数学オンラインテスト問題標準仕様 MeLQS のための基本アルゴリズムの選定

谷口哲也[*1]・中村泰之[*2]・川添充[*3]・吉冨健太郎[*3]・福井哲夫[*4]・白井詩沙香[*5]・加藤克也[*6]・中原敬広[*7]
Email: taniguchi.tetsuya@nihon-u.ac.jp

*1：日本大学医学部　*2：名古屋大学情報学研究科　*3：大阪府立高等教育推進機構　*4：武庫川女子大学生活環境学部
*5：大阪大学サイバーメディアセンター　*6：サイバネットシステム　*7：三玄舎

◎Key Words　　コンテンツ共有

1. はじめに

数学のオンラインテストにおいて，数式で入力された解答を自動採点し適切なフィードバックの提示を行う，数学 e ラーニングシステムが国内外で広まっている。STACK や Möbius Assessment などはその例であるが，個々のシステム毎に問題を実装する必要があるため，異種システム間での問題の相互利用は困難であり，重要な教育資源が互換性なく分散している。我々は，問題コンテンツの標準仕様を策定しその仕様に基づいて問題を作成できれば，異種システム間での問題共有が可能となり数学 e ラーニングの活用が促進されると考え，数学オンラインテスト問題の標準仕様 MeLQS を提案した。MeLQS は問題のアイデアを共有するための設計図としての「問題仕様」とシステムへの実装に必要なアルゴリズムやシステムに要求される情報を記述する「実装仕様」の二段階の構成とした。本発表では実装仕様で必要となるアルゴリズムについて紹介する。

2. MeLQS とは

2.1 数学 e ラーニングの問題共有について

日本で利用されている主な数学 e ラーニングシステムは STACK, MATH ON WEB, Möbius Assessment であるが，表 1 にまとめられているように，それぞれ利用している数式処理システム（Computer Algebra System, CAS）が異なり問題の実装方法も異なっている。したがって，同類の問題であってもそれぞれの実装方法に従って問題データを作成する必要があり，一部に問題の変換を行うことにより相互利用を実現できている例もあるが，制限もあり，重要な教育資源としての問題データが互換性のないまま分散している状況にあると言わざるを得ない。Mathbank[3] など特定の数学 e ラーニングシステムでの問題データ共有の仕組みはあるが，現在運用されている主要システム同士の相違を吸収し，異種システム間での相互問題データ共有を実現していくことは，数学 e ラーニングシステムの普及，ひいては，近年注目されている数理・データサイエンス教育に求められる数理リテラシー定着のためのオンライン教育を充実させていくことにつながると期待される。

表 1 主な数学 e ラーニング

	CAS	特徴
STACK	Maxima	柔軟なフィードバック
Math on Web	Mathematca	柔軟な評価アルゴリズム
Möbius Assessment	Maple	柔軟な解答方式

2.2 問題仕様書と実装仕様書について

吉冨による STACK と MATH ON WEB の問題データ構造精査という先駆的な研究により[4]，数学 e ラーニングシステムの問題データの共通性が明らかになり，異種システム間での相互問題データ共有を実現するためには，標準仕様を設定し，それに基づいた問題設計をすることが重要であるという知見を得た。その知見を踏まえて，我々は数学オンラインテスト問題の標準仕様（Mathematics e-Learning Question Specification, MeLQS）を提案した[5]。提案仕様は問題仕様と実装仕様の二段階で構成されている。問題仕様は，数学教員が問題のアイデアを共有するための設計図であり，数学教員が数学教育全般において広く教育資源(ペーパーテスト等)として活用されることも期待される。実装仕様は，問題を数学オンラインテストとして利用するために必要なアルゴリズムやシステムに要求される情報を記述するものであり，オンラインテスト作成の補助を目的としたものである。問題仕様と実装仕様の蓄積はそれぞれ教育に携わる教員と実装に携わる者との協働により行われることを想定している。

