

Squeak による「関数とグラフ」の学習におけるシミュレーション環境の開発

丸井 美保*・香月 保彦*

Development of Simulation Environment for Learning of Function and its Graph by Squeak

Miho MARUI* and Yasuhiko KATSUKI*

The programming environments “for children” allow not only for children to learn programming, but also for teachers to develop teaching materials of various subjects. For example, Squeak is one of them. The developed teaching materials can purvey the simulation environments in the classroom. It will be necessary to study the effective use of the materials on each teaching content. Here, we concentrate on function and its graph that should be learned from arithmetic in primary school to mathematics in high school. We develop the prototype of the material for drawing graphs of functions and translating graphs by Squeak, and discuss the way to use it. As a result, we found it would make children understanding concrete by using it after drawing graphs by their hands, and deep when it is used repeatedly at many school terms. It, furthermore, would be modified and/or extended to adjust other contents that should be presented by graph. We also found that the basic scripts in the material can be constructed simply enough that children be able to read and extend those by themselves. This fact will possibly lead to the consequence of the growth of their interest in arithmetic, mathematics, and/or other subjects.

Key Words (キーワード)

Squeak (スクイーク), Simulation (シミュレーション), Teaching material (教材), Arithmetic and Mathematics (算数科・数学科), Function and its Graph (関数とグラフ)

1. はじめに

コンピュータの処理能力、ソフトウェア技術の急激な向上を背景に、教育の現場での ICT の活用への期待が高まるとともに、実際に教育の手法を変えつつある。その領域は情報リテラシ、プログラム開発能力、論理的思考の育成だけでなく、各教科の学習にも及んでいる。ここで、ICT、特に学習環境を左右するソフトウェアの適切性は、開発当初の主たる目的や上記の領域でなく、学習目標に依存する¹⁾。本研究では、算数科・数学科の学習における”子ども向け”プログラミング環

境の活用について考察する。教育現場や子どもの自学習での利用を強く意識した“子ども向け”プログラミング環境として Smalltalk を基にした Squeak²⁾、Scratch³⁾、Alice⁴⁾ や文部科学省のウェブサイト上で提供されているプログラミン⁵⁾などが挙げられる。これらに関する教育上の手法、効果などに関するさまざまな理論的、実践的な研究が進められている⁶⁾が、更に多方面からの検討が必要と考える。なお、以下では、“子ども向け”プログラミング環境について検討するにおいて、学習者を子どもと表記する。

* 広島文化学園大学 社会情報学部 (Faculty of Social information Science, Hiroshima Bunka Gakuen University)

2. “子ども向け”プログラミング環境を活用した教科教育

2.1 教科教育における活用

Squeak をはじめとする上記のソフトウェアでは、子どもがプログラミングを学べるように、プログラミング言語に関する高度な知識を要求せず、プログラミング環境と実行環境を厳密に区別せず、容易にディスプレイ上のオブジェクトを動かすことができる。しかし、これらの用途はプログラミング学習に限られるものではなく、オブジェクトをそのときの学習対象に見立てることにより教科教育の動的な教材、すなわち、シミュレーション教材、或いは、その開発環境として活用できる。これらにより開発した教材は、多様な条件（パラメータ）で処理を行うことができ、教室の環境で再現できない事象をオブジェクトやその動きにより観察することができる。また、時間的制約において省略せざるを得ない繰り返しによる学習活動（多数の例示など）を補うことができる。これらの一般的なシミュレーションプログラムの利点を持つことに加え、“子ども向け”プログラミング環境は、次の観点から教材、及び、教材開発環境として有効であると考えられる。

(1) 拡張性が高い

オブジェクトや動きに修正や追加を加えることが容易であり、学習内容にあわせて拡張することができる。これにより、カリキュラムの年次進行などから、時間において関連項目を学習するとき、同じ教材を用いることができる。また、オブジェクトに何を見立てるかという解釈を変えれば、同一項目、同一科目だけでなく、異なる教科の教材として拡張できる可能性もある。

(2) 発展的学習に活用できる

教材は、子どもと教員で共有でき、子どもは発達の段階にあわせて自学習で活用できる。また、教科教育における活用は、子どもがこれらの環境に接する機会を増やし、自由な発想で教材を拡張

