

線形代数学の計算ドリルサイトの目的と実装

大阪産業大学・経済学部 服部 純典 (Yoshinori Hattori)
Faculty of Economics,
Osaka Sangyo University

1 はじめに

大学初年度に配当される数学の講義科目において近年、従前の講義形式の授業では受講学生の十分な学習到達度が得られなくなっている。これは大学の入学年齢人口が減少していることにもまして、大学への進学率が増加していることで入学生の学力レベルがかなり低下していることに起因している。さらに、大学の入学選抜方法の多様化により、入学生の学力レベルの幅が広がることでその現象を助長している。そのため授業担当者はいろいろな方策を講じて学習効果の改善に努力をしている。微積分学や線形代数学といった数学の講義科目は自然科学系の専門科目は言うに及ばず経済学をはじめとする人文・社会科学系の専門科目を学習する上での基礎となっていることもあり、その習得は欠かせないものである。特に線形代数学は新しい概念や計算法を学習することが多く、履修学生に対し習得を困難にしている。

そこでその対策として、担当している授業で受講学生との対話を重視し、理解度を把握しつつ授業の進度を調整するようにした。また、各回の授業の後半にその時間の講義内容を実際に活用する問題演習を課して、受講学生が授業内容を即時に演習して復習できるようにした。この問題演習は履修学生にとって自分の理解度を確認し、その理解を深化するものであり、また講義者にとっては履修学生全体の理解度を知ることができ、講義内容の見直しをするよい機会となる。Benjamin S. Bloom [1] による完全習得学習の理論によれば、この問題演習は診断的評価、形成的評価、総括的評価の3つの評価のうち形成的評価にあたることになる。

しかし、前に述べたように学生の学力レベルの幅が許容外に広がっていることにより、演習問題をすぐに解答し終える学生から講義時間が終了してもほとんど手のついていない学生まで様々となってきた。また、学生によって解答に必要な時間の違いがあるだけでなく、理解に必要な問題数が異なってくることもあり、基礎学力の養成という観点から学習結果を向上させるという課題を解決するために対策を講じる必要に迫られた。

2 ウェブサイト構築の背景

近年ネットワーク環境の目覚ましい発達により、以前では想像すらできないような容易さでインターネットに接続できるようになり、まさにユビキタスなネットワーク環境

が学生個々に提供されている現状となっている。特にスマートフォンをはじめとする携帯情報端末の急速な普及で、学生はほぼ全員がインターネットに常時接続できる機器を所有携帯しているのが現状である。たとえ所有していない者でも大学にはインターネットにつながったコンピュータを使用できる環境があり、その環境は学生全員に提供されている。この現状の環境を利用してウェブサーバ上のウェブサイトで演習課題が提供されるものを構築すれば、今までになかった演習形態を提供することになり、「いつでもどこでも」活用できる計算演習が可能になる。知覚心理学者 James J. Gibson [2] が提唱した造語であるアフォーダンスを用いて表現するなら、携帯情報端末はインターネットに接続して情報を取得するというアフォーダンスが存在するので、その携帯情報端末で計算の演習をするというアフォーダンスを確立することを目的としていると言える。

ところでインターネット上でこのような演習を提供するウェブサイトは、現状ほとんど見当たらない。確かに一見似たようなウェブサイトは存在するが、その考え方は自習を目的としている eラーニング型のものである。例えば大阪府立大学高等教育推進機構 [3] の「線形代数計算演習」というウェブサイトがあげられる。今回の目的は授業の補完関係となるもので、向上心のある学生を対象とした eラーニング型ウェブサイトと異なり、受講学生全員の活用する簡潔なウェブサイトを提供することである。また、提供される問題の質と量という面から考えると、現在あるウェブサイトは限られた問題数のデータベースからの出題となっていて、繰り返し利用での学習効果の向上に疑問が残る。問題量という点では、書籍で鶴丸孝司ほか [4] による問題集があるが、問題の質という点で精選不足と思われる点があり、迂闊な誤りも散見され初学者の利用は薦められない。このような点を解消するものを目指してウェブサイトを開発することを発想するに至った。

