法政大学学術機関リポジトリ
HOSEI UNIVERSITY REPOSITORY

PDF issue: 2025-07-06

Computational Complexity from Learners' Perspective in College Mathematics

森, 優貴 / MORI, Yuki

(出版者 / Publisher)
法政大学大学院情報科学研究科

(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 情報科学研究科編 / 法政大学大学院紀要. 情報科学研究科編

(巻 / Volume)
11

(開始ページ / Start Page)
1

(終了ページ / End Page)
6

(発行年 / Year)
2016-03-24

(URL)
https://doi.org/10.15002/00012906

Computational Complexity from Learners' Perspective in College Mathematics

Yuki Mori
Graduate School of Computer and
Information Sciences,
Hosei University,
Tokyo, Japan
Email: yuuki.mori.72@stu.hosei.ac.jp

***Abstract*—Suitably complex exercises motivate students and facilitates a deeper understanding. Manually constructing such problems consumes time that teachers can otherwise use to mentor students. Many software tools and services for automatic generation of mathematics problems are available on the web, but they provide only materials up to high school level. In addition, no standardized methods are provided to evaluate and control the computational complexity of generated problems. In this paper, we proposed a framework for evaluating computational complexity from the learners' perspective, aiming to apply our framework to the automatic generation of college-level mathematics problems with controlled computational complexity. Our framework helps teachers prepare learning materials and thereby save time for mentoring students.**

I. Introduction

We propose a framework for evaluating the computational complexity of college-level mathematics problems, with the aim of applying our framework to automatic generation of such problems controlled computational complexity.

Providing students with suitably complex practice problems is crucial for motivating them and facilitating deeper understanding. Manually constructing such problems consumes time, which mathematics teachers can otherwise use to mentor students.

Many software tools and services for automatic generation of mathematics problems are available on the web, but they provide only materials up to high-school level. In addition, no standardized methods are provided to evaluate and control the computational complexity of the generated problems. Among the popular web sites and services, we list some examples. Wolfram Problem Generator^{TM} [1] and Davitily Math Problem Generator^{TM} [2] deal with mathematics problems for high-school students. SuperKids Math Worksheet Creator^{TM} [3] deals with arithmetic problems for children attending elementary schools. However, our framework is new as it deals with math at the college level and introduces suitable methods for evaluating computational complexity from the learners' perspective.

Eigenvalue problems and differential equations are usually taught in linear algebra and calculus courses at engineering departments. Since most textbooks do not provide sufficient practice problems, teachers must make additional problems to be used in classes, assignments, and exams. We took this two topics as a case study to develop our framework.

The system let the user select parameters such as the algebraic number field used in the calculation, the number of the calculation steps, and the matrix dimension, which determine the outline of the problems. The user then selects the problem category and provides the required parameters that control the computational complexity. To further reduce the burden for busy users, predefined sets of recommended parameters are also stored in the system, which eases the parameter selection process. Problems with the required complexity along with model answers are generated.

In this paper, we present automatic problem generators for differential equations and diagonalization problems for Hermitian matrices to illustrate the relevance of our proposed framework.

II. Complexity from Learners' Perspective

General concepts of complexity are available in various forms in standard textbooks such as [5]. We deal with a different variety of complexity, subjective complexity, where complexity is measured by the difficulty from learners' perspective.

Designing practice problems that are sufficiently but not excessively complex is crucial for keeping a learner motivated. We propose a new framework for estimating such computational complexity and demonstrate its relevance by developing a system for automatic generation of complexity-controlled practice problems. Our framework enables us to

1) control the number of the calculation steps,
2) limit the height of rational numbers involved in a calculation, and
3) deal with algebraic numbers.

The computational complexity of generated problems is mainly determined by the sum of the heights of the rational numbers (the maximum ratio of the absolute values of the denominator and numerator) appearing in its model solution. In the hope of extending our work to other mathematics areas, we incorporated controls over algebraic number fields in our system. The user can select the calculation field from the rational number field and other algebraic fields extended by irrational numbers, especially, quadratic irrational numbers, and fourth-power irrational numbers.