3. 問題仕様書

3.1 問題仕様書とは

問題仕様書は数学教員が一目でわかるような仕様書で，MeLQS 上で編集可能である。

3.2 問題仕様書作成例

ガンマ関数の計算問題の問題仕様書の例を図 1 に示す。ガンマ関数の値 $\Gamma(6)$ を学生に計算させたいが，できれば 6 の箇所をいろいろランダムに変化させた場合のガンマ関数の値の計算をさせたい旨が備考の箇所に記述さ

れている。また， あまり大きな自然数 n に対して，Γ(n) の値を計算させるのは大変なので， n は 3 から 6 の間で変化させたいという数学教員の要望が記述されている。このように， 問題仕様書は自由記述な形式となっており，数学教員の作成したい問題の内容や意図がほぼそのまま表現できる。

次章で述べる実装仕様書を作成するときも，この問題仕様書を参考にすれば，数学教員がシステム側に寄り添った実装仕様書を容易に作成できるであろうと我々は期待している。

図 1　 ガンマ関数の計算問題の問題仕様書

4.　実装仕様書

4.1　実装仕様書とは

実装仕様書は実際の各数学 e ラーニングシステムに実装するために必要な問題の形式やアルゴリズムが含まれており， Blockly を用いて可視化されている。また，MeLQS 上で編集可能である。その実装仕様書を参考にしながら数学教員が実際の各数学 e ラーニングシステムに容易に問題を実装していくことを期待している。

4.2　Blockly をもちいた実装仕様書作成例

図 2 が， 図 1 の問題仕様書をもとに作成した， 実装仕様書である。各数学 e ラーニングシステムに実装しやすいように可視化されている。図 1 の備考における n の範囲を反映させるために問題生成処理の箇所で，n の範囲を 3 から 10 に指定している。図 2 のフィードバックと採点の箇所で ans1 に代入された学生の回答が正解である(n-1)!に等しければ「よくできました.」と学生に提示し，そうでなければ「違います.」が提示されるよう制御している。

また， 図 2 のフィードバックと採点の箇所で正解であれば，変数 mark に 1 が代入され 100％ と採点され，不正解であれば，変数 mark に 0 が代入され 0％と採点されるようになっている。ブロックを用いて表示されていることによって，より実際の数学 e ラーニングシステムに寄り添った仕様書であることがわかる。

図 2 ガンマ関数の計算問題の実装仕様書

5.　重要なブロックのリスト

参考文献 (1)， (2) の線形代数の問題の実装仕様書を作成するとき， 特殊な行列生成や行列のランクを計算するための複雑に絡みあったブロック群が頻出する問題が発生した。例えば 3 点を通る平面の方程式の問題の作成にも，正解の$ax+by+cz=d$に対応するベクトル(a,b,c)と学生が入力した平面の方程式$Ax+By+Cz=D$に対応するベクトル(A,B,C)を並べてできる 2×3行列の階数（ランク)が 1 と等しいと判定する場面がでてくるので，階数の計算が必要である。複雑なブロックを組み合わせることにより階数の計算が実現可能であるが， それらをまとめて新たなる一つのブロックと定めた方が便利である。そこで，我々は，必要となる数学的処理を実現するための複雑なブロックからなる群を一つにまとめたものを新たなブロックとして慎重に策定した。以下にそれらの新たに策定されたブロックのリストを列挙する。そのリストには被約階段行列の生成や階数の計算，行列の和やスカラー倍を計算するブロック等が含まれる。これらのブロックを用いれば線形代数の問題仕様書は容易に作成されると期待している。