していくことで、ソフトウェア本来の目的である論理的思考の育成に繋がる可能性がある。

ただし、上記はコンピュータ上の環境に関する利点であり、教育効果については、従来の静的な教材・教具との併用による効果を含め、学習項目ごとに検討しなければならない。

2.2 算数科・数学科における関数とグラフの学習への活用

平成24年度より実施されている教育指導要領⁷⁾から、中学校数学科の「数量関係」の領域が「関数」に変更された。これは、関数の概念の習得を重視したもので、国の教育課程の基準全体の見直しについての検討の要請に対する中央教育審議会の答申（平成20年1月）⁸⁾を踏まえたものである。また、同答申は、「数量や図形の意味に関する実感的な理解」、「発達や学年の段階に応じた反復（スパイラル）による理解の広がりや深まり」、「日常生活、他教科等の学習、より進んだ算数・数学の学習への活用」の重要性について言及している。関数の学習は、小学校第5学年の比例の関係から高等学校の指数関数・対数関数まで段階的に進められる。指導要領で述べられているように、一つひとつ丁寧に点をプロットしてグラフを描く活動やコンピュータなどを活用して様々なグラフをかき、その特徴を帰納的に見いだしたりする活動は、基本的・基礎的理解において重要である。また、文献1)で言及されるように、コンピュータを用いた作業は、「一つ」の関数のグラフの概形を捉えることが目標の場合は、単なる数値の入力と結果の観察というつまらない作業に陥りがちであるが、関数「群」の特徴を捉えたい場合には有効である。限られた授業時間内に、「群」の単位でグラフを描き、観察する活動において、GRAPES⁹⁾など関数グラフ作成ソフトのほかに、シミュレーション教材が有効であると考えられる。

また、シミュレーションによれば条件（パラメータ）を変更することで様々なグラフを扱うことができ、単元や学年を限定せず、共通の学習環境を提供する教材を開発することが可能であり、スパ

イラルによる学習に対応できる。更に、関数の概念、式、グラフは、さまざまな自然現象、社会現象を表現するものであり、日常生活、他教科に関連するため、ここでの教材開発は、必然的に他教科での活用を視野に入れたものとなる。ただし、上記の教育効果は、子どもが迷わず操作できるユーザインターフェースを実現できた場合にのみ望めるものである。

2.3 Squeak による「関数のグラフ・グラフの移動」のシミュレーション教材の試作

“子ども向け”プログラミング環境は、いずれもオブジェクト指向であり、幅広い目的に対応できるものである。しかし、主たる目的の違いから、プログラミング環境において扱いやすい形式で提供される機能が異なる。Scratch や Alice はアニメーション、マルチメディアコンテンツの作成を想定し、その機能に優れている^{3,4,10,11)}。これに対して、Squeak は、オブジェクトの運動とその観察に主眼が置かれている¹²⁾。

Squeak ではモーフ (Morph) と呼ばれるグラフィカルなオブジェクトを中心に環境が構築される。モーフで呼び出される一連のプログラムコードは、スクリプト (Script) と呼ばれる。スクリプトを作成する方法は2つある。一つは、スクリプトエディタでコードを記述する方法である。もう一つは、モーフに関する性質や動きを定義するタイル (Tile) を組み立てていくタイルスク립ティングと呼ばれる方法である。これにより、高度なプログラミング言語の知識を持たずに利用することが可能になっている。また、モーフは互いにスクリプトを呼び出すメッセージ (Message) を送り合うことができ、スクリプトとメッセージを組み合わせることで、シミュレーションの機能、及び、ユーザインターフェースを適切なものにできる。モーフには、楕円、四角形、線などさまざまな形状のものがあ、グラフの構成要素として充分である。また、タイルによりモーフの位置、大きさ、向きなどの幾何情報を扱うことが容易であり、グラフのプロットや移動を表現するのに適し

ている。そこで、本研究では「関数のグラフ・グラフの移動」に関するシミュレーション教材を試作し、教材としての可能性を検討するためにスクイークを用いることにする。なお、子どもの発展的な学習や教員同士の教材の共有を実現するため、可能な限り、タイルスク립ティングで作成する。

3. 試作した教材

3.1 概要

試作したシミュレーション環境「関数のグラフ・グラフの移動」は、次の4種のシミュレーションを行うものである。

- (1) 関数のグラフ描画 (点のプロット、線)
- (2) 移動に関する値の指定によるグラフ移動
- (3) マウス操作によるグラフ移動
- (4) グラフの目盛りの変更

この画面構成は図1のとおりである。画面左側にはグラフと目盛りに関するフィールドを置き、グラフの観察と(4)のパラメータ(目盛り)を指定する領域とする。関数のグラフ上の点に相当する円形のモーフは、(1)、(2)、(3)それぞれについて41個をx軸方向に関して等間隔に配置する。画面右側には、(1)、(2)、(3)それぞれについて、式や移動に関するパラメータの入出力とシミュレーション開始と停止のためのボタンを含むフィールドを配置する。なお、それぞれのシミュレーションについて、ユーザが直感的に操作できるように、グラフ上の点・線、及び、パラメータフィールドなどの関連するモーフを同一色で対応づける。(1)は緑、(2)は青、(3)は赤、(4)は黒である。以下では、パラメータに関するフィールドをそれぞれ緑フィールド、青フィールド、赤フィールド、黒フィールド、グラフ上の点を緑の点、青の点、赤の点と呼ぶ。