[0]Supervisor: Prof. Shuichi YUKITA

A. Automatic Generation of Linear Equations

This subsection is a summary of our early work on the linear equation problems aiming to control the complexity from the learners' perspective. Solving linear equations is a part of eigenvalue problems. The system generates linear equations as the elementary row operations problems and model answers. The row operations are classified as follows.

1) Row switching (A row within the matrix can be switched with another row.)
2) Row multiplication (Each element in a row can be multiplied by a non-zero constant.)
3) Row addition (A row can be replaced by the sum of that row and a multiple of another row.)

It is commonly practiced that the matrix representation of the problem is reversely constructed from the solution space. Teachers do such reverse-row operations by their hands, controlling the level of difficulty at every step of the transformation by their naive concept of complexity. Our system does all these things automatically and controls perfectly the computational complexity as specified by the user.

The system controls the number of the calculation steps by limiting the number of the reverse-row operations steps, and limits the heights of rationals in the row multiplication and row addition.

In the following sections, we will show an application of our framework to college mathematics problems and discuss the computational complexity from learners' perspective.

III. Complexity in Eigenvalue Problems

In this section, we discuss complexity of linear algebra problems from learners' perspective.

Eigenvalue problems are usually taught in linear algebra courses at engineering departments. Eigenvalue problems appear in two forms: diagonalization of Hermitian forms and Jordan canonicalization of linear endomorphisms. In this paper, we deal with diagonalization of Hermitian matrices. The process of generating complexity-controlled eigenvalue problems includes

1) creating a pool of nice looking matrices,
2) generating eigenvalues with specified multiplicities,
3) generating a set of candidate Hermitian matrices, and
4) filtering out those matrices that are not suitable for exercises with some criteria.

We explain each step in detail. First, we generate nearly the entire set of tractable unitary matrices and classify them by algebraic number fields. We define an utility function that can extract all of the irrational numbers appearing in the entries of a tentatively generated matrix. For example, this function returns the list $[\sqrt{-1}, \sqrt{2}, \sqrt{3}, \sqrt{5}]$, where

$$\begin{pmatrix} \sqrt{-5} & \sqrt{2} \\ \frac{1}{\sqrt{3}} & 7 \end{pmatrix} \tag{1}$$

is entered as a primary material that is later subject to Gram-Schmidt orthonormalization. Diagonalizing a given Hermitian matrix requires

1) calculating eigenvalues,
2) selecting mutually orthogonal eigenspaces each of which corresponds to an eigenvalues, and
3) constructing a unitary matrix using these eigenvectors.

In the requirement 2, selecting an eigenspace is either giving an eigenvector or giving a set of eigenvectors. In the latter case, the eigenvalue has multiplicity greater than 1. Note that the choice of eigenvectors in degenerate cases does not affect the final result. Hence, the number of calculation steps does not vary once all the multiplicities are specified. General Hermitian matrices can be generated from a diagonal matrix D and a unitary matrix U as

$$H = UDU^{\dagger} \tag{2}$$

where $U^{\dagger}$ is the conjugate transpose matrix of U. Equation (2) is rewritten as $D = U^{\dagger}HU$. The most difficult part is generating a unitary matrix with the specified properties. However, the number of matrices suitable for this purpose is relatively small because the entries of those matrices must be obtained from a given algebraic number field, and the heights of the involved rationals must be restricted, furthermore, all of the column vectors must form an orthonormal system. Therefore it is possible to predefine almost entire sets of unitary matrices that can be used to generate Hermitian matrices that can be diagonalized with specified complexity. This stage can be skipped once the pool is created. However, we may recreate another pool with slightly different parameters.

