

KeT-LMS のスマートフォンでの 利用促進に向けた取り組み

長野工業高等専門学校 濱口 直樹

Naoki Hamaguchi, National Institute of Technology, Nagano College

群馬工業高等専門学校 碓氷 久

Hisashi Usui, National Institute of Technology, Gunma College

山口大学・教育学部 北本 卓也

Takuya Kitamoto, Faculty of Education, Yamaguchi University

1 序論

近年、オンライン学習システムの重要性が高まり続けており、特にスマートフォンやタブレットなどのモバイル端末を活用した学習環境への関心が高まっている。これに伴い、学習者が直感的かつ効率的に操作できる教材作成プラットフォームの開発が求められている。モバイル端末の普及によって多様な入力方法やインタラクティブな機能が学習システムに必須となりつつある。

本研究では、オンライン教材作成システム KeT-LMS に対し、以下のような拡張を行った。

- **フリック入力の対応:** スマートフォン環境において広く利用されているフリック入力を可能にする機能を実装し、モバイル端末での利便性を向上させた。
- **問題に応じた図やボタンの動的追加:** 教材作成者が問題ごとに図やボタンを自由にカスタマイズできるようにし、視覚的で理解しやすい教材を提供可能にした。
- **入力に応じた図の動的変化:** 学習者の入力内容に基づき動的に変化する図を生成できる機能を追加し、学習内容の視覚的理解を促進する仕組みを実現した。
- **ログ記録とデータ送信機能:** 学習者の入力データをログとして記録し、それをサーバーに送信する機能を実装した。これにより、学習状況の把握やデータ分析による教材改善が可能となる。

本論文では、まず KeT-LMS の概要とその従来機能について述べる。その後、フリック入力の導入、キーボード入力との両立に向けた設計、問題に応じた図やボタンの追加機能、入力内容に応じた図の動的変更、さらにログ記録とデータ送信機能の詳細を論じる。最後に、これらの拡張による効果と今後の課題について考察する。本研究の目的は、オンライン教材作成プラットフォームの機能を拡充し、学習者にとって直感的かつ効果的な学習環境を提供することである。

2 KeT-LMS について

KeT-LMS は、高遠氏によって開発された学習管理システム (Learning Management System, LMS) であり、数式を扱う教育環境に特化したオンライン教材プラットフォームである。このシステムは、従来の LMS が直面していた数式の入力や送受の複雑さを解消することを目的として設計された。特に、数式の送受を迅速かつ容易に行うことを可能にするツールとして KeTMath が開発され、KeT-LMS の中核をなす機能となっている。

KeTMath の特徴は、数式の入力を効率化するために開発された独自のキーボードである。このキーボードは、従来のフルキーボードと異なり、数学教育に特化した設計となっており、数式の入力に必要な記号や関数を効率的に操作できるレイアウトを採用している。これにより、ユーザーは数式入力 of 煩雑さから解放され、スムーズに演習問題を解くことが可能となる。

さらに、KeT-LMS は、単に演習問題を配付するだけでなく、問題の入力、配付用ファイルの作成、そして数式処理システムである Maxima を用いた自動採点機能までを一貫してサポートしている。Maxima は高機能なオープンソースの数式処理システムであり、学習者が入力した解答を正確に評価することが可能である。この自動採点機能により、教員の負担が大幅に軽減される。これらの機能を統合することで、KeT-LMS は学習者と教員の双方にとって使いやすいシステムとなっている。

しかしながら、現行の KeT-LMS にはいくつかの課題が存在する。その一つは、スマートフォンやタブレットなど画面サイズが限られたデバイスにおける操作性の問題である。KeTMath 独自のキーボードはデスクトップ環境では非常に有効であるが、モバイル環境ではボタンサイズが小さくなるため、押し間違いが生じやすいという指摘がある。これは特に学習者の操作効率に影響を与えるため、改善が求められている。