3 開発の構想

3.1 開発の基本的概念

前述のような現状を踏まえ、開発するウェブサイトの基本概念を新たに学習する計算法の鍛錬に特化することにした。数学という科目の特性上、概念の理解と同時にそれを実現する計算法の体得が重要となる。これら2つは相補的な関係にもあり、学習者は形成的評価によって習熟度を自覚し、より確実に理論の理解に近づくことができる。また、線形代数学では、微積分学に比べて新しい計算法が現れる。その計算法のなかで最も重要なものが基本変形であり、線形代数学の基礎となっている。基本変形による計算のうち算出するものの性質が異なるためその違いを把握することが肝要となるのが「階数の計算」と「行列式の計算」である。つまり、「階数の計算」では行列（または連立方程式）の状態を表す指標を求めるのに対し、「行列式の計算」ではまさに値（数値）を求めることになる。受講学生には少なからずこの2つの算法を混同する者が見受けられる。したがって、今回は正方行列に関する階数と行列式の計算に絞って問題演習をするウェブサイトを開発するものとした。

3.2 提示される問題の量と質

学習を効果的にするには、ウェブサイトへ接続するごとに新たな問題が提示されることが望ましい。特に基本的計算力を養成するためには毎回新しい問題に取り組むことが効果的であり、そのためには問題を自動生成することが求められる。もちろんデータベースとして蓄積された問題群からランダムに取り出して出題する方法も考えられる。しかし、よりよい演習を実現するには重複出題をできるだけ避けた方がよく、そのためには蓄積する問題数を多くしておくことが必要となる。その場合データベースファイルからの読み出しが多数発生することで処理の遅延を招くことになりシステムの運営上好ましくない。

ところで、乱数を用いて問題を自動生成するとほぼ正則行列しか作られないので、作想的に特異行列を生成するようしなければよい演習にはならない。したがって、実装には特異行列発生率を事前に設定しておき、正則行列か特異行列かを乱数で決定することにした。さらに、特異行列の場合、階数2以上の特異行列を各階数が等確率に生成するようにした。

行列の成分については、基本変形の習得が目的であり、不必要に成分計算を難しくすることは目的に合わないので、行列の成分は整数とし、その絶対値が20以下のものに限定することとした。この設定により演習の焦点がはっきりすることになる。また、行列も正方行列のみとし、サイズは演習として意味のある3次から5次のものとした。2次の行列では両方の計算法ともあまりに容易すぎて基本変形を用いずとも解答できるので不採用とした。3次の場合は初歩の練習として必要であると判断したが、行列式の計算ではSarrusの方法を用いることができる。行列式の計算は本来基本変形をしたあと余因子展開をして次数を下げて計算する方法を学習してほしいので、Sarrusの方法が適用できず余因子展開を用いないとうまく計算できない4次以上の演習が必要となってくる。もちろん階数の計算においても4次以上になれば特異行列の階数も複数の場合があり、よい演習となる。

行列式の計算では行列式の次数が上がるにしたがって値が大きくなる傾向にある。3桁以上の整数の四則計算は時間がかかり、本ウェブサイトの趣旨から外れることになる。そこで、値が大きくなった場合は生成された行列式から成分を調整して行列式の値を小さくし、許容できる範囲の値になるようにすることが必要となる。行列式の計算での余因子展開の利用を学習させるには次数が4次以上が必要であるが、値が大きくなる傾向にあるので、最初ランダムに生成する成分の絶対値を小さくしておく調整処理の発生を抑えることが期待できる。それにより平均処理時間を短くすることができる。よって4次以上の場合、成分は絶対値が9以下の整数で生成することとした。

3.3 学習の利便性の改善

計算演習のウェブサイトインターネット上で提供すると、今日のネット接続環境の充実により、ほぼどこからでも閲覧可能となる。授業中の教室で演習させることもできるし、帰宅して復習で演習することもでき、さらには極端な話ではあるが、移動中の車内でも勉強ができるのである。しかし、それを可能にするためには、多くの機器からの