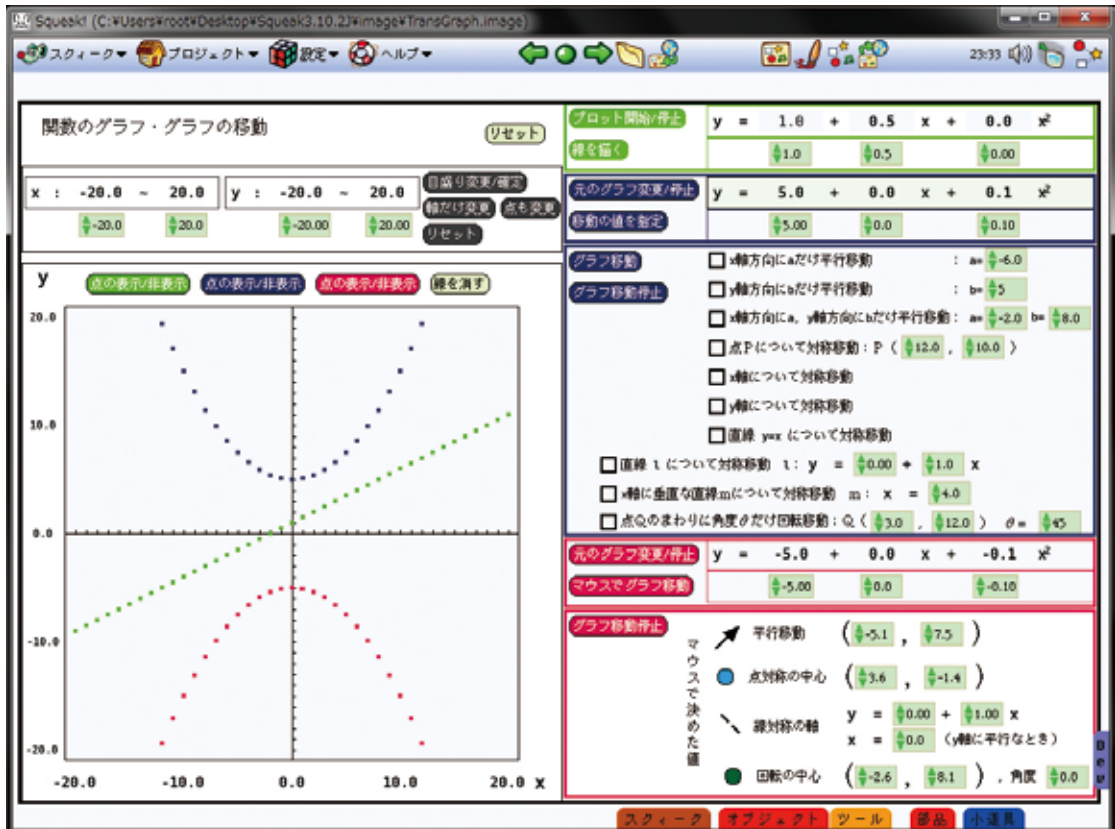


図1 試作したシミュレーション環境「関数のグラフ・グラフの移動」

3.2 シミュレーションの内容

(1) 関数のグラフ描画

・ユーザインターフェース

このシミュレーションのパラメータ(入力)は、式の係数であり、出力はグラフ上にある緑の点の座標である。シミュレーションの開始と停止のきっかけは、緑フィールドの「プロットの開始/停止」ボタンに任せ、実行中はシミュレーションのスクリプトを常に動作させ、パラメータの変更に伴い、同期するように緑の点を移動させる。また、「線を引く」ボタンは、緑の点を結ぶ線を引くものである。このボタンを押すことで、以後にプロットするグラフと比較する必要があるグラフを線として残すことができる。入力例を図2に示し、図3に出力例として、 $y = -5 + 0.5x$ のグラフ、及び、 $y = 5 + 0.5x$ のグラフをプロットした段階で線を引く、続いて $y = 10 + x$ となるよ

