote peer learning. Brief diagnostics in partial fractions and basic Python, combined with instructors' knowledge of prior performance, guided an equitable distribution of expertise. Rosters were verified in the first session and adjusted as needed. Group work was evaluated with an analytic rubric weighting communication, content accuracy, and collaboration to ensure consistent scoring.

In the project, each group solved real-world problems to demonstrate the application of partial fraction decomposition and integration. Example scenarios include fluid dynamics, economic growth modelling, and motion analysis. Students worked together to apply mathematical theories to real-world situations in these scenarios. These are a few examples problems that are used in the group projects.



(Source: Created by the author)

Figure 2. Manual Solution Demonstrates Fundamental Mathematical Skills Applied to Solving Real-World Problems

Figure 2. shows the manual process of solving a real-world problem using partial fraction decomposition and integration, demonstrating essential mathematical skills and conceptual understanding for applied fields.

> Assume that the change in the interest rate in an economic system can be expressed by a rational function
>
> $$i(t) = \frac{t+1}{t^2+5t+6}$$
>
> where t represents time in years. Calculate the accumulated investment value during the time interval from $t=1$ to $t=4$ years using this interest rate.

(Source: Created by the author)

Figure 3. Economic Word Problems Assigned for Students to Work on in Groups

> In an experiment of throwing a ball into the air, the trajectory of the ball can be represented by the equation
>
> $$y = \frac{x}{x^2+4x+3}$$
>
> Find the area under the curve representing the motion of the ball from the initial position at $x=0$ to the position at $x=3$.

(Source: Created by the author)

Figure 4. Physics Word Problems Assigned for Students to Work on in Groups

The research project used instructional materials, real-world situations, and technology to aid learning. The instructional materials included step-by-step instructions for partial fraction decomposition, tutorials on Python programming, and practical examples of these principles. We assigned students activities such as calculating fluid flow rates, analyzing economic patterns, and researching parabolic motion to help them apply theoretical knowledge to real-world problems. Python and the SymPy library were essential to this approach. It compared manual calculations with symbolic computing and encouraged students to use both computational and traditional problem-solving strategies. The combination of theory, computing, and cooperation created an engaging and effective teaching experience.

```python
# ============================
# ตรวจคำตอบด้วย SymPy: Partial Fraction + Integration
# ตัวอย่างจากรูป: ∫[t=-3 → -1]  1/(t^2 - t) dt
# ============================

import sympy as sp
sp.init_printing(use_unicode=True)

# สร้างสัญลักษณ์และนิพจน์
t = sp.symbols('t', real=True)
expr = 1/(t**2 - t)   # = 1/(t(t-1))

print("ฟังก์ชันที่จะอินทิกรัล:")
sp.pprint(expr)
print("\n================= ส่วนที่ 1: Partial Fraction =================")

# --- วิธีที่ 1: ใช้ apart (อัตโนมัติ) ---
pf_auto = sp.apart(expr, t)
print("ผลการแยกเศษส่วนย่อยด้วย apart:")
sp.pprint(pf_auto)

# --- วิธีที่ 2: ทำแบบแก้ระบบสมการเพื่อให้เห็นขั้นตอน ---
# ตั้งสมมติฐาน: 1/(t(t-1)) = A/t + B/(t-1)
A, B = sp.symbols('A B')
eq = sp.Eq(1, A*(t-1) + B*t)  # คูณสองข้างด้วย t(t-1) แล้วเทียบสัมประสิทธิ์
print("\nตั้งสมการหลังคูณด้วย t(t-1):")
sp.pprint(eq)

# เทียบสัมประสิทธิ์ของ t และค่าคงที่
eq_t = sp.Eq(A + B, 1)
eq_c = sp.Eq(-A, 1)
```

```python
print("\nได้ระบบสมการ:")
sp.pprint(eq_t)
sp.pprint(eq_c)

sol = sp.solve((eq_t, eq_c), (A, B))
A_val, B_val = sol[A], sol[B]
print(f"\nคำตอบ: A = {A_val}, B = {B_val}")

pf_manual = A_val/t + B_val/(t-1)
print("\nPartial fraction ที่ได้ (แบบทำมือ):")
sp.pprint(pf_manual)

# ตรวจว่าถูกต้อง (ควรได้ 0)
print("\nตรวจความเท่ากันกับนิพจน์เดิม (ค่าศูนย์แปลว่าถูกต้อง):")
sp.pprint(sp.simplify(pf_manual - expr))

print("\n================= ส่วนที่ 2: Integration =================")

# หา antiderivative ทีละเทอม
F = sp.integrate(pf_manual, t)
print("ปริพันธ์ไม่จำกัดเขต (antiderivative):")
sp.pprint(F)

# อินทิกรัลจำกัดเขตจาก t = -3 ถึง t = -1
a, b = -3, -1
I_def = sp.integrate(expr, (t, a, b))
print(f"\nอินทิกรัลจำกัดเขตจาก t = {a} ถึง t = {b}:")
sp.pprint(I_def)

# รวมเป็นลอการิทึมพจน์เดียว และตรวจเท่ากับ ln(3/2)
I_simplified = sp.simplify(I_def)
print("\nรูปแบบลอการิทึมรวมเป็นพจน์เดียว:")
sp.pprint(I_simplified)
```