接続に対して閲覧可能なウェブページの記述形式をとる必要がある。つまり、javascriptなどを用いて必要以上にクライアントサイドでの処理を要求せず、多くのウェブブラウザで表示可能にすることが利便性を向上させることになる。そうすることで、「いつでもどこでも」演習可能なウェブサイトが実現でき、いわば「サイバー計算ドリル」と言えるものが構築されることになる。つまり、あたかも小学校の算数で利用される計算ドリルのウェブ版と見なせるものである。この活用により学習者の能力差から生じる習得に必要な学習時間の差異の問題が緩和される。

さらに、ウェブ版という特徴を活かすために学習の効果を上げる機能を付けることを着想した。普段の授業中の演習では提出してきた答案をその場で添削し、誤答の場合は適切な指摘をして再挑戦させることで学生の理解度を改善することができる。そのことから発想すると解答を誤ったときにその場で正しい方法を参照できる機能を提供すればよい指導効果が期待できる。そこで解答に対する採点を表示したページに解答解説を見るためのボタンを配置し、解説のページへのリンクをはることにした。

解説において示すべき計算方法はいろいろな手順が考えられるが、このウェブサイトの目的を受講者全員の学力の最低線の引き上げと位置付けていることから、基本変形が理解できていない初学者に対する指導として相応しい手順を示すことにした。その計算手順は第1行第1列の成分から順に計算していく基本的なものである。もちろん個々の問題の特性を利用して効率的に計算することもできるがその工夫は基本的な計算法を習得した後に学習すべきものなので、本ウェブサイトの目的から外れると判断し採用しないこととした。さらに表示方法はその計算手順を理解しやすいような提示の仕方にするのが重要であると考えた。そこでページを進めるごとに1段階ずつ表示していく方式をとることとした。この仕掛けは筆記計算の基本的な手順に沿ったものを丁寧に計算するよう指導することを目指したものである。

以上のような機能を備えることで計算ドリルと参考書の役割を併せ持つことができ、携帯情報端末のコンパクトなポータビリティと相まって今までにない学習の利便性を獲得できるようになる。まさにこれは「サイバーハイパー計算ドリル」と言う標語で例えることができるものである。

4 ウェブサイトの実装

4.1 記述形式の選定

多くの機器で表示可能なウェブサイトを目指すことから、現在の状況を鑑みてウェブページの記述はHTML 4.01 Strictを採用し、javascriptなどクライアントサイドの処理を要求することは端末機種限定をするものなので使用しないこととした。サーバー・クライアント間のやり取りはHTTPを使うものとし、サーバーサイドでの処理はCGI技術を用いる。プログラミング言語はCGIプログラムでよく用いられる言語のPerlを用いることとした。開発途上のウェブサイトなのでスクリプト言語が適当であると判断した。HTMLではフォーム機能でサーバーへのデータ送信を設定することになるが、ページ間で継承するデータはクッキーを用いず、フォーム機能の隠しコントロールで埋め込

む。これもクッキーを受け取らないウェブブラウザを排除しないようにする配慮である。継承するデータは演習の種類（階数、行列式）と行列のサイズ（3次、4次、5次）が必須であり、さらに出題される行列の成分も盛り込む必要がある。これは問題を自動生成させることから正誤の判断や解法解説に必要となってくるのが理由である。これらのデータをサーバーマシンにファイルとして保持する方法も考えられるが、サーバーマシンでの処理でファイルの読み書きが多数回発生し、処理の遅延の原因になるので、それを避けるため採用していない。

送信データの記述には1バイトのASCII文字のみで記述することにより送信データのバイト数を少なくした。このことでCGIプログラムでの処理時におけるパーセントデコーディングを省略することもできる。その意味ではASCII文字であっても英数字以外の多くの記号ではパーセントデコーディングが必要になり、またウェブブラウザによっても送信時にパーセントエンコーディングされるか否かは差異がある。よって、記述データは正確には1バイトの英数字と符号を表すマイナス記号のみで構成することとした。継承される基本データについてinput要素での属性名とその指し示す意味との関係は次のようになる。

