

Excel で数学的思考をデザインする

－ UI/UX から見た算数・数学教育の新しい形 －

VBA エンジニア

金沢愛美

概要

本論文は、表計算ソフト Excel を、単なる計算ツールではなく数学的思考をデザインし、学習者に体験させるための学習環境として再定義することを目的とする。従来の紙媒体中心の算数・数学教育では、手続き的知識の習得に偏り、学習者が数学の本質である「構造」「関係」「一般性」を動的に理解する機会が限定されてきた。これに対し、Excel の UI/UX（セル構造、数式バー、オートフィル、グラフ、条件付き書式等）は、操作に対する即時フィードバックを通して、構造化・抽象化・一般化・関数的思考・最適化といった数学的思考の主要要素を自然に引き出す特性を持つ。

本研究では、これらの UI/UX 特性を理論的に整理するとともに、筆者が開発した九九教材、一次関数教材、データ分析教材をケーススタディ¹として分析し、Excel が数学的思考の形成にどのように寄与するかを検討した。その結果、Excel は「操作→変化→発見」の学習サイクルを生み出し、学習者の能動的・探究的理解を促進すること、さらに UI/UX に基づく教材デザイン原則（余白の活用、色の意味付け、操作フローの単線化、誤操作防止、動的変化の必然性）が、認知負荷を最適化し、数学的構造の理解を支えることが明らかになった。

加えて、Excel VBA による教材開発は、学習者を「使う側」から「作る側」へと移行させ、数学的構造を自ら設計・実装するという高度な構成的学習を可能にする。以上の知見を踏まえ、本論文は UI/UX を基盤とした数学教育の新しいモデルを提示し、Excel が現代的な数学学習環境として大きな可能性を持つことを示した。

キーワード：「UI/UX」、「Excel VBA」、「動的理解」

¹ 8. 参考[1] 参照

1. 序論

算数・数学教育において「数学的思考の育成」が重要視されて久しい。しかし、現行の教育実践においては、依然として手続き的知識の習得が中心となり、学習者が数学の本質である「構造」「関係」「一般性」を理解する機会は限定的である。特に、紙媒体を中心とした教材環境では、学習者が自ら操作し、その結果の変化を観察し、そこから規則性を発見するという動的・構成的な学習過程を十分に支えることが難しい。

一方、近年の教育研究では、学習者が自ら操作し、フィードバックを得ながら理解を深める「構成主義的学習」や「探究的学習」の重要性が指摘されている。しかし、これらの学習観を実現するための具体的な教材設計論は十分に確立されているとはいえない。

本論文では、表計算ソフト Excel を単なる計算ツールとしてではなく、数学的思考をデザインし、学習者に体験させるための学習環境として再定義する。

特に、Excel が持つ UI/UX（ユーザーインターフェース／ユーザー体験）の特性が数学的思考の諸側面（構造化・抽象化・一般化・関数的思考・最適化）をどのように支援し得るのかを理論的に整理し、筆者が開発した教材群をケーススタディとして分析する。

本研究の目的は以下の3点である。

（ア）Excel の UI/UX が数学的思考に与える影響を理論的に整理すること

（イ）数学的思考を支援する UI/UX のデザイン原則を抽出すること

（ウ）教材実践を通して、UI/UX に基づく数学教育モデルを提示すること

本論文は、数学教育学・学習科学・HCI（Human-Computer Interaction）の3つの領域を横断しながら、算数・数学教育の新しい可能性を提示するものである。

2. Excel は「数学の実験室」である

2.1. Excel の UI/UX が学習に与える構造的影響

Excel の画面構造は、数学的思考を促す複数の特徴を備えている。

2.1.1. セル構造と構造化

Excel のセルは、情報を空間的に構造化するための最小単位である。セルの配置・境界線・色分けは、数学的対象の構造を視覚的に表現し、学習者が情報を整理し、関係を把握することを支援する。

2.1.2. 数式バーと抽象化

数式バーは、具体的な値の背後にある抽象的な関係を表現する領域である。「値を入力する場所」と「関係を書く場所」が分離されていることは、数学における「具体」と「抽象」の区別を自然に体験させる。