うにパラメータを変更した結果を示す。

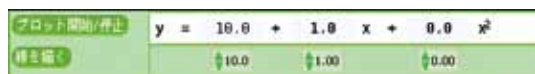


図2 (1) の入力の例

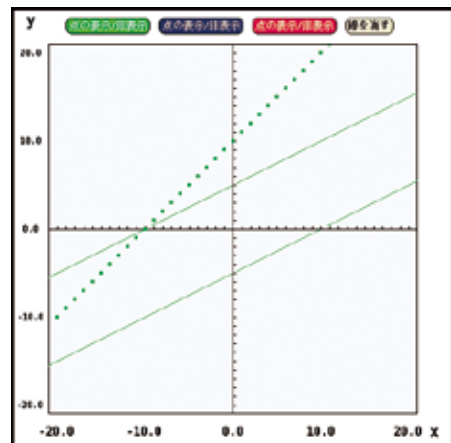


図3 (1) の出力の例

・スクリプト

Squeak では、同じスクリプトを有する兄弟モーフを作ることができる。又、モーフごとに変数を定義でき、その変数を表すタイルを配置すれば、変数への入出力が可能となる。更に、スクリプトやモーフは他のモーフに対してスクリプト実行のメッセージを送るとき、単なる「実行する」だけでなく、一定時間間隔で動作するように「スタートする」、「兄弟の皆に送る」ことが可能である。これらにより、ここでのシミュレーションは次の構成で、関数の値を求める四則演算を含め、タイルスクリプティングだけで実現できる。

- i. 「プロットの開始/停止」ボタンのスクリプト：
 - ii. のスクリプトにスタート或いはストップするメッセージを送信する。
- ii. 緑フィールドのスクリプト：
 - iii. のスクリプトを実行するメッセージを送信する。
- iii. 緑の点のスクリプト：
 - iv. それ自身の x 座標から、関数の式に従い y 座標を求め、その位置に移動する

(2) 移動に関する値の指定によるグラフ移動

・ユーザインターフェース

ここでは、グラフの平行移動、点対称移動、線対称移動、回転移動を扱う。ただし、段階的な学習に対応できるように平行移動は、「 x 軸方向に a だけ平行移動」、「 y 軸方向に b だけ平行移動」、「 x 軸方向に a 、 y 軸方向に b だけ平行移動」のいずれかをユーザが指定できるように作成する。線対称移動も同様に、「 x 軸について対称移動」、「 y 軸について対称移動」、「直線 $y = x$ について対称移動」、「直線 $l (Y = C_0 + C_1 X (C_1 \neq 0))$ について対称移動」、「 x 軸に垂直な直線 $m (X = C_x)$ について対称移動」に分類する。

これらのシミュレーションのパラメータは二種類ある。一つは、元（移動する前）のグラフに関する式の係数であり、(1) と同様に指定できる。

もう一つは、移動に関する値である。図4のように青フィールドで移動の種類を選択するとともに、そのパラメータを指定する。ここでのパラメータは、上記の a 、 b 、 C_0 、 C_1 、 C_x に加えて、点対称移動の中心 P の座標 (x_p, y_p) 、及び、回転移動に関する中心 Q の座標 (x_q, y_q) と角度 θ である。シミュレーションの出力は青の点の座標であり、実行のきっかけは「グラフ移動」のボタンである。ボタンを繰り返し押し出すことで、青の点に対して次々に移動を施し、それを観察することができる。図5に出力例として、 $y = 5 + 0.1x$ のグラフを点 $Q (0, 0)$ を中心に 90 度回転移動し、更に、 x 軸方向に 15 だけ平行移動した結果を示す。

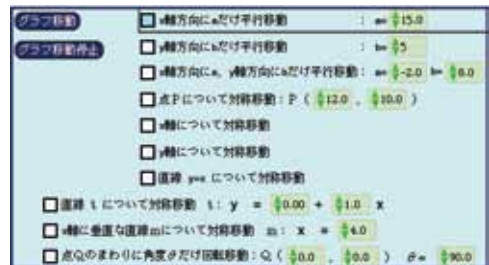


図4 (2) の入力 (パラメータ指定) の例

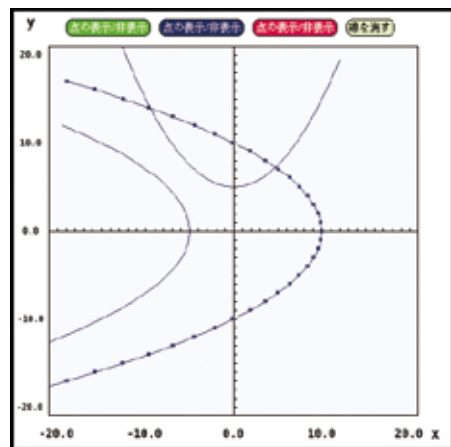


図5 (2) の出力の例

・スクリプト

元のグラフに関するスクリプトは (1) と同様に作成し、移動に関しては i 、 ii に示す構成とする。

- i. 「グラフ移動」ボタンのスクリプト：

青の点すべてに ii のスクリプトを実行する

メッセージを送信する。

ii. 青の点のスクリプト：

青の点自身の現在の座標 (X, Y) をもとに、次式に従い、移動後の座標 (X', Y') を求め、その位置に移動する。ここで、 $a, b, x_p, y_p, C_0, C_1, C_x, x_q, y_q, \theta$ は、前述のパラメータ、 (r, a) は現在の青の点の幾何情報の一部であり、回転移動の中心 Q を原点とするときの極座標に相当する。