演習の種類	name	value	行列のサイズ	name	value
階数	t	1	3 × 3	d	3x3
行列式	t	2	4 × 4	d	4x4
			5 × 5	d	5x5

4.2 数値データの実装

数学での計算は正確な数値を扱うため、通常コンピュータでの数値計算で使われる浮動小数点形式の数値表現は用いられない。今回は有理数の範囲で計算することとなるので、すべての数値データを有理数（分数）表現で取り扱う。実際には符号なし固定小数点の不定長データで表した2つの数値データを1組として扱い、

(符号)(分子の数値データ) v (分母の数値データ)

というように英文字「v」で連結して表す。数値の符号は負の場合のみ先頭に「-」を置く。例えば、 $-\frac{2}{3}$ は「-2v3」となる。もちろんこの表現では同じ値を表すのに自由度があるので、既約分数で表すことにして表現の一意性を実現している。ゼロは「0v1」と表し、零除などのエラーが出た場合には「1v0」などで表現する。このエラー表現は分子の部分を変えることでエラーの種類を表現することも可能となる。これを数値データの標準表現とし、四則計算が終わるごとにデータを標準表現に変換して保持する。

また、行列成分のデータをウェブページのHTMLコードに埋め込むため、その表現はinput要素のhiddenタイプでname属性を行番号と列番号を英字「x」で連結した文字列で設定し、その成分の値をvalue属性に記述することとする。例えば、第1行第3列成分が $-\frac{1}{2}$ ならばHTMLコードは次のようになる。

```
<input type="hidden" name="1x3" value="-1v2">
```

さらに、行列式の計算の途中ではくくり出した数値も継承しなければならないので、そのときは `name` 属性を「`m`」にして数値データを `value` 属性に記述することとした。

4.3 計算式の実装

前節で定義した有理数データをそのまま計算する処理を実装しなければならないので、式を表す文字列を解釈して計算する内部処理を設定した。式の表現はポーランド記法を採用した。これは式の表現に括弧を使用しなくて済み、処理は右から1パスで処理していけるからである。計算式は数値データと演算子 (+, -, *, /) に1バイトの空白文字の分離記号を挟んで文字列を構成することとした。この計算式文字列はサーバーサイドでのみ使用するので演算子を表す文字が HTTP 伝送で変換されるか否かは考慮しなくてもよい。式表現の例をあげると、通常の数式表現から次のような表記に変換される。

$$2 \times 4 + \frac{2}{3} \times 6 \quad \Rightarrow \quad + * 2v1 \ 4v1 * 2v3 \ 6v1$$

4.4 解法解説の実装

本ウェブサイトの一つの重要なポイントである解法解説の機能であるが、それは学習者が誤謬を認識して訂正することを支援するもので、学習者が着実に基本的な計算手順を理解し身に着けられるようなものが望ましい。それを実現するために表示方法は1ページに1手順のみを示し、確実に1段階ずつの計算を理解していくように配慮した。さらに、変形の仕方を式表現だけでなく説明文を挿入することで理解の助けとした。1手順ごとにページを更新するためネットワークの通信が発生するが、ネットワークトラフィックの増加より教育効果の向上が優先されると考えた。

解法手順の方針は、初学者がまず身に着けるべき手順を示すことにした。階数の計算では、第1列から順に階段行列を形成する基本的な計算手順をふむ。実際の処理では表示を1段階ずつ示すため、基本変形の処理が決定したら中断して出力処理をすることとなる。それを具体的に表すと次のようになる。なお行列の第 i 行第 j 列成分を a_{ij} と表記する。