2.1.3. オートフィルと一般化

オートフィルは、数学の一般化を体験するための強力な UI である。1つの規則を作り、それを全体へ拡張する操作は、数学的な一般化のプロセスと一致する。

2.1.4. グラフ機能と関数的思考

Excel のグラフは、入力と出力の関係を視覚的に表現する。学習者は、値を変更することでグラ

フが動的に変化する様子を観察し、関数の本質を体験的に理解する。

2.1.5. 条件付き書式とアルゴリズム的思考

条件付き書式は、条件に応じてセルの色や形式を変化させる。これは「条件→判定→表現」というアルゴリズム的思考を視覚的に体験させる UI である。

2.2. 紙媒体では不可能な「動的理解」

Excel の最大の特徴は、操作に対して即時にフィードバックが返ってくる点にある。学習者が値を変更すると、表・グラフ・色などが瞬時に変化し、その変化を通して関係性を理解することができる。この「操作→変化→発見」の循環は、数学の本質である「関係の理解」を支える重要な学習過程である。紙媒体では、値を変えても結果は自動で変わらない、複数の条件を比較するには手作業が必要、誤答のパターンを可視化することが難しいといった制約がある。Excel はこれらの制約を取り払い、学習者が自ら操作しながら理解を深める構成的・探究的な学習環境を提供する。

3. UI/UX から見た数学的思考の 5 原則

ここでは、Excel の UI/UX を通して立ち上がる「数学的思考」を 5 つの原則として整理し、それぞれをもう一段深く掘り下げて論じる。

3.1. 構造化 (Structure) — 「どのように並べるか」が思考を決める

構造化とは、対象を「まとまり」と「関係」に分けて整理する営みである。数学では、数列・表・図形・グラフなど、あらゆるものが「構造」として捉えられる。Excel の UI は、この構造化を次のように支援する。

- セルの格子構造：行と列の交差によって、情報が「位置」を持つ。例えば九九表では、行が「かけられる数」、列が「かける数」となり、セルはその積を表す「構造の交点」として機能する。
- 領域の分割とグルーピング：見出し行・見出し列・枠線・塗りつぶしによって、「ここからここまでが一つの意味のかたまり」という認識が生まれる。これは、集合・部分集合・ベクトルの成分など、数学における「まとまり」の感覚と対応する。
- 視覚的階層構造：太字・色・罫線の違いによって、「上位概念」と「下位の具体」が視覚的に区別される。例えば、一次関数の表で上部に「式」、下部に「x と y の対応表」を配置することで、「式が上位のルールであり、表はその具体例である」という階層的構造が UI によって示される。

このように、Excel の UI は、「どのように並べるか」「どこを区切るか」を通して、学習者の思考の構造そのものをデザインする。

3.2. 抽象化 (Abstraction) — 「値」と「関係」を分けて扱う

抽象化とは、具体的な値や事例から、それらに共通する「関係」や「性質」を取り出す営みである。Excel の UI には、抽象化を促す重要な分離がある。

- セル（具体）と数式バー（抽象）の分離：セルには具体的な値が入り、数式バーには「その値同士の関係」が記述される。例えば、セル A1 に「3」、B1 に「5」、C1 に「=A1+B1」と入力したとき、学習者は「3 と 5」という具体と、「A1 と B1 の和」という抽象を別々の場所で扱うことになる。
- 参照記号による抽象化：A1+B1 という記述は、「このセルとこのセルの和」という位置に基づく抽象である。値そのものではなく「セルの関係」を書くことで、学習者は「どの値か」ではな

く「どう結びついているか」に意識を向ける。

- 式の再利用と抽象度の上昇：一度書いた式を他のセルにコピーすると、具体的な値が変わっても「関係の形」は保たれる。これは、数学における「一般式」と「具体例」の関係に近く、学習者は「式は値に依存しない関係の表現である」という抽象的理解へと導かれる。

このように、Excel は UI レベルで「値」と「関係」を分離し、関係の側に意識を向けさせることで、抽象化のプロセスを自然に支援する。

3.3. 一般化 (Generalization) — 「1つの規則を、全体へ広げる」

一般化とは、個々の事例から規則を見出し、それをより広い範囲へ適用する営みである。Excel のオートフィルは、この一般化を体験させるための極めて強力な UI である。