(Source: Created by the author)

Figure 5. Python (Sympy) Workflow Used by Students to Validate Manual Solutions: *apart()* for Partial-Fraction Decomposition and *Integrate()* for Symbolic Integration.

Procedures

We carried out the study over the course of four weeks, with weekly sessions intended to encourage collaboration and contribute to the development of knowledge and abilities. It was essential to maintain a consistent format during each session in order to guarantee logical learning development.

1. An Introduction to the Session: The lecturer presented an overview of the mathematical topics at the beginning of each session. Some examples of these concepts included partial fraction decomposition and its function in the process of calculating integrals. We utilized live demonstrations to showcase Python's pertinent functionalities, including the SymPy.apart() tool for decomposition and the integrate() method for symbolic integration. These sessions underscored the significance of the connection between mathematical theory and computing practice.
2. Problem-Solving Through Collaboration: After the introductory session, the students collaborated within their groups to resolve the assigned issues. Every member of the group took on a distinct function, such as a coder, a problem solver, a recorder, or a presenter, in order to guarantee that every individual that participated made a contribution that was important to the job at hand. For instance, one group examined fluid flow by rational function integration, while another group modeled economic growth using partial fraction decomposition to study cumulative interest rates. The same group carried out both of these activities. These collaborative exercises encouraged students to share their

thoughts, resolve any misunderstandings, and apply both manual and computational methods to solve problems (Boye, 2019).

3. Presentation and Reflection: At the conclusion of each session, the groups presented their solutions, providing an explanation of their methodology and highlighting the integration of Python-based and manual procedures. Through the use of presentations, students were able to clearly communicate their reasoning, defend their ideas, and receive constructive comments from both their classmates and the instructor. Additionally, groups participated in reflective talks, documented their experiences, difficulties, and lessons gained in reflective journals, and engaged in reflective discussions. These reflections presented a qualitative perspective on how the collaborative process transformed the participants' understanding and abilities.

Data Collection

We used a variety of data collection techniques to provide a comprehensive view of the learning process and the outcomes it produced.

1. Reflective notebooks: Throughout the entirety of the activity, students kept journals in which they documented their experiences, difficulties, and insights. These entries provide qualitative information regarding the dynamics of both individuals and groups, as well as the efficacy of combining manual and computational methods.
2. Behavioral Observations: During the sessions in which the group was working to solve problems, the teacher observed the interactions between the members of the group and made comments on patterns of communication, role allocation, and collaboration. The data collected through observations shed light on the development of group dynamics, revealing how students resolved disagreements, divided duties, and supported each other.
3. Group Presentations: We evaluated each group's presentations based on a variety of factors, such as the clarity and accuracy of their solutions, the integration of manual and computational approaches, and the effectiveness of their teamwork. Through their presentations, students showed how well they understood and applied the principles to difficult problems.