	ブロックリスト
1	[階数指定行列] (m, n, r, am) 入力:行数m,列数n,階数r, 成分の絶対値の上限am 出力:階数rのK成分mxn行列 補足:複素数が必要な場合は/Z[i] 生成方法:被約階段行列の左からGLn(Z)またはGLn(Z[i])の要素Mをかける．Mの要素の最大値amは制御するが, 結果の成分最大値は直接制御しない.
2	[行列式指定行列] (n, d, am) 入力:正方行列の次数n(>1), 自然数am, 整数d 出力:成分の最大値がam以下であるような行列式dのn次正方行列A∈Mn(Z) 実装:(1) am以下の乱数で行列式がdになるまでループを回す. (2) 上半三角と下半三角で行列式dの行列を生成, SLn(Z)の要素で成分の最大値条件を満たすまで積をとる.
3	[被約階段行列] (n, m, piv, am) 入力: n,m,piv({1,...,n}のsublist),am 出力: ピボットがpivであるようなmxnの行列Aで成分の絶対値がam以下の被約階段行列R 実装: 指定されたpivot を持つ被約階段行列をam以下の乱数を使って生成
4	[行列](m,n,am) 入力:自然数n,m,am 出力:成分の絶対値がam以下のｍｘｎ行列
5	[対角行列] (dL) 入力:整数リストdL 出力:対角成分がdLであるような対角行列
6	[基本行列P] (n, I, j, c) **入力**:**行列サイズ**n, (I, j) **成分が** c(**有理数**) **出力**:n**次基本行列**Pn(I, j;c)
7	[基本行列Q] (n, I, c) **入力**:**行列サイズ**n, **対角位置**I, **成分**c 出力:n次基本行列Qn(I;c)
8	[基本行列R] (n, I, j) **入力**:**行列サイズ**n, (I, j) **入れかえ** 出力:n次基本行列Rn(I,j)
9	[列指定置換行列] (n, p) 入力:自然数n,{1,…,n}の順列p 出力:順列pに対応する置換行列R(列の置換に対応する) 補足:転置をとれば[行指定置換行列]となる. *利用方法 [対角行列],[基本行列P],[列指定置換行列]から選んで作る. *不要もしくはランダムシャッフルでよいのは？ 実装上は行列式の問題などであった方がいいのでとりあえず残す
10	[正規直交系/R](n,m,am) 入力:自然数n(>1),自然数m,am 出力:R^nの正規直交系で各ベクトル成分の分母分子の絶対値がam以下のもの 利用:R=[正規直交系](n,n,am)の成分の1部をマスクして出題 実装:パラメータ化されたいくつかのパターンを用意する
11	[被約階段行列/Z] (n, m, piv, am) 入力: n,m,piv({1,...,n}のsublist),am 出力: ピボットがpivであるようなmxnの行列Aで成分の絶対値がam以下の被約階段行列R 実装: 指定されたpivot を持つ被約階段行列をam以下の乱数を使って生成
12	[被約指定行列] (R, am) 入力: 被約階段行列R,成分の絶対値の上限am 出力: 基本変形でRとなる成分の最大値がamである行列A 補足: 正則行列生成は時前で行う(最大値制御のため) 実装: am より小さい最大値で正則行列PをPRの成分の最大値がam以下になるまでループ生成する.
13	[行列の階数] (A) 入力:行列A 出力:Aの階数
14	[行列式] (A) 入力:正方行列A 出力:Aの行列式
15	[Jordan標準形生成]([[a1,n1],…,[am,nm]]) 入力:固有値a1,サイズn1のジョルダン細胞を[a1,n1]として, ジョルダン細胞のリスト 出力:与えられたジョルダン細胞の直和であるジョルダン標準形 *利用方法 対角化不可なもの=[Jordan標準形生成]([a,2],[b,1]) (a=b も含む), 対角化可能なもの=[Jordan標準形生成]([a,1],[b,1],[c,1]) (a=b,b=c,a=cも含む)として, 行列式1正則行列PでPJP^{-1}を利用