- 1セルから複数セルへ：あるセルに「=A1+B1」と入力し、それを下方向へオートフィルすると、「=A2+B2」、「=A3+B3」…と式が展開される。学習者は「1つの関係の形」が多数の事例へと広がっていく様子を視覚的に体験する。
- パターン認識と規則性：連続する数列 (1, 2, 3, …) や日付、曜日などをオートフィルすると、Excel は背後にある規則性を推定して補完する。これは、学習者が「並び方にはルールがある」という数学的なパターン認識を体験する場になる。
- 「式の形」が保たれることの理解：オートフィルによって式が展開されるとき、学習者は「式の形は変わらず、参照先だけが変わる」という一般化の構造を目で追うことができる。これは、一次関数の一般式「 $y=ax+b$ 」がさまざまな x に対して同じ形で適用されることと対応する。

このように、オートフィルは「1つ作る→全体へ広げる」という数学的操作を UI として具体化しており、一般化の思考を身体的・視覚的に体験させる。

3.4. 関数的思考 (Functional Thinking) — 「入力と出力の関係を見る」

関数的思考とは、「ある量が変化すると、それに応じて別の量がどのように変化するか」を関係として捉える営みである。Excel は、この関数的思考を次のように支援する。

- 入力セルと出力セルの分離：例えば、一次関数の教材では、横並びの3つのセルのうち左側を「 x の入力欄」、中央を「式」、右側を「 y の出力欄」として配置することで、「入力→関係→出力」という構造が UI によって明示される。
- グラフによる動的可視化： x の値を変更すると、グラフ上の点が移動し、直線全体の位置や傾きが変化する。学習者は「式の変化がグラフの変化にどう対応するか」を目で追うことができ、関数の本質を体験的に理解する。
- 条件付き書式との連動：例えば、ある条件を満たす行だけを強調表示することで、「入力値の条件」と「出力の振る舞い」の関係を色の変化として捉えることができる。これは、関数における「定義域」「値域」「特定条件下の振る舞い」を UI レベルで体験させるものである。

関数的思考は、単に「式を覚える」ことではなく、「入力と出力の関係を、変化として理解すること」です。Excel は、この変化をリアルタイムで可視化することで、関数的思考を強く支援する。

3.5. 最適化 (Optimization) — 「最小の操作で最大の効果を得る」

最適化とは、限られた資源（時間・手順・セル数など）の中で、目的を最も効率的に達成する方法を見出す営みである。Excel の操作そのものが、この最適化思考を自然に誘発する。

- 手順の短縮を意識する UI：同じ結果を得るために、「手入力ですべて打ち込む」のか、「式を1つ

作ってオートフィルする」のか、学習者は常に「どちらが効率的か」を考えることになる。これは、数学における「一般式を使う方が計算が楽になる」という感覚と直結している。

- 再利用可能な構造の設計：一度作った表や式を、別の問題や別のデータに対して再利用できるように設計することは、「汎用性の高い構造」を作る営みである。これは、数学における「定理」「公式」「アルゴリズム」の再利用と同じ構造を持つ。
- UI 自体の最適化：ボタンの配置、入力欄の位置、色の使い方などを工夫することで、学習者が迷わず操作できる UI を設計することは、「認知負荷の最適化」と言える。これは、学習者の思考を本質的な部分に集中させるための重要なデザイン行為である。

Excel を使う学習者は、「どうすればもっと少ない操作で、同じ、あるいはより良い結果を得られるか」を自然に考えるようになる。この経験は、数学における「よりよい方法を探す」「構造をうまく利用する」という最適化思考の土台となる。

3.6. 5 原則の相互連関

ここまで述べた 5 つの原則は、それぞれ独立したものではなく、Excel の UI/UX 上で互いに重なり合いながら働く。構造化された表は、抽象化と一般化の土台となり、抽象化された式は、関数的思考を支え、それらを効率よく扱おうとする過程で、最適化思考が育まれる。つまり、Excel の UI/UX は、「構造化→抽象化→一般化→関数的理解→最適化」という数学的思考の連鎖を、画面上の配置・操作・フィードバックを通して一つの体験として結びつけていると言える。