4. Instructor feedback:

   The instructor gave feedback during and after presentations on solution accuracy, approach coherence, and collaboration quality. The objective of this feedback was to act as an assessment tool for both formative and summative use.

Data Analysis and Limitations

We conducted a thematic analysis of observation notes, reflective journals, and presentation evidence using an inductive coding approach, allowing themes to emerge directly from the data rather than being pre-determined by existing theoretical frameworks. A single researcher carried out two independent code–recode rounds two weeks apart (50 units; intra-coder agreement = 90%). Below are five sample analysis units, drawn from a corpus of fifty units, illustrating the data source, unit wording, and preliminary codes that guided our thematic analysis and code–recode procedure

Table 1. Sample Analysis Units, Data Sources, and Preliminary Codes (5 of 50)

#	Analysis Unit	Data Source	Preliminary Codes
1	Team members took turns explaining the manual partial-fractions steps, then a peer used SymPy's `apart()` to check correctness.	Observation	collaboration; computational support
2	A quiet student increased participation after rotating into the coder role.	Observation	participation; role rotation
3	"Seeing the output of `apart()` helped me understand why we multiply by (t−1) to obtain an equivalent equation."	Reflective journal	Conceptual understanding; computational support
4	The group solved by hand first, then used `integrate()` to verify the antiderivative and evaluate the expression.	Presentation	Procedure clarity; computational support
5	We stalled while debugging because the variable t was not declared with real=True.	Reflective journal	challenge; computational support

An audit trail documented codebook revisions and analytic decisions. Triangulation across data sources and reflexive memos were used to enhance trustworthiness. Major subjects in the reflective diary included improving teamwork, integrating computational tools like Python, and understanding math. Behavioral observations explained group dynamics, role distribution, and communication changes throughout the activity (Moreland & Levine, 2014). These observations showed the students had adjusted to the collaborative framework. We evaluated group presentations during problem-solving based on how well they connected with mathematical concepts, presented solutions accurately, and collaborated. These qualitative evaluations showed how CL affected individual and group learning (Johnson & Johnson, 2015). They showed the many benefits of integrating collaborative problem-solving with computational methodologies. Future research should address the flaws in the study, even though it confirmed the benefits of CL. Without quantitative indicators like pre- and post-tests, academic performance changes were difficult to measure (Faniyi, 2023). Students' Python and mathematics knowledge varied which could affect results. Using more evaluation methodologies and intense computational skill training in later research could help examine learning outcomes more thoroughly (Smirnova et al., 2018). Our research showed CL can revolutionize teaching because it promotes active participation, structured cooperation, and reflective practices. This integration of manual and computer methods gave students the skills to solve complicated, interdisciplinary problems and created a model for creative mathematics training (Kynigos & Diamantidis, 2022).

Ethical Considerations

Participants were informed of the study's aims, procedures, and their right to withdraw at any time. Written consent was obtained before participation. All data, including journals and observations, were kept confidential and used only for academic purposes. Identities remained anonymous, and no personal information appeared in reports or publications.

RESULTS AND DISCUSSION

Collaborative learning (CL) enhances conceptual understanding and problem-solving in mathematics, especially in partial fraction decomposition. Group performance was evaluated by accuracy, teamwork, and use of Python tools. Higher-scoring groups showed stronger collaboration and communication, while others struggled due to limited prior knowledge. These results emphasize balanced skills in collaborative settings. Variations in prior knowledge affected consistency, suggesting future research should include targeted support. Given the qualitative design, findings reveal patterns of improved participation, teamwork, and conceptual linkage. Students applied both Python and manual methods to simulate economic growth, fluid flow, and interest rates.