図 3 ブロックリスト 1~15

	ブロックリスト
16	[被約階段行列変形] (A, rc) 入力:行列A 出力:rc=行 or 列に関する基本変形により階段行列に変形する [ピボット取得](A) 入力:行に関する階段行列X 出力:Xの主成分列(ピボット)のリスト *利用方法 次元:rが正解,基底(1):B=[列階段行列変形](A)の0でない最初のr個の列ベクトル. (2):piv=[ピボット取得]([行階段行列変形](A)),Aのpiv[1],…,piv[r]列が正解例 補足:被約階段列変形は転置と行変形の組み合わせを使ってもらう(ブロックは用意しない)
17	[行列式](A) 入力:正方行列A 出力:行列式$\lvert A\rvert$
18	[内積指定直交化] (IP, vL) 入力:vLの要素に対する内積を計算する関数IP,ベクトルのリストvL 出力:内積IPによるvLの直交化(正規化しない)リスト
19	[内積指定正規直交化] (IP, vL) 入力:vLの要素に対する内積を計算する関数IP,ベクトルのリストvL 出力:内積IPによるvLの正規直交化リスト
20	[直交化] (vL) 入力:ベクトルのリストvL 出力:標準内積によるvLの直交化(正規化しない)リスト
21	[正規直交化] (vL) 入力:ベクトルのリストvL 出力:標準内積によるvLの正規直交化
22	[行結合] (aL) 入力:列数が同じ行列のリストaL 出力:aLの成分を縦に並べた行列(rowcat)
23	[列結合] (aL) 入力:同じ行数の行列のリストaL 出力:aLの成分を横に並べた行列(colcat) 実装:転置と行連結の組み合わせで列結合
24	[多項式のベクトルの係数ベクトル取り出し] (v, varlist) 入力:指定リストvarlist内の変数の多項式を成分とするベクトルv 出力:各varlistの要素の変数の係数からなるベクトルの列(係数ベクトル)のリスト *利用方法 解答が各変数の文字式と仮定する. 解答ベクトルをvとし, v=v0+pv1+qv2+…とする. [等価判定(行列)]をAv0とb, Av1,Av2,…と0 で行う. [等価判定(数値)]をp,q,..の数と n-r で行う. Q.これだと変数p, q,…の取り出しが必要では？ A.使用する変数は限定し, 範囲内の変数すべての係数を取り出すことで対応可
25	[外積ベクトル] (u, v) 入力:3次ベクトルu,v 出力:3次ベクトルw (uxv:外積) 実装:w1=u2v3-u3v2, etc. 利用:{a,b,c}=(u2-u1,u3-u1)の出力として a(x-u11)+b(y-u12)+c(z-u13)=0を展開した式 ax+by+cz=d が正解
26	[ベクトル非0定数倍判定] (u1, u2) 入力:n次ベクトル u1,u2 出力:u2=ku1 またはu1=ku2 (k<>0)ならTrue, それ以外はFalse 補足:u1,u2の一方が0のときはFalseとする. Rank利用+非0判定の組み合わせ 利用: u1={a,b,c,d},u2=ans1
27	[解空間] (A) 入力:行列A 出力:Ax=0の解空間の基底(ベクトルのリスト)
28	[直交行列判定] (A) 入力:行列A 出力:Aが直交行列ならば真,でなければ偽
29	[ユニタリ行列判定] (A) 入力:行列A 出力:Aがユニタリ行列ならば真,でなければ偽
30	[対角行列判定] (A) 入力:行列A 出力:Aが対角行列ならば真, でなければ偽
31	[行列の和] (A, B) 入力:行列A,B 出力:和A+Bが定義されるときはA+B,定義されないときはNull
32	[行列のスカラー倍] (A, c) 入力:行列A,スカラーc 出力:cA
33	[行列の積] (A, B) 入力:行列A,B 出力:積ABが定義されるときはAB, 定義されないときはNull 補足:定義されないときの戻り値を指定しておく方が後々便利かと思われる. ただし, Null にするかどうか要検討(和も同様)

図 4 ブロックリスト 16~33

参考までに，ブロックリスト 1～3 について，数学 e ラーニングシステム STACK による実装例を図 5 に示す。

	ブロックリスト	STACK実装

1	[階数指定行列] (n, n, r, am)	rnz(m):=(rand(m)+1)*(2*rand(2)-1);
		pv:sort(makelist(random_permutation(makelist(i,i,1,n))[i],i,1,r));
		re:genmatrix(lambda([i,j],if i<length(pv) and j=pv[i] then 1 else 0),m,n);
		/* rnz(2) の2を大きくすると成分が大きくすることができます */
		chkDet1(Ma,s):=block(for k:0 while (notequal(determinant(Ma),1)) do
		(Ma:genmatrix(lambda([i,j],rnz(am)),s,s)),return(Ma));
		Ma:genmatrix(lambda([i,j],rnz(am)),m,m);
		Mb:genmatrix(lambda([i,j],rnz(am)),n,n);
		Ma:chkDet1(Ma,m);Mb:chkDet1(Mb,n);
		matA:Ma.re.Mb;
		/* この実装では成分の上限amは保証されない，改訂の必要あり，動作はする，*/
		/* (Ma: .. から matA:Ma.re.Mb までを成分の上限が am におさまるまでループすると
		止まらない可能性あり，要検討 */