平行移動：

$$X' = X + a, \quad Y' = Y + b.$$

点対称移動：

$$X' = 2x_p - X, \quad Y' = 2y_p - Y$$

線対称移動：

・対称軸 l が $Y = C_0 + C_1X$ ($C_1 \neq 0$) のとき、

$$X' = \frac{1}{C_0^2 + 1}(X + C_0Y - C_0C_1),$$

$$Y' = \frac{C_0}{C_0^2 + 1}(X + C_0Y - C_0C_1) + C_1.$$

・対称軸 m が $X = C_x$ のとき、

$$X' = 2C_x - X, \quad Y' = Y.$$

回転移動：

$$X' = x_q + r\cos(a + \theta),$$

$$Y' = y_q + r\sin(a + \theta).$$

このスクリプトを作成するにあたり、回転移動において三角関数を用いる部分はエディタでコードを記述せざるを得ないが、その他はタイルスクリプティングが可能である。

(3) マウス操作によるグラフ移動

・ユーザインターフェース

赤フィールドで、平行移動、点対称移動、線対称移動、回転移動のいずれかを選択し、この移動を行う。ただし、パラメータは数値で指定せず、次の i, ii によりグラフ上でマウスを動かすことでシミュレーションを実行する。

i. 移動のインジケータに相当するモーフをグラフ上に配置し、マウスの動きに連動させ、その幾何情報（位置、大きさ、向きなど）か

らシミュレーションのパラメータの値を決定する。

ii. シミュレーション開始のきっかけは矢印型のインジケータをクリックすることとし、これにより赤の点の移動後の座標が求められ、実際に点が移動する。

図6と図7に、インジケータとシミュレーションの出力の例を示す。いずれも $y = -5 - 0.1x^2$ のグラフからの移動である。図6は、点 $(-5, 0)$ についての点対称移動、図7は $Y = 5 + x$ を軸とする線対称移動である。

なお、平行移動のインジケータは赤の代表点から移動先に延びる矢印型のモーフである。回転移動のインジケータは、中心点と回転角を示すものであり、図7の線対称移動の軸を示すインジケータと同じ形状である。また、マウスで指定されたパラメータの値は赤フィールドに表示され、確認できる。

・スクリプト

Squeak では、コーディング（スクリプティング）環境とユーザによる実行の環境を区別せず、モーフに関して入力用・出力用のいずれかに用途を制限することを前提としない。その発想による環境では、シミュレーション実行中にモーフをマウスで掴み動かすこと、その幾何情報を利用することを想定したスクリプト設計が容易である。上記のユーザインターフェースは、この特性を利用し、変化するインジケータの幾何情報をグラフ移動のパラメータ変数に送ることで実現する。確定したパラメータの値に対する赤の点の動きは(2)のスクリプトを利用できる。なお、グラフの位置が連続的に変化する平行移動、回転移動に関しては、より直感的に赤の点を掴み、移動することで、インジケータを介さずにシミュレーションを実現できる。しかし、ここでは、点対称移動、線対称移動とあわせて統一的なユーザインターフェースとするために、敢えてすべてインジケータを介することとする。