Table 2. Group Evaluation Scores Highlight the Link Between Collaboration Efficiency and Computational Accuracy

Evaluation Criteria	Evaluation Guidelines	G. 1	G.2	G. 3	G. 4	G. 5	G. 6
1. Accuracy of Results	Were the results correct and consistent when validated with Python?	5	5	5	5	5	5
2. Steps in Manual Problem-Solving	Were the steps logical and did they align with the expected outcomes?	5	5	5	5	5	5
3. Python Programming Steps	Were the Python coding steps appropriate and aligned with the problem requirements?	5	4	5	5	5	4
4. Comparison of Results	Was there a clear comparison between manually derived results and Python outputs in terms of accuracy?	5	5	5	5	5	5
5. Team Collaboration	Did all team members participate effectively, and did their teamwork demonstrate efficiency?	5	4	5	5	5	4
6. Reflection and Suggestions	Did the group provide constructive feedback and propose useful improvements for the future?	5	4	5	5	5	5
Total Scores		30	27	30	30	30	28

Note. Scale: 0 = not met; 1–2 = partially met; 3–4 = minor errors; 5 = fully met and accurate

The scoring rubric was as follows: 0 for unmet criteria, 1–2 for partially completed tasks with errors, 3–4 for mostly correct work with minor issues, and 5 for fully accurate completion. Table 2 shows effective cooperative problem-solving, with most groups earning high scores. Groups 1, 3, 4, and 5 achieved perfect scores by integrating Python and manual methods through strong teamwork and logical reasoning. Groups 2 and 6 scored slightly lower due to minor coding and communication issues. These results highlight how collaborative learning (CL) strengthens mathematical understanding and computational skills. Teams that communicated well and used Python effectively performed better (Serçe et al., 2010). More Python training and clearer role distribution could further enhance group performance and future applications.

To promote active participation, students were assigned CL roles such as problem solver, coder, recorder, and presenter. This structure clarified responsibilities and strengthened teamwork. Group discussions helped resolve misunderstandings through debate and strategy sharing (Wayesa, 2020). Presentations and reflection activities assessed learning outcomes and group performance.

Table 3. Evaluation Criteria and Scores

Evaluation Criteria	Evaluation Guidelines	G. 1	G. 2	G. 3	G. 4	G. 5	G. 6
1.Communication Skills	Utilized clear, concise language to deliver information and effectively captured the audience's attention.	5	4	5	5	5	5
2. Content Clarity	Presented accurate, complete information in a well-organized and logical manner.	5	5	5	5	5	4
3. Question Responses	Answered questions with accuracy, logical reasoning, and confidence.	5	4	5	5	5	4
Total Score		15	13	15	15	15	13

Note. Scale: 0 = not met; 1–2 = partially met; 3–4 = minor errors; 5 = fully met and accurate

The scoring was defined as: 5 for excellent, 4 for good, 3 for satisfactory, 2 for needs improvement, and 1 for major development required. Table 3 presents the evaluation results of six groups based on communication, topic clarity, and inquiry responses. Groups 1, 3, 4, and 5 earned perfect scores of 15, demonstrating clear communication and confident, well-organized answers. Groups 2 and 6 scored slightly lower (13) due to unclear delivery, limited confidence, and minor organizational errors. These issues affected their ability to handle audience questions effectively. Overall, the results indicate that strong communication, structured content, and teamwork contribute to high performance (Akilandeswari et al., 2015). Groups 2 and 6 may improve through additional coaching and feedback.

Each group demonstrated their problem-solving process, integrating Python-based and manual calculations. Evaluation covered communication, topic clarity, and responses to questions. Communication was judged by delivery effectiveness, content by accuracy and logic, and question responses by precision, confidence, and attentiveness.

This assessment approach highlighted students' comprehension and problem-solving skills. Along with presentations, they practiced reflection to evaluate their experiences and teamwork. Through self-assessment, students reviewed their contributions, deepened task understanding, and proposed ways to improve group performance. Combined reflection and assessment fostered both individual and group development through active participation.

Integrating manual problem-solving with Python validation enhanced theoretical understanding (Landau et al., 2024). Students used SymPy.apart() and SymPy.integrate() to verify manual work, identify errors, and explore complex cases efficiently. This iterative verification process linked theory with practice and demonstrated the strengths of both traditional and computational methods. The collaborative approach further developed teamwork and communication, though early challenges included unclear roles and uneven participation (Martínez-Venegas, 2022).