また、教員からはさらなる機能改善の要望が挙げられている。具体的には、以下の二点が主要な要望として報告されている。第一に、キーボードのボタンサイズを大きくすることで押し間違いを防止し、操作性を向上させること。第二に、学習者が入力した解答に応じて、グラフが動的に描かれるなどのインタラクティブな要素をシステムに導入することで、学習の興味を引き出すことである。これらの要望は、KeT-LMS を学習者にとってさらに直感的で効果的な学習プラットフォームへと進化させるための重要な課題となっている。

本研究では、これらの課題に対応するため、KeT-LMS の拡張を行う。特にフリック入力の導入やキーボードのボタン配置の改良に加え、学習者の入力に応じて動的に変化するインタラクティブな要素の追加を進めることで、システム全体の利便性と機能性を向上させることを目指す。フリック入力とは、特にスマートフォンでのテキスト入力に広く使用されている方式であり、限られた画面スペースで効率的に文字や記号を入力することが可能である。KeT-LMS にフリック入力を実装することで、学習者がモバイルデバイスでの数式入力をより快適に行えるようになることが期待される。この新しい入力方法は、既存のキーボード形式との併用による柔軟性の向上や、操作ミスの削減にもつながると考えられる。

3 フリック入力について

フリック入力は、限られたスペースで効率的に入力操作を行うための方法であり、特にスマートフォンやタブレットなどのモバイル端末で広く利用されている。本研究では、KeT-LMS においてフリック入力を導入し、モバイルデバイスでの操作性を向上させることを目指した。この章では、フリック入力機能の設計および実装について詳述する。

3.1 JavaScript を用いたマウス・タッチ操作の検知

マウス・タッチ操作の検知は JavaScript を用いて行う。具体的には、次のイベントに対して JavaScript のイベントリスナーを用いて対応するロジックを実装している。

- **タッチ開始**：タッチまたはマウスクリックが行われた時点で、対象となるキーを検知する。
- **タッチ移動（ムーブ）**：指またはマウスカーソルの移動を検知し、方向を判断する。
- **タッチ終了**：最終的な移動方向に基づき、選択されたキーを確定する。

フリック入力を実現するためには、マウス操作とタッチ操作の両方に対応する必要がある。これは、デスクトップ環境（主にマウス操作）とモバイル環境（主にタッチ操作）の両方で一貫した操作性を提供するためである。2つの操作に対応するため、それぞれのイベント（mouse イベントと touch イベント）に対応するプログラムの設計が求められた。

3.2 Cinderella を用いたプログラムの実装

フリック入力における具体的なキー操作の判定や、十字キーの表示と選択については、教育用動的幾何ソフトウェアである Cinderella のプログラミング言語 CindyScript を用いて実装している。

```
1  if(clickf==1,  
2      select=1;  
3      nx=ceil((click1x-kx0)/kdx);  
4      ny=ceil(3+(ky0-click1y)/kdy)  
5      butnum=ny*5+nx;  
6  );
```

```
1  if(clickf==2,  
2      select=1;  
3      if(click2x > click1x+2,select=4);  
4      if(click2x < click1x-2,select=2);  
5      if(click2y > click1y+2,select=3);  
6      if(click2y < click1y-2,select=5);  
7  );
```

図1：タッチされたキーの判定 (左図) と十字キーの表示と選択の判定 (右図)

3.4 フリック入力とキーボード入力の両立

学生に対し、キーボード入力方式(図3左)とフリック入力方式(図3右)の両方を提示で、どちらを使うかを問うアンケートを行った。結果はキーボード入力方式を選択したものが4名(44.4%)で、フリック入力方式を選択したものが5名(55.6%)でそれほど差はなく、どちらを使うのかは学生によることが分かった。この結果を踏まえ、システムではキーボード入力方式も残し、どちらも選択できるように設定するようにした。入力方式の切り替えには、ユーザーに表示される画面上部に配置されている切り替えボタンを用いる。

