4ZF-1

計算量が制御された連立方程式の問題及び模範解答の自動生成

森　優貴
法政大学 情報科学部 ディジタルメディア学科

1　序論

本研究では, 未知数の数, 方程式の数, 解の次元, 係数の整数部・虚部絶対値の最大値, 同次・非同次, 式変形のステップ数の最大値, 解答速度が指定されたときに, 条件を満たす連立方程式の問題と, 解に至る計算過程を含む模範解答を自動生成し, また標準解答時間の見積もりも出力するシステムを開発した.

先行研究である橋本 [3] のプログラム「連立方程式の自動生成」では, 乱数によって生成した係数行列の要素の絶対値の最大値と, その係数行列の行列式の値の絶対値を指定された値以下とすることで計算量を制御していた. しかし, 係数行列の要素の値及び, その行列式の値の制限を行うだけでは計算過程の係数が制御できず, また, 解に辿り着くまでのステップ数も制御されない. それゆえに, 解答時間の計測は問題作成者が実際に生成した問題を手で解いて行う必要が生じ, 計算量のコントロールを自動的に行うという目的が達成されているとは言えない. また, 解の次元の入力は行わせるが, 生成する問題ではそれをほとんど無視している, 模範解答は解のみが表示され, 計算過程が示されていないという問題点もあった. そこで, 先行研究の改良として

1. 計算過程での係数の制御
2. 標準解答時間の定義
3. 解の次元の反映
4. 計算過程を含む模範解答の生成

を行う. また, Web ブラウザ上の GUI インターフェースを作成し, ユーザの利便性を高める. システム全体の最終的な完成図は図 1 のようになっている.

本研究の手法は, 線形代数の他の題材や, 代数, 解析などの他の分野にも適用可能である. GUI を利用することによって, システムの基幹部分と UI を分離して操作ミスによる事故を防ぎ, 併せてユーザの利便性を高めたが, 本論文では, 基幹部分の説明に焦点を絞る.

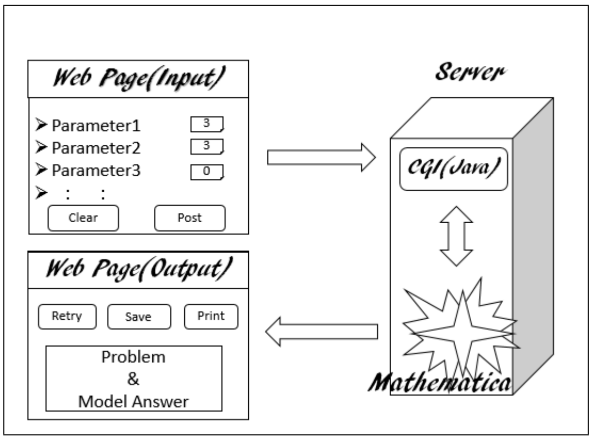

図 1　システムの BigPicture

2　計算量の制御された問題の生成

数学の問題を解く上で, 計算過程に出現する有理数演算の高さ (分子および分母, または虚部の整数絶対値の最大値) は計算量と大きく関係がある. 橋本 [3] は, 生成する連立方程式の係数の大きさを制限することで計算量を制御していたが, 計算過程で出現する数値も考慮する必要がある. よって今回, 問題を生成するにあたって, 計算過程で出現する数値の制限に「行逆変形」という独自の手法を用いた. 行基本変形は拡大係数行列の係数行列部を被約階段行列に変形し, 複雑な係数行列を単純化し解を求める操作であるが, 「行逆変形」はその逆の操作で, 入力パラメータの条件を満たす解をあらかじめ生成しておき, その解を基本変形の逆操作で複雑にし 問題を生成する. また, 逆変形の操作の過程で計算量を制御することで, 生成された問題を行基本変形によって実際に解く際も計算量が制御される. 拡大係数行列を S とし, その p 行 q 列の要素を S_{pq} と表記する. S_{pq} に S_{iq} の k 倍を加える逆変形を行った場合, 新しい方程式が現れる. $k = a$(a は整数) について, 逆変形後に現れるその方程式の係数の絶対値は, 整数係数の絶対最大値が $N_{\max}$ で与えられたとき式 (1) のように制限した.

$$|S_{iq} + k \times S_{iq}| \leq N_{\max} \tag{1}$$

Automatic Generation of Problems in Linear Algebra and Sample Solutions with a Specified Computational Complexity
Yuki Mori
Faculty of Computer and Information Sciences, Hosei University

k について解くと式 (2) となり, 列ごとにおける k が取りうる範囲の共通部分の範囲で乱数を生成し逆変形を行う. つまり, 係数の大きさが指定された数値を超えない範囲で, S_{pq} に k 倍した S_{iq} を加える操作である. 0 は何倍して加えても変形前の値と変わらないため, $S_{iq}=0$ のとき絶対最大値は $(2\times N_{\max}+1)$ で与えた.