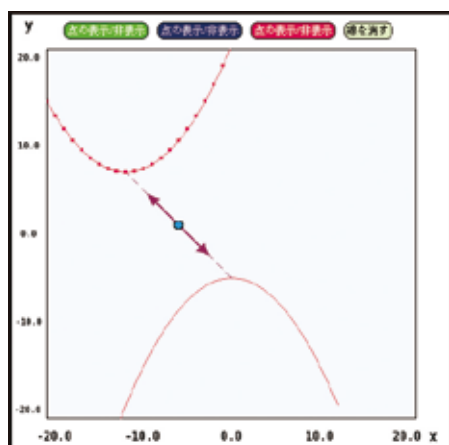


図6 (3) 点対称移動のインジケータと出力の例

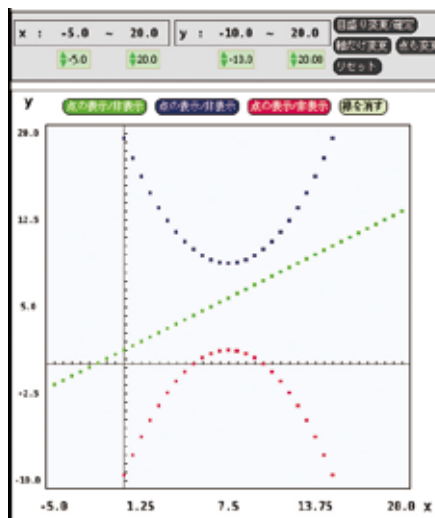


図8 (4) の入力と出力の例

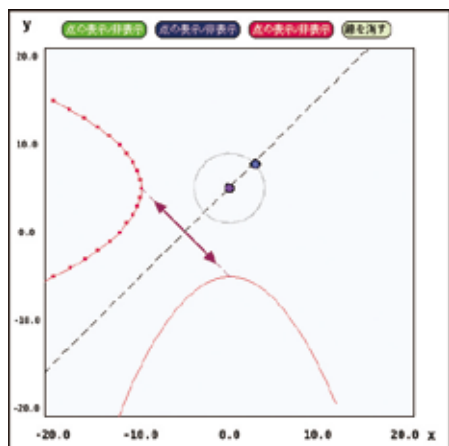


図7 (3) 線対称移動のインジケータと出力の例

(4) グラフの目盛りの変更

パラメータとして、黒フィールドで、 x 軸と y 軸について、それぞれ最小値と最大値を指定する。「軸だけ変更」ボタンで、 x 軸と y 軸を移動し、グラフの目盛りを変更する。「点も変更」ボタンで、「軸だけ変更」の動きに加えて、(1), (2), (3)で作成したスクリプトを実行し、緑の点、青の点、赤の点をプロットし直す。図8は、パラメータの指定(入力)と出力の例であり、図1で示した状態に「軸だけ変更」を行った結果である。

4. 教材としての可能性の検討

(1) 関数のグラフ描画

試作した教材は、3.2に示したようにユーザが設定したパラメータの値に対応するグラフを描く。マウスだけで操作できるため、教員が授業などで見せるグラフや、子ども自身が作成するグラフの数を飛躍的に増やすことができる。また、パラメータの変化に連動して、次々にグラフを描くので、パラメータとグラフの形の関係、グラフ同士の関係、対称軸や頂点との位置関係を観察するのに有効である。一方、表計算ソフトなどでも、グラフを描くことは可能であるが、操作法を先に修得する必要がある、パラメータの変更に対して行うべき操作が多く、観察だけに集中することは難しい。

また、ここでは、画面上の式の表示を2次関数にあわせて整えたが、2次の項を隠すなどの作業はSqueakでは容易にできる。更に、 x の値に対応する y の値を求めるただ1つのスクリプトを変更するだけで、この他の関数のシミュレーションにも対応できる。このスクリプトは図9に示すように平易なもので、タイルスクリプティングで作成されている。スクリプトの引数である数値が x に対応し、 fx の fx (fx という名前のモーフが持

つ変数 fx) が y に対応するものである。小学校から学習が始まる反比例の式 $y = 1/x$ に相当する項を加えるには、「 fx の $fx \leftarrow fx$ の $fx + 1$ /数値」というフレーズをタイルで組み立てれば良い。高等学校で学ぶ指数関数や対数関数を実現するには、このスクリプトをエディタで表示し、Squeak が持つ数学関数を用いて変更を加えれば良い。このように、小学校から高等学校までの関数の学習で共通に活用できる教材とすることができること、拡張そのものが容易であることがわかる。



図9 関数の式にもとづき x の値から y の値を決めるスクリプト

以上から、試作した関数のグラフ描画に関して、教育の現場で有効に活用できると考えるが、次の注意が必要である。手作業でグラフを作成する場合は、グラフの点をプロットする前に、パラメータを決めた式に従い、 x の値に対応する y の値を求めるといった過程を経る。本教材では、これに対応するスクリプトの動作はディスプレイ上で表示していない。ただし、Squeak では変数の値を観察するためのタイルを表示でき、この過程を意識できるユーザインターフェースへの修正は可能である。しかし、数学的な概念の定着には、一般的に丁寧な手作業による活動は重要である。 x の値から y の値が決まるという関数の本質的な概念を修得させるにあたっては、手作業でのグラフ作成活動を省略すべきでないとする。