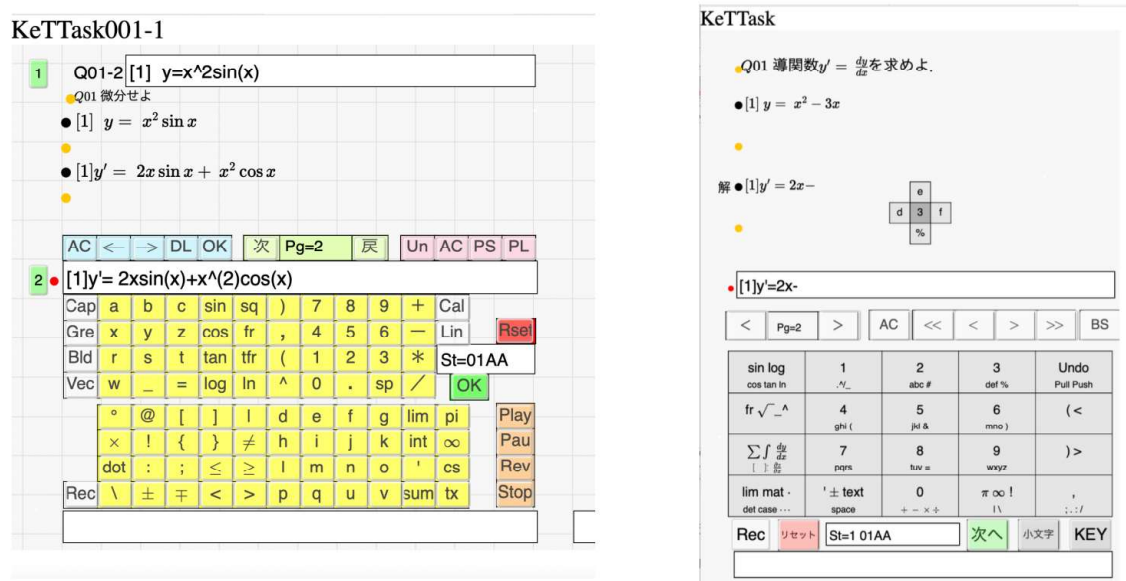


図3: キーボード入力方式(左)とフリック入力方式(右)

4 問題に合わせた図やボタンの追加

KeT-LMS の拡張の一環として、教材作成者が演習問題に応じて図やボタンを自由に追加・編集できる機能を実装した。この機能により、教材作成の柔軟性が大幅に向上し、問題の内容に応じた直感的で視覚的な補助が可能となった。本章では、この機能の設計および実装について詳述する。

4.1 編集モードでの図やボタンの操作

図やボタンは編集モードで自由に配置および操作が可能である。これにより、教材作成者が問題の内容や目的に応じたカスタマイズを行うことができる。以下の操作方法を提供している。