Reflective practices were used to address these challenges, encouraging students to evaluate group performance and adjust strategies. This process improved communication, delegation, and use of peer strengths (Biktagirova & Kasimova, 2017). Discussions comparing manual and Python results enhanced critical thinking and problem-solving. Diverse mathematical and programming abilities allowed mutual learning, as students shared expertise in both computation and theory (Muller et al., 2019; Adigun et al., 2021). Reflection fostered appreciation for collaboration (O'Reilly, 2021), and teamwork made complex concepts clearer and more engaging for learners (Rudawska, 2017).

Beyond academic gains, the activity emphasized technology's role in solving mathematical problems. Python aided not only in verifying manual work but also in exploring alternative solutions. It enabled modelling of interest rate changes and visualization in fluid dynamics, deepening conceptual understanding (Gorjão et al., 2022). Despite initial hesitation, peer collaboration and guided tutorials helped students engage with Python effectively (Goswami et al., 2023; Nagar, 2017). This experience improved their confidence, problem-solving accuracy, and appreciation of technology's value in modern mathematics education (Kumar & Mohd, 2024).

Reflection helped students overcome challenges and deepen learning (Joglar & Rojas, 2019). Reflective diaries captured insights, progress, and areas for improvement, revealing greater awareness of role distribution, communication, and mutual support. Regular reflection and dialogue reduced misunderstandings and enhanced group cohesion (Kazeni & McNaught, 2020). Teachers observed improved collaboration through active listening, compromise, and consensus-building (Uğurlu & Şemin, 2020). Students became proactive problem-solvers, increasing group efficiency and confidence when presenting and defending their ideas (Jahara & Daulay, 2023).

Despite its success, the activity revealed areas for improvement. Many students initially struggled to shift from manual computation to Python, causing confusion between both methods. Future sessions should include early Python training to build confidence and consistency. While collaboration enhanced group dynamics, additional guidance on communication and role clarity remains needed. Templates for task delegation and conflict resolution could strengthen teamwork (Perez & Bresciani, 2015). Aligning real-world contexts with students' prior knowledge may further boost motivation. Overall, integrating Python with collaborative learning proved comprehensive and effective. The activity bridged theory and practice, enhancing students' confidence and competence (Kharitonenko, 2022). Combining manual and computational methods improved understanding, while collaboration promoted teamwork and communication (Flynn et al., 2023). Reflection encouraged students to overcome challenges and adopt a growth mindset (Xhaferi & Xhaferi, 2017). Overall, the study highlights the value of innovative math instruction and offers guidance for creating dynamic, future-oriented learning environments.

CONCLUSION

This study shows that integrating collaborative learning (CL) with Python enhanced students' problem-solving, comprehension, and teamwork. Combining computational tools with CL deepened understanding of concepts such as partial fraction decomposition and integration. The approach also strengthened communication and collaboration essential for real-world problem-solving. Applying this model in economics or physics could enrich theoretical learning and support interdisciplinary teaching. Limitations include a small, single-year sample and varied Python proficiency. Future research should involve broader, more diverse groups. Overall, CL and Python offer a framework combining collaboration, reflection, and computation that empowers educators to create inclusive and effective learning experiences.

REFERENCES

Adigun, A., Osofisan, O., Robert, B., & Kolawole, M. (2021). The Development of Adaptive Collaboration in a Dynamic Environment for Information Sharing. *Current Topics on Mathematics and Computer Science Vol. 4*, 134-141. https://doi.org/10.9734/bpi/ctmcs/v4/11931D

Akilandeswari, V., Kumar, A., Freeda, A., & Kumar, S. (2015). Elements of Effective Communication. New Media and Mass Communication, 37, 44–47.

Biktagirova, G. F., & Kasimova, R. S. (2017). Communicative abilites and their formation at primary school children. *Modern Journal of Language Teaching Methods*, *7*(3), 597.