$$\begin{cases} \dfrac{-N_{\max}-S_{pq}}{S_{iq}} \le k \le \dfrac{N_{\max}-S_{pq}}{S_{iq}} & (S_{iq}\neq 0) \\ \\ -(2\times N_{\max}+1) \le k \le (2\times N_{\max}+1) & (S_{iq}=0) \end{cases} \tag{2}$$

また, $k=a+bi(a,b$ は共に整数$)$ のとき, $S_{pq}=x+yi$, $S_{iq}=v+wi$ とすると, 虚部の絶対最大値が $I_{\max}$ と与えられたとき, 式 (3) のように整数部・虚部について制御を行う.

$$\begin{cases} |x+av-bw| \le N_{\max} & (\mathit{Real\ part}) \\ \\ |y+bv+aw| \le I_{\max} & (\mathit{Imaginary\ part}) \end{cases} \tag{3}$$

3 計算量の時間変換

一般的に, 連立方程式の計算量は未知数の数の 3 乗に比例するが, 解の構造やステップ数, 有理数演算の高さ (分子と分母の絶対値の最大値) による計算の複雑さも考慮したい. 「行逆変形」によって計算過程での値の制限を行ったことで, 計算過程で現れる有理数の絶対値の期待値は $(N_{\max}+I_{\max})/2$ であり, またその桁数の期待値を d 桁とおく. 行基本変形自体の計算量は未知数の数 n, 数値の桁数 d, 式変形のステップ数 s がそれぞれ既知のとき, 数字の乗除加減算に分解できる. 数値の桁数を d とすると乗算の計算量は $O(d^2)$ で, 加減算は $O(d)$ である. また除算は乗算と加減算の組み合わせの繰り返しであり, 発生する除算は逆変形の特性上すべて整除可能なため, 繰り返しは高々 d 回で $O((d^2+d)\times d)=O(d^3)$ である.

行基本変形の計算には乗算と加減算 ($S_{pq}+k\times S_{iq}$ の計算), 除算 (k を求める計算) の 2 つがあり, それぞれを合わせ計算時間 F は式 (4) のように定めた. 式 (f_1) に示すように乗算と加減算は列要素の個数 $(n+1)$ とステップ数 s に比例する. 式 (f_2) に示すように除算はステップ数 s に比例する.

$$F_{(n,s,d)} \doteq \begin{cases} (n+1)\cdot s\times(2d^2+d) & ...(f_1) \\ \qquad + & \\ s\times(2d^2+d)\cdot d & ...(f_2) \end{cases} \tag{4}$$

また, 未知数の数や解の構造によって解答の記述時間が決まる. 解答として記述する項目は計算過程の拡大係数行列, 式変形の説明, 最終的な解の 3 つであり, それぞれを合わせ解答の記述時間 G を式 (5) のように定めた. 方程式の数を m, 解の次元を D_{sol} とすると, 式 (g_1) に示すように計算過程の記述時間は拡大係数行列の列数 $(n+1)$, 行数 m, ステップ数 s に比例する. また式 (g_2) に示すように式変形の説明の記述時間はステップ数 s に比例する. 式 (g_3) に示すように最終的な解の記述時間は $(2+D_{\mathrm{sol}})$ と未知数の数 n に比例する.

$$G_{(n,m,s,d,D_{\mathrm{sol}})} \doteq \begin{cases} (n+1)\cdot m\cdot s\times d & ...(g_1) \\ \qquad + & \\ s\times(10+d) & ...(g_2) \\ \qquad + & \\ (2+D_{\mathrm{sol}})\cdot n\times d & ...(g_3) \end{cases} \tag{5}$$

(4) 計算時間, (5) 記述時間を合わせ, d 桁の数字を加減算または記述する時間 t(秒) を掛けて標準解答時間 $T(n,m,s,d,D_{\mathrm{sol}},t)$ として式 (6) と定義する.

$$T_{(n,m,s,d,D_{\mathrm{sol}},t)} = (F+G)\times t_{(\text{秒})} \tag{6}$$

4 結論

本研究では, 数式処理システムを利用して, 橋本 [3] のプログラム「連立方程式の自動生成」の大幅な改良を行い, システム全体を再構築した. 計算量のコントロールとして, 計算過程の有理数演算の高さの制御を行い, 式変形に要するステップ数と解の構造から標準解答時間を設定した. Web を利用することでより多くの人にシステムを提供することができ, ブラウザ上の GUI インターフェースで入出力を行うことで, いっそう問題作成者の手間を省くことができる.

今回, 問題の自動生成に「行逆変形」という独自の手法を用いたが, 解の構造から計算過程の逆を辿り, 問題を生成するこの考えは, 連立方程式のみならず他多くの代数学や解析学の題材の問題の自動生成に応用できる.

参考文献

[1] 小林道正著, Mathematica による線形代数, 朝倉書店, 東京, 1996.

[2] 長谷川浩司著, 線型代数-Linear Algebra, 日本評論社, 東京, 2004.

[3] 橋本龍太, 計算量と解の次元を指定した線形代数の問題の自動生成, 法政大学卒業論文, 東京, 2012.

[4] Wolfram Mathematica 9 Documentation, http://reference.wolfram.com/mathematica/guide/Mathematica.html, 2013.