A. Generation of Unitary Matrices

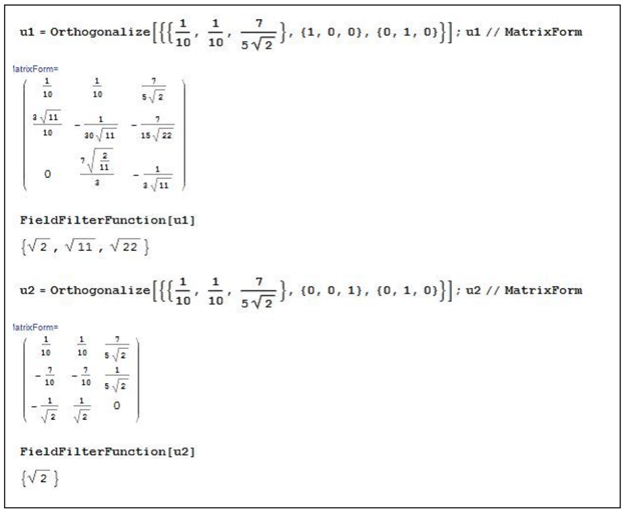
This section describes the generation of 3×3 unitary matrices through examples. The procedure for generating a matrix comprises four major steps:

1) generating a unit column vector,
2) verifying that all the entries belong to the given number field,
3) constructing an orthonormal basis from two other linearly independent column vectors using the Gram-Schmidt procedure, and
4) verifying again that all of the entries of those basis vectors belong to the given number field.

In step 1, we generate various unit vectors that are the form in Eq. (3).

$$\boldsymbol{e}_1{}^t = (\pm\frac{i}{j\sqrt{k}}, \pm\frac{l}{m\sqrt{n}}, \pm\sqrt{1-((\frac{i}{j\sqrt{k}})^2+(\frac{l}{m\sqrt{n}})^2)}), \tag{3}$$

where $i, j, l,$ and m are rational integers, and k and n are $2, 3, 5, 7,$ or 1. In step 2, all of the irrational numbers such as $\sqrt{2}, \sqrt{3},$ and $\sqrt{-1}$ are extracted from the vector and matrix. We select the vectors whose entries belong to the specified number field. In step 3, e_1 and two other vectors are orthogonalized. Only an additional two vectors are required to form a linearly independent triple together with e_1. Hence, for easy calculations, we can take them from sparse matrices, where only the positions of nonzero entries are important. Figure 1 indicates that the field may possibly be extended after orthogonalization.



u1 = Orthogonalize$\left[\left\{\left\{\frac{1}{10}, \frac{1}{10}, \frac{7}{5\sqrt{2}}\right\}, \{1, 0, 0\}, \{0, 1, 0\}\right\}\right]$; u1 // MatrixForm

latrixForm=

$$\begin{pmatrix} \frac{1}{10} & \frac{1}{10} & \frac{7}{5\sqrt{2}} \\ \frac{3\sqrt{11}}{10} & -\frac{1}{30\sqrt{11}} & -\frac{7}{15\sqrt{22}} \\ 0 & \frac{7\sqrt{\frac{2}{11}}}{3} & -\frac{1}{3\sqrt{11}} \end{pmatrix}$$

FieldFilterFunction[u1]

$\{\sqrt{2}, \sqrt{11}, \sqrt{22}\}$

u2 = Orthogonalize$\left[\left\{\left\{\frac{1}{10}, \frac{1}{10}, \frac{7}{5\sqrt{2}}\right\}, \{0, 0, 1\}, \{0, 1, 0\}\right\}\right]$; u2 // MatrixForm