(2) グラフの移動

移動に関するパラメータを少しずつ変化させて、それに連動してグラフを変化させることで、移動を実感的に理解するとともに、平行移動、点対称移動、線対称移動、回転移動などの概念を明確にするのに役立つと考える。表計算ソフトで、移動後のパラメータの値をあらかじめ求めてお

き、グラフを表示する手法も考えられるが、これは実際にオブジェクトを移動させるものではなく、移動前と移動後の点の対応関係が明確にならない。これを子ども自身が行う場合は、移動を理解した後の学習活動にしか適用できず、移動の概念の理解には試作したようなシミュレーション環境の方が有効と考えられる。

また、移動後のグラフを更に移動できるので、2次関数 $y = ax^2$ から、平行移動を介して到達する一般的な2次関数 $y = ax^2 + bx + c$ の理解、教員による解説に活用できる。これは高等学校の数学 I で学習する内容である。一般的に、移動など動的な現象が、理解における本質的な役割を果たす概念では、板書や印刷物などの静的な教材に比べて、シミュレーション教材による効果が高いと予想する。

関数のグラフの一般的な対称移動は、算数科・数学科の指導要領で直接的に示される学習の範囲を超える。しかし、「図形」の領域では、中学校1年で点対称移動、線対称移動、回転移動を扱う。図形の移動により、その図形を表す式のパラメータがどのように変化するかを観察することなどは、項目を超えた総合的理解につながる。理数に興味・関心の高い子どもの自学習に適するテーマであり、そのための教材の1つになると考える。

(3) マウス操作によるグラフ移動

グラフ領域内のオブジェクトをマウスで動かすという操作は、グラフの移動を実感するのに適している。特に、グラフ上の点が連続的に移動する平行移動や回転移動は、図10の例のようにグラフそのものを掴み、動かすように操作ができる。ここで、図10は $y = -5 + 0.1x^2$ のグラフを x 軸方向に10だけ、 y 軸方向に15だけ平行移動したものである。また、その移動に関するパラメータの値が表示されるので、実感を伴いつつ、数学的に正しい理解を促すことができる。静的な出力を主たる目的とするワープロソフト、表計算ソフト、作図ソフトでも、図形を移動したり回転したりすることはできるが、移動に関するパラメータの値

やグラフの式との関連を示すには、充分ではない。

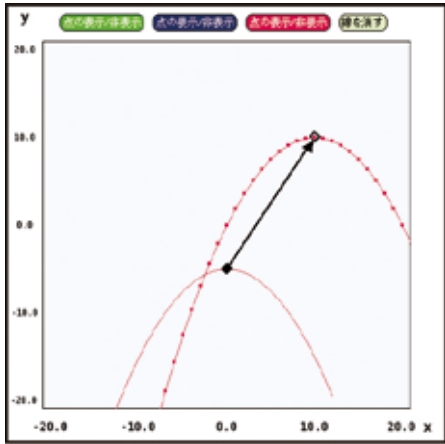


図10 (3)の平行移動のインジケータと出力の例

更に、Squeakは特定の命令セットを実行するVirtual Machineに対応できるOSであれば、どのようなプラットフォームでも動作する。よって、教材は、iPadなどタッチデバイスによる機器でも3.で示したと同様に、開発と実行ができる。タッチデバイスを用いる場合、マウス操作を指での操作に置き換えることで、より直感的・直接的なグラフ移動の学習ができると考える。教育の現場におけるタッチデバイスによる機器の導入が進みつつあり、各教科での活用と教育効果の検討は急務と考える。

(4) グラフの目盛りの変更

グラフを用いた学習は、算数科・数学科だけに限らず、他教科でも行われる。試作した環境は、任意の単位(目盛り)に対応でき、関数の式のパラメータもユーザが決定できるので、他教科の教材として活用することも可能である。“子ども向け”プログラミング環境では、教科教員のプログラミング言語に関する知識に大きく依存することがないため、教科間の教材共有の可能性は高いと予想される。

また、(1) グラフの描画や(2)、(3)のグラフの移動に対して、先に目盛りを確定して実行するだけでなく、それらの結果を残したまま、グラフの目盛りだけを変更することができる。これによ

り、座標変換と移動変換の関係を観察することが可能である。これは、数学における重要な概念であり、自学習のための教材の1つになると考える。

(5) 子ども自身による発展的学習

Squeakは、コンピュータそのものの環境に依存せず、そこで作成されたシミュレーション環境はプロジェクトファイルとして教員間や教員と子ども間で共有できる。試作した「関数のグラフ・グラフの移動」も、そのただ一つのプロジェクトファイルを複製して、子ども自身の有するコンピュータで動かすことができる。