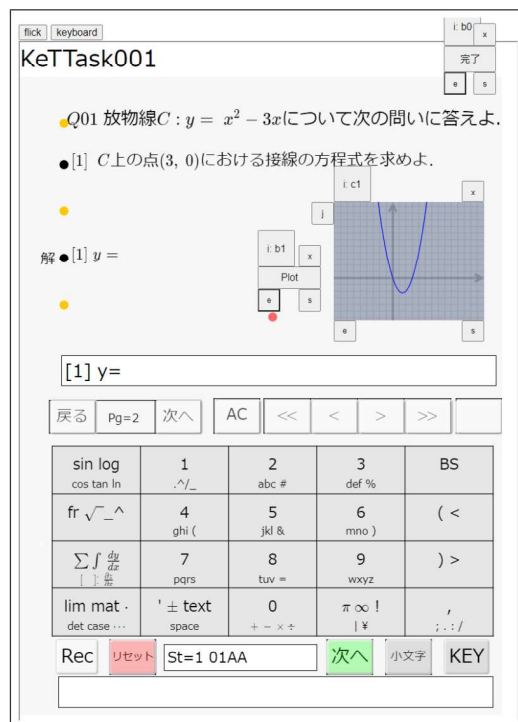


図 4: 自由な位置への配置

- **自由な位置への配置**：図やボタンは、編集モードにおいて任意の位置に配置できる。この機能は、図 4 に示されているように、ユーザーが直感的にレイアウトを調整できるよう設計されている。
- **移動**：各図やボタンの左上にある灰色の正方形をドラッグすることで、画面上で自由に移動が可能である。
- **拡大縮小**：右下に配置された灰色の正方形をドラッグすることで、図やボタンのサイズを調整できる。
- **削除**：右上の灰色の正方形をクリックすると、図やボタンを削除できる

これらの操作により、教材作成者は問題に最適な画面構成を簡単に作り上げることができる。

4.2 二層スクリーン構造による実現

図やボタンの追加は、画面に二層のスクリーン構造を採用することで実現されている。この構造は HTML5 の機能を活用しており、以下のように設計されている。

- **下部スクリーン**：KeT-LMS のシステム本体を表示する。
- **上部スクリーン**：図やボタンを追加・編集するためのレイヤーとして機能する。

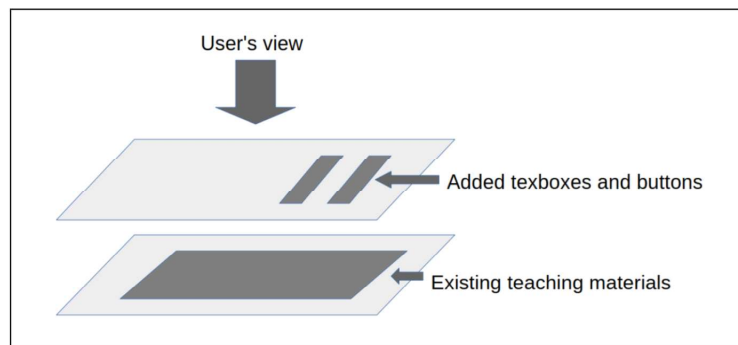


図 5:二層スクリーン構造

この構造を採用することで、KeT-LMS のシステム全体を大幅に変更することなく、視覚要素を自由にカスタマイズできるようになった。具体的には、図 5 に示すように、システム本体の動作を損なうことなく、画面上にボタンや図をオーバーレイ表示できる。この方法は、既存のシステムへの影響を最小限に抑えながら機能拡張を行うための効果的なアプローチである。

4.3 図やボタンの表示制御

教材作成者が問題番号に応じて図やボタンを表示・非表示にする機能も実装されている。この機能は、指定されたテキストボックスに JSON 形式で指定することで柔軟に制御可能であり、例えば

```
{ "#c1": [2,3], "#f1": [1] }
```

と記述した場合、次のように動作する。

- id が c1 の図は、問題番号が 2 または 3 のときにのみ表示される。
- id が f1 の図は、問題番号が 1 のときにのみ表示される。

このような動的な表示制御を行うことで、問題ごとに適した図やボタンを提示できるようになり、学習者にとって分かりやすい教材を作成することが可能となった。

4.4 機能のメリット

この図やボタンの追加機能には以下のような利点がある。

- **柔軟な教材設計:** 教材作成者が問題内容に応じて視覚的要素を柔軟に追加できるため、より効果的な学習をサポートする。
- **既存システムの保守性:** 二層スクリーン構造により、既存の KeT-LMS のシステムに最小限の変更で機能拡張を実現できる。

- **直感的な操作:** 編集モードでのドラッグ&ドロップや拡大縮小操作により、非専門家でも簡単に教材のカスタマイズが可能である。
- **動的な表示切替:** 問題番号に応じた図やボタンの表示制御により、学習者に適切な補助情報を提供できる。

5 入力に応じた Cinderella の図の変更

KeT-LMS の拡張機能の一つとして、学習者が入力した関数に応じてグラフが描画される動的な機能を実装した。この機能は、教育用動的幾何ソフトウェア Cinderella を用いて実現されており、学習者が入力した関数の即時的な視覚化を可能にする。本章では、この機能の仕組みと実装について詳述する。