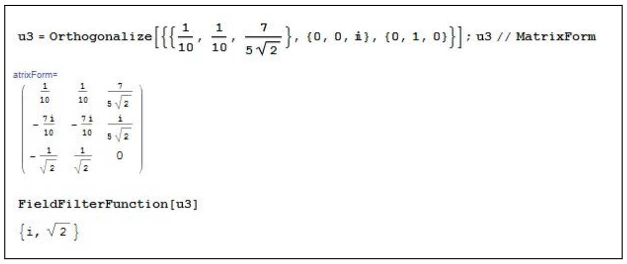
latrixForm=

$$\begin{pmatrix} \frac{1}{10} & \frac{1}{10} & \frac{7}{5\sqrt{2}} \\ -\frac{7}{10} & -\frac{7}{10} & \frac{1}{5\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{pmatrix}$$

FieldFilterFunction[u2]

$\{\sqrt{2}\}$

Fig. 1. Example of orthogonalization. The upper part shows failure in retaining the number field after orthogonalization in the case in which $(1,0,0)^t$ and $(0,0,1)^t$ are appended. The other shows success in retaining the field, when $(0,0,1)^t$ and $(0,0,1)^t$ are appended.



u3 = Orthogonalize$\left[\left\{\left\{\frac{1}{10}, \frac{1}{10}, \frac{7}{5\sqrt{2}}\right\}, \{0, 0, i\}, \{0, 1, 0\}\right\}\right]$; u3 // MatrixForm

atrixForm=

$$\begin{pmatrix} \frac{1}{10} & \frac{1}{10} & \frac{7}{5\sqrt{2}} \\ -\frac{7i}{10} & -\frac{7i}{10} & \frac{i}{5\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{pmatrix}$$

FieldFilterFunction[u3]

$\{i, \sqrt{2}\}$

Fig. 2. Example of expansion of the orthogonalization to imaginary matrices.

We generated three patterns of unitary matrices by changing the position of the non-zero element of each vector. If the user wants a complex number field, then $\sqrt{-1}$ must be added in some entry of an initial vector (see Fig. 2). In step 4, the number field of the components is verified again. This step is necessary because Gram-Schmidt orthogonalization takes square roots which may cause further algebraic extension of fields. We select matrices all of whose entries have rationals of low heights in their subexpressions. These forms the basic set of tractable unitary matrices. Of the 500,000 generated unitary matrices in a preliminary stage, the filter selects 681 matrices according to the criterion described in later sections. The number of predefined matrices are listed in Tabs. I and II. Though the basic set is relatively small (681), we can generate other tractable matrices by multiplying them among themselves and by taking direct sums as follows: given two matrices of the same dimension

$$U_1 \text{ and } U_2 \in U(n), \tag{4}$$

we obtain

$$U_1U_2 \in U(n). \tag{5}$$

Given two matrices of possibly different dimensions

$$U_1 \in U(m) \text{ and } U_2 \in U(n), \tag{6}$$

we obtain

$$U_1 \bigoplus U_2 \in U(m+n). \tag{7}$$

Note that the number field involved is preserved under both multiplication and direct sum operations. For example,

$$\begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} \sqrt{-1} & 0 \\ 0 & 1 \end{pmatrix}. \tag{8}$$

Using such methods, we can obtain sufficient unitary matrices, as presented in Tabs. I and II.

TABLE I. PREDEFINED ORTHOGONAL MATRICES

Field \ Dimension	2×2	3×3	4×4
Q	65	20	4245
$Q(\sqrt{2})$	27	33	2503
$Q(\sqrt{3})$	12	10	930
$Q(\sqrt{5})$	20	25	1714
$Q(\sqrt{7})$	12	3	927

TABLE II. PREDEFINED UNITARY MATRICES

Field \ Dimension	2×2	3×3	4×4
$Q(\sqrt{-1})$	129	20	25066
$Q(\sqrt{2},\sqrt{-1})$	65	33	8488
$Q(\sqrt{3},\sqrt{-1})$	30	11	2862
$Q(\sqrt{5},\sqrt{-1})$	68	26	9068
$Q(\sqrt{7},\sqrt{-1})$	24	5	2142

IV. Complexity in Differential Equations

In this section, we discuss complexity of differential equation problems from learners' perspective.