Boye, T. (2019). The collaborative classroom in computing higher education. Proceedings of the 2019 ACM Conference on International Computing Education Research, https://doi.org/10.1145/3291279.3339436

Callo, E. M. E., & Ubayubay, R. M. (2024). Perceptions of peer coaching technique by the alternative learning system community implementer. *International Journal of Research Publications*, *144*(1), 29-49. https://doi.org/10.47119/IJRP1001441320246138

de Oliveira Santos, F. A. (2018). A tool for assisted correction of programming exercises in Java based in computational reflection. *Journal on Computational Thinking (JCThink)*, *2*(1), 51-51. https://doi.org/10.14210/jcthink.v2.n1.p51

Faniyi, A. O. (2023). Enhancing student academic performance through educational testing and measurement. *Edumania-An International Multidisciplinary Journal*, *1*(2), 162-171. https://doi.org/10.59231/edumania/8981

Flynn, D., Hughes, J., & Robb, J. (2023). Improving Mathematics Learning Through Computational Participation. *Journal Of Educational Informatics*, *4*(1). https://doi.org/10.51357/jei.v4i1.214

Gorjão, L. R., Hassan, G., Kurths, J., & Witthaut, D. (2022). MFDFA: Efficient multifractal detrended fluctuation analysis in python. *Computer Physics Communications*, *273*, 108254. https://doi.org/10.1016/j.cpc.2021.108254

Goswami, L., Zeinoddin, P. S., Estier, T., & Cherubini, M. (2023). Supporting Collaboration in Introductory Programming Classes Taught in Hybrid Mode: A Participatory Design Study. Proceedings of the 2023 ACM Designing Interactive Systems Conference, https://doi.org/10.1145/3563657.3596042

He, S., Shi, X., Choi, T.-H., & Zhai, J. (2023). How do students' roles in collaborative learning affect collaborative problem-solving competency? A systematic review of research. *Thinking Skills and Creativity*, *50*, 101423. https://doi.org/10.1016/j.tsc.2023.101423

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296-310. https://doi.org/10.1016/j.compedu.2018.07.004

Jahara, A., & Daulay, N. (2023). Solution-focused group counseling: A solution to enhance students' self-confidence. *KONSELI: Jurnal Bimbingan dan Konseling (E-Journal)*, *10*(1), 59-64. https://doi.org/10.24042/kons.v10i1.16606

Joglar, C., & Rojas, S. (2019). Overcoming obstacles to the formulation and use of questions in the science classroom: Analysis from a teacher reflection workshop. *Research in Science Education*, *49*(4), 1125-1139. https://doi.org/10.1007/s11165-019-9857-5

Johnson, D. W., & Johnson, R. T. (2015). Theoretical approaches to cooperative learning. *Collaborative learning: Developments in research and practice*, 17-46.

Kamalov, F., Santandreu, D., Leung, H. H., Johnson, J., & El Khatib, Z. (2023). Leveraging computer algebra systems in calculus: A case study with SymPy. 2023 IEEE Global Engineering Education Conference (EDUCON), https://doi.org/10.1109/EDUCON54358.2023.10125108

Kazeni, M., & McNaught, C. (2020). Using Group Discussion and Reflection for Developing Student Teacher Autonomy. *International Journal of Work-Integrated Learning*, *21*(1), 1-12.

Kharitonenko, L. (2022). Innovations and traditions in Ukrainian language teaching at the educational establishments of Ukraine: cases, models of the future. *Futurity Education*, *2*(1), 68-85. https://doi.org/10.57125/FED.2022.25.03.7

Kumar, I., & Mohd, N. (2024). Ways of Using Computational Thinking to Improve Students' Ability to Think Critically. In *Infrastructure Possibilities and Human-Centered Approaches With Industry 5.0* (pp. 253-266). IGI Global Scientific Publishing. https://doi.org/10.4018/978-1-6684-9591-2.ch013

Kynigos, C., & Diamantidis, D. (2022). Creativity in engineering mathematical models through programming. *ZDM–Mathematics Education*, *54*(1), 149-162. https://doi.org/10.1007/s11858-021-01314-6

Landau, R. H., Páez, M. J., & Bordeianu, C. C. (2024). *Computational physics: Problem solving with Python*. John Wiley & Sons.