5.1 「Plot」 ボタンによる動的なグラフ描画

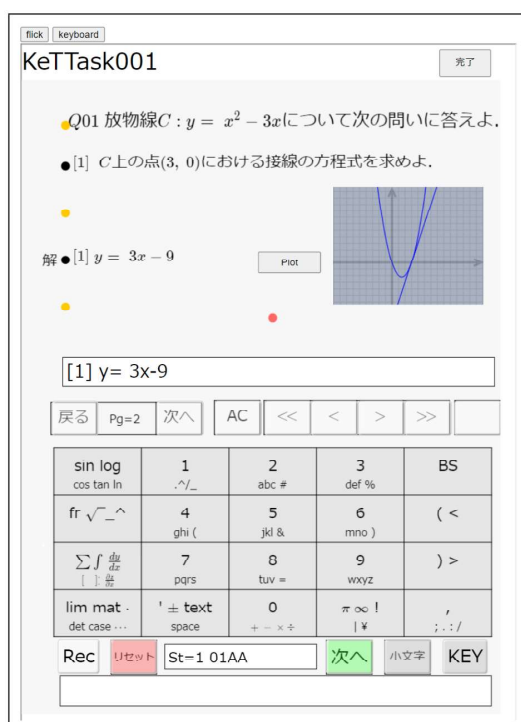


図 6: 「Plot」 ボタンによる動的なグラフ描画

図 6 の画面上に配置された「Plot」 ボタンをクリックすると、Cinderella の図 (画面中央の右側にある図) にユーザーが解答としてテキストボックスに入力した関数のグラフが描画される。このプロセスは主に JavaScript を用いて実装されており、以下のコードがその動作を制御している。

ソースコード 1: テキストボックスに入力した関数のグラフを描画する

```
1 var fx = document.getElementById('test_iframe').contentWindow.cdy.  
  evalcs('Text2').value.text;  
2 var fx1 = fx.split("=")[1];  
3 var fx2 = Palette_Cindy.cinderella_obj[1].evalcs('Tocindyform("' + fx1  
  + '")').value;  
4 Palette_Cindy.cinderella_obj[1].evokedCS("csdraw() := (plot(x*x-3*x);  
  plot("+fx2+"))");
```

以下に、コードの動作を行ごとに解説する。

- **1行目 入力文字列の取得**：test_iframe を id として埋め込まれている KeT-LMS から、変数 Text2 に代入されている文字列を取得し、JavaScript の変数 fx に代入する。この Text2 には、学習者が画面中央にあるテキストボックスに解答として入力した文字列が格納されている。
- **2行目 入力文字列の分割**：取得した文字列 fx を = で分割し、その後ろの文字列（数式部分）を取り出して変数 fx1 に代入する。これにより、ユーザーが入力した関数の右辺（数式）が抽出される。
- **3行目 数式の変換**：fx1 に格納された文字列を Cinderella が理解できる形式に変換する。この処理は Cinderella 上で命令 Tocindyform() を実行することで行われ、変換された数式が変数 fx2 に格納される。たとえば、ユーザーが入力した $2x+1$ は、Cinderella 内で解釈可能な $2*x+1$ に変換される。
- **4行目 グラフの描画**：Cinderella 上で命令 csdraw() を実行する。この命令は、指定された関数 fx2 のグラフを描画する。さらに、このグラフは既定の関数 $y = x^2 - 3x$ のグラフとともに描画され、ユーザーの入力と基準グラフを比較できるようにしている。

5.2 機能の動作イメージ

この機能の動作は、以下のような流れで進行する。

1. 学習者が KeT-LMS 上で関数を入力する。
2. 「Plot」ボタンをクリックすると、入力した関数が Cinderella の形式に変換される。
3. 変換された関数が、基準グラフとともに描画され、学習者に視覚的なフィードバックを提供する。