Squeakは、本来、子どもが自由な発想で利用することを重視する環境である。特に、本研究で試作した環境は、最も基本的な機能であるモーフの移動を主として構成しており、教科の学習活動にこの発想を活かすことができるものと考えられる。例えば、図11で示したスクリプトは、(2) グラフの移動において青の点の1つ(pB000)に定義した平行移動に関するスクリプトである。この1行目は、現在(移動前)のx座標に移動分(青フィールドtransに定義した変数shiftXdotの値)を加えた値を点のx座標に代入するもので、移動の意味をここで改めて読み取ることができる。2行目はy座標に関するフレーズ、3行目は平行移動後の処理のためのフレーズである。このように、スクリプトの多くは、子どもが一人で読み、それにより関数のグラフ、グラフの移動について理解を深めるのに有効であると考えられる。

また、教材を基にプログラミングを学習できること、Squeakによるプログラミングを体験している場合は自由な発想による拡張で、異なる何かに気づく可能性にも注目すべきと考える。



図11 グラフの点を表すモーフに定義した平行移動を行うスクリプト

5. まとめ

“子ども向け”プログラミング環境の用途は、プログラミング学習に限られるものではなく、オブジェクトに見立てる対象によってさまざまな教科教育の動的な教材、すなわち、シミュレーション教材として活用できる。その教育効果を検討することは重要であり、学習項目ごとにこれを行う必要がある。本研究では、算数科・数学科の新指導要領（平成24年度より実施）において、以前より重視されることになった「関数の概念の習得」について検討を行った。Squeakによるシミュレーション教材を試作し、それを具体的に検討した結果、シミュレーション教材が、指導要領の範囲、及び、その発展的学習の範囲で、「実感的な理解」、「発達や学年の段階に応じた反復（スパイラル）による理解の広がりや深まり」、「他教科等の学習、より進んだ算数・数学の学習への活用」に有効であることを示した。また、教科間、教員間での教材共有、それぞれの内容に適合させる調整や拡張が可能であること、及び、その主なコードはタイルスクリプトだけで構築でき、子ども自身が読み取り、拡張できることを示した。ただし、学習効果は、従来の手作業によるグラフ作成の活動などとの組み合わせで期待されるものである。他の教材との併用の観点から教育プログラムを検討し、試作した環境をこれに基づき改め、学習効果に関する実践研究を行うことが課題として残っている。

参考文献

- 1) 飯島 康之, 1995, 作図ツールの導入によるカリキュラム研究のための基礎的考察－数学教育研究に対するテクノロジーの影響について－, 数学教育論文発表会論文集, 28, 519-524.
- 2) <http://www.squeak.org/>(Squeak)
<http://squeakland.jp/index.html/> (ようこそ, スクイーランドへ!)
- 3) <http://http://scratch.mit.edu/>
(SCRATCH|Home|imagine・program・share)
- 4) <http://www.alice.org/index.php> (Alice.org)
- 5) <http://www.mext.go.jp/programin/> (プログラミング | 文部科学省)
- 6) ・稲垣卓弥・阿部和広・山崎謙介・横川耕二, 2009, 「教具」としての Squeak eToys とその小学校算数への適用, 社団法人情報処理学会研究報告, 2009-CE-98 (9)
- ・澤谷拓郎・高数学・村田晴紀, 2010, EEP <Environment for Education by Programming> の教材活用法の検討～LOGO, Squeak, Scratch, Aliceの比較を通じて～, 東京学芸大学紀要 人文社会学系Ⅱ, 61, 173-179.
- ・村田晴紀・高数学・澤谷拓郎, 2009, Squeakを利用した課題解決能力育成のための教材開発および実践研究の検討, 東京学芸大学紀要 人文社会学系Ⅱ, 60, 161-168. 他
- 7) 文部科学省,
小学校学習指導要領, 平成20年3月,
中学校学習指導要領, 平成20年3月, 平成22年11月一部改正.
高等学校学習指導要領 平成21年3月.
- 8) 中央教育審議会, 幼稚園, 小学校, 中学校, 高等学校及び特別支援学校の学習指導要領等の改善について(答申)(平成20年1月)
- 9) <http://www.osaka-kyoiku.ac.jp/~tomodak/grapes/>
(WelCome To GRAPES)
- 10) Jerry Lee Ford, Jr., 2008, Scratch Programming for Teens, Course Technology, 3-51
- 11) Wanda P. Dann, Stephen Cooper, Randy Pausch, 2011, Learning to Program with Alice 3rd ed., Prentice Hall, 3-19
- 12) 福村好美・湯川高志・五百部敦志, 2007, Squeak プログラミング簡単に作れるビジュアル教材, 東京電機大学出版局, 15-35, 95-111