Differential equations are usually taught in linear algebra courses at senior classes. The types of differential equations we treat here are:

1) separable
2) homogeneous
3) total, and
4) linear.

Compared with eigenvalue problems, the number of the calculation steps is quite important to evaluate the complexity of differential equations. This is because a differential equation includes various types of integration, and also the most tedious work is to solve integration of the problem. In our framework, the calculation steps of basic types of integration are predefined. For exanple, in the case of integration of $1/\sin x$, the solution includes:

1) multiplying $\sin x$ to both the denominator and the numerator,
2) rewriting the denominator as $1-\cos^2 x$,
3) parting the fraction into sum of $(1/2)\{\sin x/(1-\cos x)) + (\sin x/(1+\cos x)\}$, and
4) integrating each separated fraction.

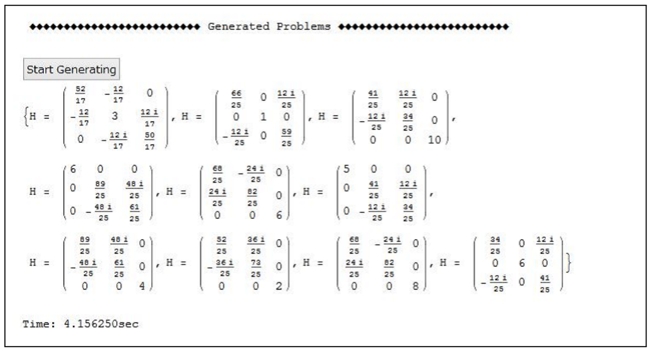
Therefore, the number of the calculation steps is 4 to integrate $1/\sin x$. We predefined the numbers of the calculation steps of basic types of integration. See Tab. III.

TABLE III. NUMBERS OF THE CALCULATION STEPS OF SIMPLE TYPES OF INTEGRATION

Function	Integration	Steps
$\int a dx$	$ax + C$	1
$\int x^a dx$	$\frac{x^{a+1}}{a+1}$	1
$\int x^{-1} dx$	$\log x$	1
$\int \frac{dx}{1-x^2}$	$\arcsin x$	4
$\int \frac{dx}{1+x^2}$	$\arctan x$	4
$\int e^x dx$	$e^x + C$	1
$\int a^x dx$	$\frac{a^x}{\log a}$	1
$\int \log x dx$	$x(\log x - 1)$	3
$\int \sin x dx$	$-\cos x$	1
$\int \cos x dx$	$\sin x$	1
$\int \tan x dx$	$-\log \cos x$	2
$\int \csc x dx$	$\frac{1}{2} \log \lvert \frac{1-\cos x}{1+\cos x} \rvert$	4
$\int \sec x dx$	$\frac{1}{2} \log \lvert \frac{1+\sin x}{1-\sin x} \rvert$	4
$\int \cot x dx$	$\log \lvert \sin x \rvert$	2
$\int \sin^2 x dx$	$\frac{1}{2}(x - \sin x \cos x)$	4
$\int \cos^2 x dx$	$\frac{1}{2}(x + \sin x \cos x)$	4
$\int \tan^2 x dx$	$\tan x - x$	3
$\int \cot^2 x dx$	$-\cot x - x$	3
$\int \sin ax \sin bx dx$	$\frac{\sin(a-b)x}{2(a-b)} - \frac{\sin(a+b)x}{2(ab)}$	6
$\int \sin ax \cos bx dx$	$\frac{\cos(a-b)x}{2(a-b)} - \frac{\cos(a+b)x}{2(a+b)}$	6
$\int \cos ax \cos bx dx$	$\frac{\sin(a-b)x}{2(a-b)} - \frac{\sin(a+b)x}{2(a+b)}$	6
$\int x \sin x dx$	$\sin x - x \cos x$	3
$\int x \cos x dx$	$\cos x + x \sin x$	3
$\int x^2 \sin x dx$	$(2 - x^2) \cos x + 2 \sin x$	5
$\int x^2 \cos x dx$	$(x^2 - 2) \sin x + 2x \cos x$	5
$\int \sec^2 x$	$\tan x$	1
$\int \arcsin x dx$	$\sqrt{1-x^2} + x \arcsin x$	6
$\int \arccos x dx$	$-\sqrt{1-x^2} + x \arccos x$	6
$\int \arctan x dx$	$x \arctan x - \frac{1}{2} \log(1 + x^2)$	8