数学的思考の原則	Excel の UI/UX 要素	支援される学習プロセス
構造化 (Structure)	● セル構造 ● 見出し行・列 ● 罫線・色分け	● 情報の整理と構造把握 ● 「位置」に基づく関係理解 ● まとまりの認識
抽象化 (Abstraction)	● 数式バー ● セル参照 (A1, B1 など) ● 式の再利用	● 値と関係の分離 ● 一般式の理解 ● 具体→抽象の移行
一般化 (Generalization)	● オートフィル ● 相対参照 ● 繰り返し構造	● 規則の拡張 ● パターン認識 ● 一般化の体験
関数的思考 (Functional)	● グラフ機能 ● 動的更新 ● 条件付き書式	● 入力と出力の対応理解 ● 変化の可視化 ● 定義域・値域の理解
最適化 (Optimization)	● 式の再利用 ● フィルタ・並べ替え ● UI 配置 (入力欄・ボタン配置)	● 効率的な手順の選択 ● 最小操作で最大効果 ● 認知負荷の最適化

上の表は Excel の UI/UX 要素が、数学的思考の 5 原則（構造化・抽象化・一般化・関数的思考・最適化）とどのように対応しているかを整理した図である。UI/UX の設計が数学的思考の形成を直接支援する構造が示されている。

4. ケーススタディ：教材分析

本章では、筆者が開発した複数の Excel 教材 (https://ivysakura.com/04_menu/36_learningPaper/paper.html) を対象として、前章で整理した数学的思考の 5 原則（構造化・抽象化・一般化・関数的思考・最適化）がどのように UI/UX を通して学習者に体験されるかを分析する。教材はすべて Excel の標準機能（セル構造、数式、グラフ、条件付き書式等）を用いて設計されており、紙媒体では得られない動的理解を可能にしている。

4.1. 九九の教材：構造化と一般化の基盤としての UI

4.1.1. UI/UX の特徴

九九教材は、Excel のセル構造を最大限に活用した典型例である。行方向に「かけられる数」、列方向に「かける数」を配置し、交点のセルに積が表示される構造を採用している。この配置は、「かけ算の構造」を UI レベルでそのまま可視化するものである。さらに

- 行・列の見出しを色分け
- 入力セルと出力セルを明確に区別
- 誤入力を防ぐためのデータ入力規則

などの UI 設計が施されている。

4.1.2. 数学的思考の分析

① 構造化

九九の表は、かけ算の構造を「二次元の格子」として表現できる。学習者は、セルの位置関係を通して「行×列→積」という構造を自然に理解する。

② 一般化

オートフィルによって、「 $=A2*B1$ 」のような式が行方向・列方向へ展開される過程は、「かけ算の規則が全体へ広がる」一般化の体験そのものである。

③ 誤答パターンの可視化

条件付き書式により、誤答セルが自動的に色分けされる。これは、学習者が自ら誤りの傾向を発見し、構造的に理解することを支援する。

4.1.3. 教育的意義

九九教材は、構造化→一般化→誤答の発見→再構造化という学習サイクルを UI/UX によって実現できる。紙媒体では不可能な「動的な誤答の発見」が可能であり、学習者の自己調整学習を促進する。

4.2. 一次関数教材：関数的思考の可視化と抽象化の深化

4.2.1. UI/UX の特徴

一次関数教材では

- 左側に「 x の入力欄」
- 中央に「式 ($y=ax+b$)」
- 右側に「 y の出力欄」
- 下部に「グラフ」という構造

が採用されている。

x の値を変更すると、 y の値が自動計算され、グラフ上の点および直線が動的に変化する。

4.2.2. 数学的思考の分析

① 抽象化

式「 $=ax+b$ 」は、具体的な値ではなく「関係そのもの」を表現する。学習者は、セルに入力した値と数式バーの関係を通して、「式は値ではなく、関係の表現である」という抽象的理解を得る。