図6に示すように、描画されたグラフは KeT-LMS の学習画面に統合されており、学習者が即座に確認できるインタラクティブな環境を実現している。

5.3 この機能の意義

この動的なグラフ描画機能には以下のような利点がある。

- **視覚的な理解の促進:** 学習者が入力した関数の形状を即座に視覚化することで、数式とグラフの関係を直感的に理解できる。
- **フィードバックの即時性:** 学習者は自分の入力に対するフィードバックを瞬時に得られるため、試行錯誤を通じて効果的な学習が可能となる。
- **教材設計の柔軟性:** 教材作成者は基準となるグラフや問題に応じて動的に描画されるグラフを設定でき、さまざまな教育目的に対応できる。

6 ログの記録とデータ送信

KeT-LMS の拡張機能として、生徒の入力方法に関するログデータを記録し、それを解答データとともにサーバーへ送信する機能を実装した。この機能により、学習者の操作データを収集することが可能となり、教材の改善や学習者の行動分析に役立てることができる。本章では、この機能の動作と実装について説明する。

```
1 const intervalId = setInterval(() =>{  
2     takelog();  
3     if(timecount > timeout){  
4         clearInterval(intervalId);  
5     }},  
6 1000);
```

図 7: ログの記録を制御する JavaScript のプログラム

6.1 入力方法ログの記録

生徒が問題を回答している間、現在使用中の入力方法（例: キーボード入力、フリック入力）を 1 秒ごとに記録する仕組みを構築した。これらのデータは、非表示のテキストエリアにリアルタイムで記録される（非表示のテキストエリアを使用することで、ユーザーの操作を妨げることなくデータを蓄積できる）。ログの記録を制御する JavaScript のプログラムが図 7 に示されている。このプログラムの内容について解説する。

- **takelog() の呼び出し:** 1 秒ごとに、2 行目の関数 takelog() を呼び出し、現在の入力方法をログとして保存する。
- **takelog() の内容:** 関数 takelog() では取得したログを非表示のテキストエリアに追加し、後のデータ送信に備える。

```

1 function check submit data() {
2   let con = confirm("問題の解答とキー入力のログをサーバーに送り、課題を終
   了しますが、それで良いですか?");
3   if (!con) {
4     return(0);
5   }
6   document.getElementById('test_iframe').contentWindow.cdy.evokedCS('rec=
   rehead+Getcurtime()+";"; ansL_nqu=Textedit(ch,""); str="";
   forall(1..(length(ansL)),n,str=str+ansL_n+";";); rec=rec+
   substring(str,0,length(str)-2); Subsedid(9,rec); StrL_4=rec;');
7   document.getElementById('test_iframe').contentWindow.cdy.evokedCS('
   jsrec = replace(rec,"\\", "[dash]");');
8   let ansdata = document.getElementById('test_iframe').contentWindow.cdy
   .evalcs('jsrec').value.replaceAll("[dash]","") + "\n";
9   let logname = document.querySelector("#logname_data_1").value;
10  let fname = "Ket-LMS/" + logname + ".log";
11  let uname = ansdata.split(";")[0];
12  let funame = "Ket-LMS/" + logname + "_" + uname + ".klog";
13  let ans = file_check(funame);
14  if (ans == "TRUE") {
15    alert("課題は既に提出されています");
16  } else {
17    submit_data(ansdata,logname,fname,funame);
18  }
19 }

```

図 8: 解答データとログデータをサーバーに送信するプログラム

6.2 ログデータの送信

生徒が問題を回答した後、画面右上に配置された「完了」ボタン（図 6 参照）をクリックすることで、解答データとログデータがサーバーに送信される。このプロセスは図 8 に示されている JavaScript のプログラムによって実行される。プログラムでは、課題の終了をユーザーに確認した後、ユーザーが入力した解答とともに非表示のテキストエリアに記録されている入力ログをサーバーに送信する。