Li, G., Liu, X., & Hou, J. (2023). Research on the teaching model of interdisciplinary computational thinking cultivating from the perspective of problem-solving. International Conference on Computer Science and Educational Informatization, https://doi.org/10.1007/978-981-99-9499-1_36

Martínez-Venegas, L. (2022). Towards a Collaborative Learning Framework for D-learning based on improving the teamwork skills of adolescents. *Avances en Interacción Humano-Computadora*, *7*(1), 33-36.

Moreland, R. L., & Levine, J. M. (2014). Socialization in organizations and work groups. In *Groups at work* (pp. 69-112). Psychology Press.

Muller, M., Fussell, S. R., Gao, G., Hinds, P. J., Oliveira, N., Reinecke, K., Robert Jr, L., Siangliulue, K., Wulf, V., & Yuan, C.-W. (2019). Learning from team and group diversity: Nurturing and benefiting from our heterogeneity. Companion Publication of the 2019 Conference on Computer Supported Cooperative Work and Social Computing, https://doi.org/10.1145/3311957.3359440

Nagar, S. (2017). *Introduction to Python for Engineers and Scientists: Open Source Solutions for Numerical Computation*. Apress. https://doi.org/10.1007/978-1-4842-3204-0

O'Reilly, C. (2021). Reflection on practice: an exploration of a virtual online collaboration as preparation for the year abroad. https://doi.org/10.21827/jve.4.35781

Perez, M., & Bresciani, S. (2015). The role of visual templates on improving teamwork performance. 2015 19th International Conference on Information Visualisation, https://doi.org/10.1109/IV.2015.66

Putri, S. W., Almufidah, A., & Gusmaneli, G. (2024). Strategi pembelajaran berbasis masalah dalam meningkatkan kemampuan problem solving peserta didik. *Jurnal Kajian Penelitian Pendidikan dan Kebudayaan*, *2*(2), 179-187. https://doi.org/10.59031/jkppk.v2i2.401

Rania, N., Coppola, I., & Pinna, L. (2021). Reflective practices to study group dynamics: Implement empowerment and understand the functioning of groups. *Frontiers in Psychology*, *12*, 786754. https://doi.org/10.3389/fpsyg.2021.786754

Rudawska, A. (2017). Students’ team project experiences and their attitudes towards teamwork. *Central European Management Journal*, *25*(1), 78-97. https://doi.org/10.7206/jmba.ce.2450-7814.190

Serçe, F. C., Alpaslan, F. N., Swigger, K., Brazile, R., Dafoulas, G., & Lopez, V. (2010). Strategies and guidelines for building effective distributed learning teams in higher education. 2010 9th International Conference on Information Technology Based Higher Education and Training (ITHET),

Smirnova, Z. V., Vaganova, O. L., Frolova, N. V., Ogorodova, M. V., & Rozarenov, V. V. (2018). Modern Means and Technologies for Estimation of Results of Training. SHS Web of Conferences, https://doi.org/10.1051/shsconf/20185001174

Uğurlu, C. T., & Şemin, F. K. (2020). Peaceful solution of conflicts in schools. In *Empowering multiculturalism and peacebuilding in schools* (pp. 198-224). IGI Global. https://doi.org/10.4018/978-1-7998-2827-3.ch009

Wayesa, N. N. (2020). Research on: Improving Students Group Discussion Skill in Mathematics Class. *Science*, *8*(4), 9499. https://doi.org/10.11648/j.sjedu.20200804.11

Xhaferi, B., & Xhaferi, G. (2017). Enhancing learning through reflection–A case study of SEEU. *Seeu Review*, *12*(1), 53-68. https://doi.org/10.1515/seeur-2017-0004

Zhang, Y. (2019). Fostering Practical Developers in Computer Science Classrooms: A PBL Approach. In *Global Perspectives on Fostering Problem-Based Learning in Chinese Universities* (pp. 55-87). IGI Global.

Zhu, J.-M., & Luo, Q.-M. (2023). Partial-fraction decomposition of a rational function and its application. *Advances in Continuous and Discrete Models*, *2023*(1), 1. https://doi.org/10.1186/s13662-022-03734-4