② 関数的思考

x の値を変えると y の値が変わり、その変化がグラフに即時反映される。これは、関数の本質である「入力と出力の対応関係」を動的に体験するものである。

③ 一般化

x の値を複数行に入力し、オートフィルで y の値を展開することで、「関数の規則が複数の入力に対して同じ形で適用される」という一般化の構造の理解を得る。

4.2.3. 教育的意義

一次関数教材は、式→表→グラフという数学的表現の三層構造を UI/UX によって統合し、学習者が「関数の多面的理解」を獲得することを支援する。

4.3. データ分析教材：構造化・最適化・統計的思考の統合

4.3.1. UI/UX の特徴

データ分析教材では

- フィルタ
- 並べ替え
- 条件付き書式
- グラフ生成

などの機能を組み合わせ、学習者がデータの構造を探索できるように設計されている。

4.3.2. 数学的思考の分析

① 構造化

大量のデータを表形式で整理し、カテゴリ・数値・条件などで分類することで、データの構造が明確になる。

② 最適化

フィルタや並べ替えは、「必要な情報を最小の操作で取り出す」という最適化思考を促す。

③ 統計的思考

平均・中央値・分散などの統計量を数式として記述することで、「データの特徴を抽象化する」という統計的思考を育む。

4.3.3. 教育的意義

データ分析教材は、数学と情報の接続を UI/UX レベルで実現し、現代的なデータリテラシーの基礎を体験的に学ばせる。

5. Excel VBA が拓く構成的学習

Excel VBA は、学習者を「使う側」から「作る側」へと移行させる。これは、構成主義的学習 (constructivism) の観点から極めて重要である。

5.1. 学習者が「作り手」になることの意義

VBA を用いて教材やツールを作成する過程は、以下の 4 つの思考プロセスを含む。

1. アルゴリズムの設計をどのような手順で処理を行うかを考える
2. UI の設計ボタンの配置、入力欄、色、動作の流れを設計する
3. 実装コードとして記述し、動作させる
4. フィードバックと改善実行結果を見て、コードや UI を修正する

これは、数学における「問題設定→解法設計→実行→検証→改良」という構成的プロセスと完全に一致する。

5.2. VBA が育てる数学的思考

5.2.1. 抽象化

VBA では、変数・関数・プロシージャを用いて「処理の一般的構造」を記述する。これは数学における「一般式」「定理」「アルゴリズム」と同じ抽象度を持つ。

5.2.2. アルゴリズム的思考

If 文、For 文、Do 文などの制御構造は、数学的な「場合分け」「反復」「条件判定」と対応する。

5.2.3. 最適化

コードを短く、効率的に書くことは、数学における「よりよい解法を探す」営みと同質である。

5.2.4. UI/UX の設計思考

学習者が迷わず操作できる UI を作ることは、認知負荷の最適化を意識した高度な設計行為である。

5.3. VBA による「動く教材」の構成的理解

VBA を用いて教材を作ることは、学習者が数学的対象を「動く構造」として理解することにつながる。例えば

- 九九の表を自動生成するコード
- 一次関数のグラフを動的に描画するコード
- データ分析を自動化するコード

などは、数学的構造を「手続き」として実装する営みである。これは、数学の理解を「知識として持つ」から「構造として作る」へ転換させる。

5.4. 教育的意義

Excel VBA は、

- 数学的思考
- 情動的思考
- UI/UX デザイン思考

を統合する学習環境であり、学習者が自ら学習環境を構築するという高度な構成的学習を可能にする。

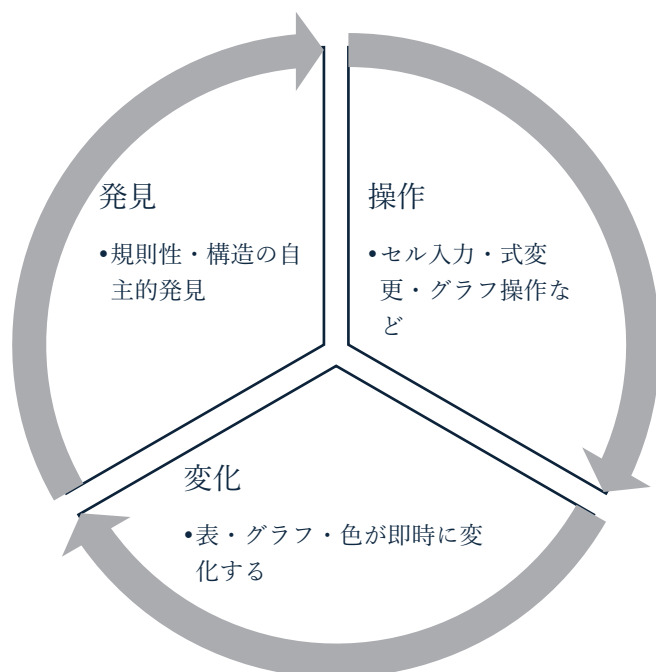
6. UI/UX×数学教育の新しいモデル

本章では、前章までの理論的整理および教材分析を踏まえ、Excel の UI/UX を基盤とした数学教育の新しいモデルを提示する。このモデルは、従来の紙媒体中心の教材では実現が困難であった動的・構成