TABLE IV. NUMBERS OF THE CALCULATION STEPS OF DIFFERENTIAL EQUATIONS

Equation	Steps
$y' = f(x)g(y)$	$N(\int \frac{dy}{g(y)}) + N(\int f(x)dx)$
$y' = f(ax + by + c)$	$8 + N(\int \frac{du}{bf(u)+a})$
$y' = f(\frac{y}{x})$	$8 + N(\int \frac{du}{f(u)-u})$
$P(x,y)dx + Q(x,y)dy = 0$	$3 + N(\int P(x,y)dx) + N(\int \{Q(x,y) - \frac{\partial}{\partial y} \int P(x,y)dx\}dy)$


◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆ Generated Problems ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

$$\left\{H = \begin{pmatrix} \frac{52}{17} & -\frac{12}{17} & 0 \\ -\frac{12}{17} & 3 & \frac{12 i}{17} \\ 0 & -\frac{12 i}{17} & \frac{50}{17} \end{pmatrix}, H = \begin{pmatrix} \frac{66}{25} & 0 & \frac{12 i}{25} \\ 0 & 1 & 0 \\ -\frac{12 i}{25} & 0 & \frac{59}{25} \end{pmatrix}, H = \begin{pmatrix} \frac{41}{25} & \frac{12 i}{25} & 0 \\ -\frac{12 i}{25} & \frac{34}{25} & 0 \\ 0 & 0 & 10 \end{pmatrix},\right.$$

$$H = \begin{pmatrix} 6 & 0 & 0 \\ 0 & \frac{89}{25} & \frac{48 i}{25} \\ 0 & -\frac{48 i}{25} & \frac{61}{25} \end{pmatrix}, H = \begin{pmatrix} \frac{68}{25} & -\frac{24 i}{25} & 0 \\ \frac{24 i}{25} & \frac{82}{25} & 0 \\ 0 & 0 & 6 \end{pmatrix}, H = \begin{pmatrix} 5 & 0 & 0 \\ 0 & \frac{41}{25} & \frac{12 i}{25} \\ 0 & -\frac{12 i}{25} & \frac{34}{25} \end{pmatrix},$$

$$\left. H = \begin{pmatrix} \frac{89}{25} & \frac{48 i}{25} & 0 \\ -\frac{48 i}{25} & \frac{61}{25} & 0 \\ 0 & 0 & 4 \end{pmatrix}, H = \begin{pmatrix} \frac{52}{25} & \frac{36 i}{25} & 0 \\ -\frac{36 i}{25} & \frac{73}{25} & 0 \\ 0 & 0 & 2 \end{pmatrix}, H = \begin{pmatrix} \frac{68}{25} & -\frac{24 i}{25} & 0 \\ \frac{24 i}{25} & \frac{82}{25} & 0 \\ 0 & 0 & 8 \end{pmatrix}, H = \begin{pmatrix} \frac{34}{25} & 0 & \frac{12 i}{25} \\ 0 & 6 & 0 \\ -\frac{12 i}{25} & 0 & \frac{41}{25} \end{pmatrix}\right\}$$

```
Time: 4.156250sec
```

Fig. 3. Samples of generated Hermitian matrix

Secondary, the number of the calculation steps of differential equation were predefined. For example, a homogeneous differential equation requires such calculation steps as follows. When a given

$$\frac{dy}{dx} = g(x, y) \tag{9}$$

can be written as

$$\frac{dy}{dx} = f(\frac{y}{x}), \tag{10}$$

the differential equation is homogeneous. A homogeneous differential equation can be solved by change of variables.