6.3 ログ機能の意義

このログ記録および送信機能は、学習者の操作に関する詳細なデータを収集するための重要な役割を果たす。このデータは、「どの入力方法が学習者にとってわかりやすく、どの入力方法が改善を必要としているかの評価」に活用できる。

7 まとめと今後の課題

本研究では、オンライン教材作成システム KeT-LMS の操作性と機能性を向上させるために、複数の機能拡張を行った。

まず、スマートフォンやタブレットなどモバイル端末での使用を念頭に、フリック入力を導入した。これにより、限られた画面スペースでも効率的な入力が可能となり、モバイル環境での操作性が大きく向上した。また、教材作成者が演習問題に応じて図やボタンを自由に配置できる機能を追加し、問題内容に合わせた柔軟な教材設計が可能となった。この実装には HTML5 の二層スクリーン構造を活用しており、既存のシステムに大幅な変更を加えることなく新機能を実現した。

さらに、学習者が入力した関数を動的にグラフ化する機能を追加し、数式とそのグラフの関係を視覚的に理解するための環境を提供した。この機能により、入力結果が即座に視覚化されるため、学習者は試行錯誤を通じてより深い理解を得ることができる。加えて、学習者の入力行動を記録し、それを解答データとともにサーバーへ送信する機能を実装した。この機能は、学習者の行動データを収集・分析し、教材の改善や学習プロセスの可視化に役立つ。

これらの機能拡張により、KeT-LMS は学習者に対して効率的でインタラクティブな学習環境を提供すると同時に、教員にとっても使いやすい教材作成ツールとなったのではないかと考える。

本研究では KeT-LMS の利便性と機能性を大幅に向上させたが、いくつかの課題が残されている。フリック入力については、さらなる精度の向上や、デバイス特有の操作性への最適化が必要である。また、学習者の興味をさらに引きつけるため、入力内容に応じて動的に変化するグラフや図にアニメーション効果や追加情報を付加することも検討すべきである。

さらに、AI を活用した学習支援機能の導入が重要な課題として挙げられる。具体的には、学習者の行動データや解答履歴を分析し、個別に最適化されたフィードバックや追加問題を自動生成する仕組みを構築することが考えられる。このような機能は、学習者の理解度向上や学習効率の改善に大きく寄与すると期待される。

また、多様な学習者に対応するため、インターフェースの多言語対応や視覚的にわかりやすいデザインの改良も必要である。最後に、収集されたログデータをより効果的に活用し、学習プロセスの分析や教育研究に貢献できるような仕組みを整備することが求められる。以上の課題に取り組むことで、教育現場における多様なニーズに応えるプラットフォームとしてその価値を高めて行きたい。

謝辞

本研究は JSPS 科研費 22K02972 の助成を受けている。

参考文献

- [1] 高遠節夫, 濱口直樹, 北本卓也, 1 次元表現ルールに基づいた数式の送受と授業実践, 城西大学数学科数学教育紀要 4 (2023), 23–34.
- [2] 高遠節夫, 碓氷久, 西浦孝治, 濱口直樹, KeT-LMS の開発と授業実践, 城西大学数学科数学教育紀要 5 (2024), 38–49.
- [3] KeTCindy Home, <https://s-takato.github.io/ketcindyorg/indexj.html>
- [4] T. Kitamoto, M. Kaneko, S. Takato : "E-learning system with Computer Algebra based on JavaScript programming language", Proc. of ATCM 2018, Yogyakarta, 2018, 123-133.
- [5] Cinderella Official Home Page : URL <https://www.cinderella.de/tiki-index.php> (2024 年 11 月 15 日閲覧)
- [6] PyScript Official Home Page : URL <https://pyscript.net/> (2024 年 11 月 15 日閲覧)
- [7] Apache Echarts Official Home Page:URL <https://echarts.apache.org/en/index.html> (2024 年 11 月 15 日閲覧)

[8] JspradSheet Official Home Page : URL <https://jspradsheet.com/> (2024 年 11 月 15 日閲覧)