的・探究的な学習を、UI/UX の設計原理を通して体系化するものである。

6.1. 「操作→変化→発見」の学習サイクル

Excel の UI/UX がもたらす最大の特徴は、学習者の操作に対して即時にフィードバックが返る点にある。この特性は、数学的理解を深めるための「操作→変化→発見」という学習サイクルを自然に形成する。



Excel の即時フィードバック機能により、学習者は「操作→変化→発見」の循環的学習サイクルを経験する。このサイクルは構成主義的学習の基盤であり、数学的構造の理解を深める重要なプロセスである。

6.1.1. 操作 (Action)

学習者はセルに値を入力し、式を変更し、グラフを動かす。この操作は、数学的対象に対する「介入」であり、学習者が能動的に学習過程へ参加する契機となる。

6.1.2. 変化 (Feedback)

操作に応じて、表・グラフ・色などが即時に変化する。この変化は、数学的関係を視覚的・動的に表現するものであり、学習者は「自分の操作が数学的構造にどのような影響を与えるか」をリアルタイムで観察できる。

6.1.3. 発見 (Discovery)

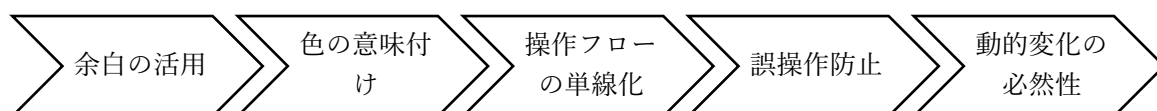
変化を観察することで、学習者は規則性・構造・関係性を自ら発見する。これは、構成主義的学習における「学習者が自ら意味を構築する」過程に対応する。

6.1.4. サイクルとしての統合

この3段階は循環的に働き、学習者は操作を繰り返す中で理解を深める。紙媒体では、操作と変化の間に手作業が介在するため、このサイクルは断続的になりやすい。Excel の UI/UX は、このサイクルを連続的・高速・低負荷で実現する点に教育的価値がある。

6.2. UI/UX に基づく教材デザインの原則

筆者の教材開発実践から抽出された UI/UX ベースの数学教材デザイン原則を以下に整理する。



上記は UI/UX ベースの教材デザイン原則を示した図である。これらの原則は、学習者の認知負荷を最適化し、数学的構造の理解を支える UI 設計の基盤となる。

6.2.1. 余白を活かす (CognitiveSpace)

余白は、情報の区切りを示し、学習者の認知負荷を軽減する。数学的構造を理解するためには、「どこに何があるか」が明確であることが重要であり、余白はその視覚的手がかりとなる。

6.2.2. 色に意味を持たせる (SemanticColoring)

色は単なる装飾ではなく、数学的意味を持つ視覚的記号として機能する。例えば、入力欄を青、出力欄を緑、誤答を赤で示すことで、学習者は「どこを操作すべきか」「何が起きたか」を直感的に理解できる。

6.2.3. 操作の流れを一本道にする (LinearInteractionFlow)

学習者が迷わず操作できるよう、UI は「一本道」の流れを持つべきである。入力→計算→結果→グラフという一方向の流れは、数学的思考の流れと一致する。

6.2.4. 誤操作を防ぐ UI (ErrorPrevention)

データ入力規則、保護セル、条件付き書式による警告などは、学習者の誤操作を防ぎ、思考の流れを中断させないために重要である。

6.2.5. “動く” ことに必然性を持たせる (MeaningfulDynamics)