$$y = ux, \tag{11}$$

where $u = y/x$. And then

$$\frac{dy}{dx} = u'x + u. \tag{12}$$

Given equation can be written as follows.

$$u'x + u = f(u) \tag{13}$$

By separating variables x and u,

$$\int \frac{du}{f(u) - u} = \int \frac{dx}{x}. \tag{14}$$

The right hand side can be integrated, so we get

$$\int \frac{du}{f(u) - u} = \log x + C_1, \tag{15}$$

where C_1 is constant. Where integration of $1/(f(u) - u)$ is $F(u)$ and C_2 is constant,

$$F(u) = \log x + C_2. \tag{16}$$

Finally, we reverse the change of variables $u = y/x$,

$$y = H(x), \tag{17}$$

where $H(x)$ satisfies the given equation. Therefore, a homogeneous type takes at least 8 steps. Except the eight steps, the number of the calculation steps depends on the steps of integration of $1/(f(u) - u)$. The complexity of homogeneous equations can be controlled by restricting the difficulty of integration of $1/(f(u) - u)$.

Table IV indicates the number of the calculation steps of differential equations. $N(\int dy/y)$ means the number of the calculation steps of $\int dy/y$.

V. DEMONSTRATION AND DISCUSSION

In this section, we demonstrate automatic generations of eigenvalue problems for Hermitian matrices and differential equation problems, and then evaluate the complexity of produced problems.

A. *Demonstration of Eigenvalue Problems*

Figure 3 presents the results of ten generated Hermitian matrices. The dimension of each matrix, multiplicity of eigenvalues, algebraic number field, height h of numerical calculation, and number n of problems is 3, 1, $\sqrt{-1}$, 100, and 10, respectively. The system generates n problems on demand. Each matrix

1) has a maximum absolute value of involved rationals less than h,
2) has entries that belong to specified number field, and

TABLE V. TIME (SEC) FOR GENERATING PROBLEM 1

	2×2	3×3	4×4
Q	5.33	17.14	1615.72
$Q(\sqrt{2})$	702.81	1185.03	768.2
$Q(\sqrt{3})$	702.00	1324.84	810.72
$Q(\sqrt{5})$	353.91	1677.19	484.39
$Q(\sqrt{7})$	320.05	3154.67	1684.18

TABLE VI. TIME (SEC) FOR GENERATING PROBLEM 2

	2×2	3×3	4×4
$Q(\sqrt{-1})$	110.61	170.72	1712.11
$Q(\sqrt{2}, \sqrt{-1})$	655.34	2175.56	809.42
$Q(\sqrt{3}, \sqrt{-1})$	599.29	4585.61	924.39
$Q(\sqrt{5}, \sqrt{-1})$	691.75	3439.33	500.33
$Q(\sqrt{7}, \sqrt{-1})$	3543.17	22911.11	1821.26

3) differs from already generated matrices.

In this case, generating ten problems took 4.16 seconds. The time measurements for generating problems are listed in Tabs. V and VI. The machine's specification is as follows. The adopted software is Worflam Mathematica 9.0 (Windows 8.1, 64 bit, Intel(R) Core i5-4300U CPU 1.9 GHz. The main memory is 8.0 GB.)

Tests were conducted generating 1,000 problems for each number field. Generating one problem takes 1.18 seconds on average. As the tables indicate, generation consumes more time when the number field is complex because the system needs to decompose all of the matrix entries to check the maximum height of the rational numbers.

As a preliminary experiment, we asked 10 CS department students to solve two sets of problems: one set consists of the problems with controlled complexity, the other consists of uncontrolled ones. See Tabs. VII and VIII. On average, the former case, students required 2 min to calculate eigenvalues and 6 min to construct a unitary matrix, while for the latter case students required 6 min to calculate eigenvalues and 7 min to form a unitary matrix. In addition, 4 students could not reach the answers of the uncontrolled problems at first.

