

# 数式自動採点システム STACK を用いた数学学習支援 iOS アプリケーションの開発

## Development of mathematics learning support iOS application using STACK

高塚 朝陽<sup>\*1</sup>, 中村 泰之<sup>\*2</sup>

Asahi Takatsuka<sup>\*1</sup>, Yasuyuki Nakamura<sup>\*2</sup>

<sup>\*1</sup>名古屋大学情報学部

<sup>\*1</sup>School of Informatics, Nagoya University

<sup>\*2</sup>名古屋大学教養教育院

<sup>\*2</sup>Institute of Liberal Arts and Sciences, Nagoya University

Email: [takatsuka.asahi.z6@s.mail.nagoya-u.ac.jp](mailto:takatsuka.asahi.z6@s.mail.nagoya-u.ac.jp)

**あらまし**：本研究では、数式自動採点システム STACK を用いた数学 e ラーニングを、スマートフォン上で快適に利用するための iOS アプリケーションを開発した。従来の Web ブラウザ経由の利用では、PC を前提とした画面レイアウトや複雑な数式入力操作がモバイル学習の障壁となっていた。そこで本研究では、Moodle Mobile Web Service を利用した通信基盤を確立し、さらに、フリック入力方式を採用した独自の数式入力特化型キーボード「iFlickMath」を実装した。これにより、サーバー側への追加プラグイン導入を不要とする高い汎用性を確保しつつ、モバイル環境における数式入力式問題の演習効率を大幅に向上させることができた。

**キーワード**：数学教育, STACK, モバイルラーニング, iOS, 数式入力

### 1. はじめに

近年、GIGA スクール構想による端末整備や、新型コロナウイルス流行に伴うオンライン学習の普及により、場所を選ばない「モバイル・ラーニング」の需要が急速に高まっている。しかし、理数系教育において重要となる「数式入力形式の演習」をモバイル環境で実現することには課題が多い。既存の学習アプリの多くは選択式であり、数式の構築プロセスや論理的思考力を評価することが困難である。

この課題に対し、数式処理システムをバックエンドに持つ STACK (System for Teaching and Assessment using a Computer algebra Kernel) が注目されている<sup>(1)(2)</sup>。STACK は解答の代数的等価性を判定でき、詳細なフィードバックが可能な、数式自動採点システムである。例えば  $x + 1$  が正解の問題に対し、 $1 + x$  と入力しても正解と判定できる。しかし、従来の STACK は PC ブラウザでの利用を前提としており、スマートフォンでは「数式入力の煩雑さ」や「画面レイアウトの乱れ」が発生する場合があります。Moodle 公式アプリも標準では STACK に対応しておらず、サーバー側の改修なしには利用できない。

そこで本研究では、STACK の強力な評価機能を維持しつつ、モバイル端末に最適化されたインターフェースを持つ iOS アプリケーションを開発することを目的とした。

### 2. 手法・システム構成

本システムは、クライアントサイド (iOS アプリ) とサーバーサイド (今回のアプリでは大学の Moodle サーバー) から構成される。サーバー側には既存の

標準的な Moodle および STACK プラグインが稼働している環境を想定し、アプリ利用のためにサーバー側へ特別なプラグイン (Local plugin 等) を追加インストールする必要がない設計とした。これは、教育機関が既存の LMS 環境を変更することなく導入できる高い汎用性を担保するためである。

両者の通信には、Moodle 標準の外部サービス Moodle Mobile Web Service API<sup>(3)</sup> を利用する。認証、問題データの取得、解答の送信、採点結果の受信といった一連の処理を REST API 経由で行うことで、LMS を BaaS (Backend as a Service) のように活用する (図 1 参照)。

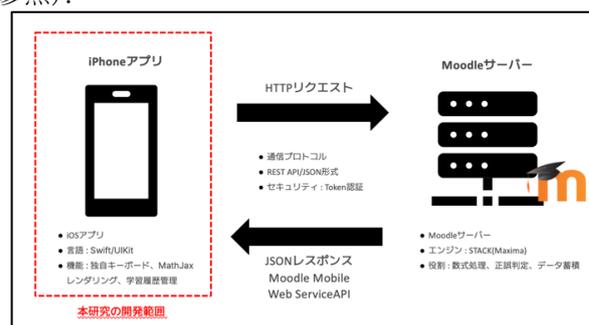


図 1 システム構成図

### 3. 実装技術

開発言語には Swift、UI フレームワークには UIKit を採用した。前述したように従来の Web ブラウザ経由での Moodle 利用は、モバイル端末では学習効率が著しく低下する課題があった。本アプリでは、OS ネイティブの描画性能と独自のインターフェースを実装することでこれらの技術的制約を克服し、ブラウザ版では実現不可能な快適な演習環境を提供する。