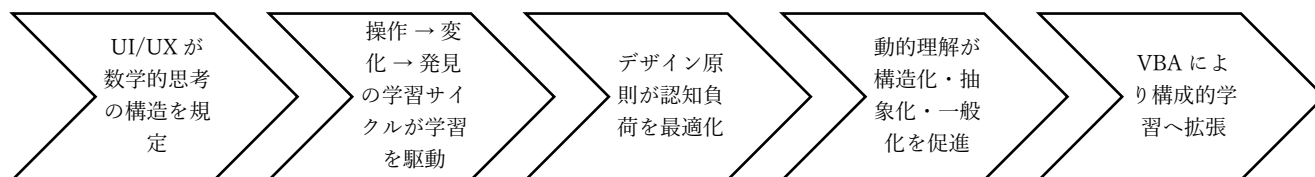
動的变化は、数学的意味を持つ場合にのみ用いるべきである。例えば、一次関数のグラフが動くことは「関数の変化」を理解するために必然的であるが、意味のないアニメーションは認知負荷を増大させる。

6.3. UI/UX×数学教育モデルの統合図式

以上の議論を統合すると、Excel を用いた数学教育は以下の特徴を持つ。

1. UI/UX が数学的思考の構造を規定する
2. 操作→変化→発見のサイクルが学習を駆動する
3. デザイン原則が学習者の認知負荷を最適化する
4. 動的理解が構造化・抽象化・一般化を促進する
5. VBA によって構成的学習へ拡張される

このモデルは、数学教育における「静的理解」から「動的理解」への転換を示すものであり、Excel の UI/UX がその中心的役割を担っている。



Excel の UI/UX を基盤とした数学教育モデルを統合的に示した図である。UI/UX→学習サイクル→デザイン原則→動的理解→構成的学習という階層的構造が、数学的思考の形成を支える。

7. 結論

本論文では、Excel を数学的思考をデザインし、学習者に体験させるための学習環境として再定義し、その教育的可能性を UI/UX の観点から理論的・実践的に検討した。

7.1. 本研究の主要な知見

本研究から得られた主要な知見は以下の通りである。

7.1.1. Excel の UI/UX は数学的思考の 5 原則を支援する

セル構造、数式バー、オートフィル、グラフ、条件付き書式などの UI は、構造化・抽象化・一般化・関数的思考・最適化という数学的思考の主要要素を自然に引き出す。

7.1.2. 動的理解を中心とした学習サイクルが形成される

Excel の即時フィードバックは、操作→変化→発見の学習サイクルを生み出し、学習者の能動的・探究的な理解を促進する。

7.1.3. UI/UX に基づく教材デザイン原則が学習効果を高める

余白、色、操作フロー、誤操作防止、動的变化の必然性など、UI/UX の設計原則は、学習者の認知負荷を最適化し、数学的構造の理解を支える。

7.1.4. VBA は構成的学習を可能にする

VBA による教材開発は、学習者を「使う側」から「作る側」へと移行させ、数学的構造を自ら設計・実装するという高度な構成的学習を実現する。

7.2. 本研究の意義

本研究の意義は、Excel を単なる表計算ソフトとしてではなく、数学的思考を体験としてデザインするための学習環境として位置づけた点にある。従来の数学教育では、「手続きの習得」や「静的な理解」が中心であったが、Excel の UI/UX は、動的・構造的・探究的な理解を可能にする。

7.3. 今後の展望

今後の研究課題として、以下が挙げられる。

- UI/UX に基づく教材デザインの体系化
- 学習者の操作ログを用いた理解過程の分析
- VBA を用いた構成的学習の発達段階モデルの構築
- 数学教育と情報教育の統合カリキュラムの開発

これらの課題に取り組むことで、Excel を中心とした数学教育の新しい地平が開かれるだろう。

7.4. 結語

Excel は、数学的思考を“動かしながら理解する”ための現代的な学習環境である。UI/UX の視点を取り入れることで、算数・数学教育はより直感的で、構造的で、探究的なものへと変わる。筆者の教材開発の実践は、その未来の具体例として位置づけられる。

8. 参考

- [1] 金沢愛美 (2026) 「Excel で数学的思考をデザインする：授業実践例・ケーススタディ」
ivy's vie (https://ivysakura.com/04_menu/36_learningPaper/paper.html)