- [0] i, j をともに1とする。
- [1] $a_{ij} = 0$ なら $a_{kj} \neq 0, k > i$ となる成分 a_{kj} があれば、第 i 行と第 k 行を入れ替えて、出力処理に行く。なければ、 j を1増やして [5] に行く。
- [2] 第 k 行の成分全体で共通因数があれば、その行全体を共通因数で割って、出力処理に行く。
- [3] $|a_{ij}| > |a_{kj}| > 0, k > i$ となる成分 a_{kj} があれば、第 i 行と第 k 行を入れ替えて、出力処理に行く。
- [4] $|a_{kj}| > 0, k > i$ となる成分 a_{kj} があれば、第 k 行から第 i 行を a_{kj}/a_{ij} 倍したものを引いて、出力処理に行く。なければ、 i と j をそれぞれ1増やす。
- [5] i, j がともに行列の添字の範囲であれば、[1]に戻る。

[6] 0でない成分がある行の行数を階数として、出力処理に行く。

ここで、行についての共通因数とはある行のすべての非零成分の(分子の最大公約数)/(分母の最小公倍数)のことでこれが1以外の場合を指す。

行列式の計算でも第1列から順に処理をする。ここでは、原則として第1列で余因子展開するように基本変形を選択することになる。計算の途中で出てくる行列式の係数を m で表す。

- [0] m を1とする。
- [1] 行列式の次数が2であれば、成分のたすき掛けで計算した値に m を掛けたものを行列式の値として、出力処理に行く。
- [2] ある行の成分がすべて0か、ある列の成分がすべて0ならば行列式の値を0として、出力処理に行く。
- [3] ある行の成分に共通因数があるか、ある列の成分に共通因数があれば、くくり出す。つまり、その行または列の成分をすべて共通因数で割り、 m に共通因数をかけて、出力処理に行く。
- [4] 第1列の非零成分の絶対値の最小のものが第1行第1列の成分でなければ、それを満たすように行の入れ替えをして、 m の符号をかえて、出力処理に行く。
- [5] $|a_{k1}| > 0, k > 1$ となる成分 a_{k1} があれば、第 k 行から a_{k1}/a_{11} 倍した第1行を引き、出力処理に行く。
- [6] 第1列で余因子展開する。 m を a_{11} 倍し、行列式の成分を a_{11} の余因子に置き換えて、出力処理に行く。

この手順内でも行または列についての共通因数は階数の計算の場合に準じる。

これらの基本変形の処理手順は手計算で行うべき基本的手順を実現している。なお、改良が必要な場合には処理プログラムを改変することで容易に対応できる。これも自作プログラムの利点である。

5 今後の開発点

今回はこのウェブサイトの中心となるところを開発したが、まだ開発余地がある。次にその予定をあげておく。

- 演習範囲の拡大する。当面は内積のっていない線形空間について線形代数学における計算練習を包括的に学習できるようにする。具体的には掃き出し法、固有値問題などを順次取り込む。
- 学習支援の付加機能を装備して、学習履歴を記録する。そのことにより、授業支援の機能向上を図り、学習者の学習意欲を高めることを狙う。
- 多言語化して国際的利用を目指す。対応可能な英語、フランス語、ロシア語、ドイツ語、スペイン語、中国語を予定している。

なお、現在このウェブサイトは既存のサーバーマシン

<http://www.eco.osaka-sandai.ac.jp/mth/>

にて公開しているが、将来は独立のマシンで運営を希望している。

参考文献

- [1] Benjamin S. Bloom, "Learning for Mastery", *UCLA-CSEIP-Evaluation Comment*, 1[2], 1-16, 1968.
- [2] James J. Gibson, *The Ecological Approach to Visual Perception*, Boston: Houghton-Mifflin, 1979.
- [3] 大阪府立大学高等教育推進機構, "線形代数計算演習", webMathematica で学ぶ大学数学, <http://webmath.las.osakafu-u.ac.jp/top/std/stdmon01.jsp?BCD=020000>, (参照 2015-08-31).
- [4] 鶴丸孝司ほか, 『行列と行列式 辞書式配列 1800 問』, 内田老鶴圃, 1992 年.