2	[行列式指定行列] (n, d, am)	rnz(m):=(rand(m)+1)*(2*rand(2)-1); chkDet1(Ma,s):=block(for k:0 while (notequal(determinant(Ma),d)) do (Ma:genmatrix(lambda([i,j],rnz(am)),s,s)),return(Ma)); Ma:genmatrix(lambda([i,j],rnz(am)),n,n); A:chkDet1(Ma,n);

3	[被約階段行列] (n, m, piv, am)	rnz(m):=(rand(m)+1)*(2*rand(2)-1);
		/* Return which column is first non-zero entry of i-th row vector */
		getFirstNZ(M,i):=block([t,p,q],[p,q]:matrix_size(M),if M[i]=makelist(0,q) then return(q+1) else for t: 1
		thru q do (if not M[i,t]=0 then return(t)));
		/* reduced echelon form */
		redeche(M):=block([p,q,r,i,j,t],[p,q]:matrix_size(M),M:echelon(M),r:rank(M),if r=1 then return (M),for i:r
		thru 2 step -1 do (j:getFirstNZ(M,i),for t:1 thru i-1 do (M:rowop(M,t,i,M[t,j]))),M);
		/* randechelon : random echelon form */
		randechelon(i1,j1,pv):=block([r,i,j,ret,tmp],r:length(pv);tmp[i,j]:=if j < pv[i] then 0 else if j=pv[i] then 1
		else rand(5)-2;
		/* ---------- under const -------------- */
		re:genmatrix(lambda([i,j],if i<length(pv) and j=pv[i] then 1 else 0),m,n);
		randechelonpv(n,m,pv):=block([re,i,j,r,k],r:length(pv),re:makelist(makelist(if i> r then 0 elseif j<pv[i] the
		0 else if j=pv[i] then 1 else rnz(am),j,1,n),i,1,m),if r>1 then for i:2 while(i<=r) do (j:pv[i],for k:1 while (k<
		do re[k,j]:0,return(re));
		randechelonpv(n,m,pv):=block([re,i,j,r,k],r:length(pv),re:makelist(makelist(if i> r then 0 elseif j<pv[i] the
		0 else if j=pv[i] then 1 else rnz(am),j,1,n),i,1,m),if r>1 then for i:2 while(i<=r) do (j:pv[i],for k:1 while (k<
		do re[k,j]:0,return(re));

図 5　STACK による実装例

6. おわりに

今後は実装仕様書より，まず，STACK の問題に変換するツールを構築し，Möbius Assessment やその他のシステムの問題に変換できるツールも構築する予定である。

参考文献

(1) 茂木勇，間下克哉：基礎演習シリーズ線形代数，裳華房，(1981).

(2) 横井英夫，尼野一夫：線形代数演習(数学演習ライブラリー1)，サイエンス社（1984）.

(3) Yasuyuki Nakamura, Tetsuya Taniguchi, Takahiro Nakahara: “Item Bank System for the Mathematics e-Learning System STACK”, Research Journal of Mathematics & Technology 3, 2, pp. 77-85 (2014).

(4) 吉冨賢太郎：“Mathematica 利用の数学到達度評価システムと STACK とのコンテンツ相互利用”, CIEC 第 100 回研究会，(2014).

(5) Yasuyuki Nakamura, Kentaro Yoshitomi, Mitsuru Kawazoe, Tetsuo Fukui, Shizuka Shirai, Takahiro Nakahara, Katsuya Kato, Tetsuya Taniguchi: “Effective Use of Math E-Learning with Questions Specification”, Distance Learning, E-Learning and Blended Learning in Mathematics Education, pp.133-148 (2018).