具体的には、学習者がストレスなく演習を行えるよう、特に「数式の表示」と「数式の入力」の2点において重点的な最適化を行った。

### 3.1 Moodle API との連携

ユーザー認証には `tool_mobile_get_public_config` 等を用い、Web サービス利用トークンを取得してセッションを管理する。問題の取得には `mod_quiz_get_attempt_review` を利用するが、レスポンスに含まれる HTML データには数式レンダリングライブラリが含まれていない。そのため、クライアント側で HTML を解析し、後述する表示処理と結合するロジックを実装した。解答送信においては、STACK 特有のデータ構造(入力欄識別子やシーケンスチェック用 ID) をアプリ内で生成し、`mod_quiz_process_attempt` を通じて送信することで、ブラウザ利用時と同等の挙動を再現している(図1参照)。

### 3.2 iFlickMath (数式入力キーボード)

モバイル利用の最大のボトルネックである入力操作を改善するため、数式入力特化型カスタムキーボード iFlickMath を開発した。日本語のフリック入力を援用し、JavaScript で開発された FlickMath<sup>(4)</sup> の概念を参考にしつつ、iOS の UIResponder を活用したネイティブ実装を行うことで、遅延のない操作性を実現した。本キーボードは、数学頻出記号( $\int$ ,  $\sum$ ,  $\sin$  等) をタブごとに集約し、フリック操作とタップ操作を組み合わせることで、標準キーボードと比較して大幅に入力手数を削減した。また、入力された数式はリアルタイムで MathJax によりレンダリングされ、プレビュー表示されるため、学習者は入力ミスを即座に確認できる(図2,3参照)。

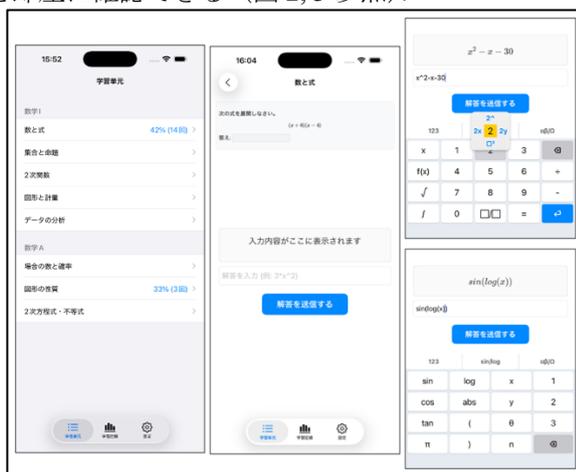


図2 左から単元選択画面, 演習画面, iFlickMath

### 3.3 数式表示と学習履歴の可視化

取得した問題文や解説に含まれる LaTeX 形式の数式コマンドを、アプリ内の WebView 上で正しく表示するため、MathJax ライブラリを動的にインジェクションする仕組みを実装した。同時に、PC 向けの不要なナビゲーション要素を DOM 操作により削除し、

スマホ画面に最適化したレイアウトを提供している。また、API から取得した過去の解答データに基づき、単元ごとの演習回数と正答率をグラフ化して表示する機能を実装し、学習者のメタ認知を支援する(図3参照)。特にグラフから一目で進捗がわかるのはアプリならではの機能である。



図3 左から解説表示, 演習回数, 正答率

## 4. 結論と今後の展望

本研究では、Moodle API を活用し、サーバー側の変更を不要とする STACK 対応 iOS アプリケーションを構築した。独自開発の iFlickMath により、モバイル環境における記述式数式入力の負荷を軽減し、隙間時間を活用した演習を可能にした。

今後は、本システムの高い汎用性を活かし、高等学校や学習塾といった教育機関単位での広範な導入を促進する。組織的な導入により大規模かつ均質な学習データ(ログ)が蓄積されれば、教師・開発者双方に以下の新たな価値をもたらすと考える。

第一に、教師側のメリット(学習管理の効率化)である。多数の生徒の学習状況を一元管理し、クラス全体で正答率が低い「躓きポイント」を容易に可視化できるため、経験則のみに頼らない、客観的データに基づいた授業内容の改善が可能となる。

第二に、開発者側のメリット(適応学習の精度向上)である。単なる正答率の算出に加え、問題間の相関関係や、誤答パターンに基づく条件分岐といったロジックの構築が可能となる。これらの分析に基づき、最終的には個々の学習特性に最適化された問題を推奨・出題する「アダプティブ・ラーニング」機能の実装を目指す。

### 参考文献

- (1) Sangwin, C. J.: Computer Aided Assessment of Mathematics, Oxford University Press (2013)
- (2) 中村 泰之: “数学 e ラーニング”, 東京電機大学出版局, (2010)
- (3) Moodle HQ: Moodle Mobile Web Service API Reference, <https://moodledev.io/general/app/development/plugins-development-guide/api-reference> (2026年1月30日閲覧)
- (4) Nakamura, Y. and Nakahara, T.: “A New Mathematics Input Interface with Flick Operation for Mobile Devices”, MSOR Connections, Vol. 15, No. 2, pp. 76-82, (2017)