B. Discussion of Eigenvalue Problems

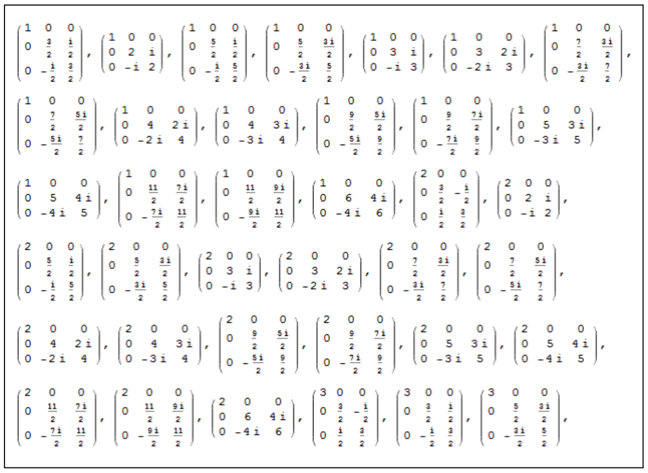
In our experiments, some learners felt excessive difficulty in calculating characteristic equations to get eigenvalues. The complexity of calculating eigenvalues can be reduced by restricting unitary matrices to more tractable ones. In other words, some of predefined unitary matrices are not tractable. We knew that the unitary matrices which generate good Hermitian matrices can be used to generate other good Hermitian matrices by rechoosing eigenvalues.

TABLE VII. THE NUMBER OF THE STUDENTS BELONG TO EACH TIME (MIN) FOR SOLVING THE GENERATED PROBLEMS.

	Eigenvalues			Eigenvectors
1s	1 student		4s	3 students
2s	5 students		5s	5 students
3s	4 students		6s	2 students

TABLE VIII. THE NUMBER OF THE STUDENTS BELONG TO EACH TIME (MIN) FOR SOLVING THE NON-COMPLEXITY-CONTROLLED PROBLEMS.

	Eigenvalues			Eigenvectors
5s	1 student		6s	1 students
6s	3 students		7s	4 students
7s	6 students		8s	5 students



$$\begin{pmatrix}1&0&0\\0&\frac{3}{2}&\frac{i}{2}\\0&-\frac{i}{2}&\frac{3}{2}\end{pmatrix}, \begin{pmatrix}1&0&0\\0&2&i\\0&-i&2\end{pmatrix}, \begin{pmatrix}1&0&0\\0&\frac{5}{2}&\frac{i}{2}\\0&-\frac{i}{2}&\frac{5}{2}\end{pmatrix}, \begin{pmatrix}1&0&0\\0&\frac{5}{2}&\frac{3i}{2}\\0&-\frac{3i}{2}&\frac{5}{2}\end{pmatrix}, \begin{pmatrix}1&0&0\\0&3&i\\0&-i&3\end{pmatrix}, \begin{pmatrix}1&0&0\\0&3&2i\\0&-2i&3\end{pmatrix}, \begin{pmatrix}1&0&0\\0&\frac{7}{2}&\frac{3i}{2}\\0&-\frac{3i}{2}&\frac{7}{2}\end{pmatrix},$$

$$\begin{pmatrix}1&0&0\\0&\frac{7}{2}&\frac{5i}{2}\\0&-\frac{5i}{2}&\frac{7}{2}\end{pmatrix}, \begin{pmatrix}1&0&0\\0&4&2i\\0&-2i&4\end{pmatrix}, \begin{pmatrix}1&0&0\\0&4&3i\\0&-3i&4\end{pmatrix}, \begin{pmatrix}1&0&0\\0&\frac{9}{2}&\frac{5i}{2}\\0&-\frac{5i}{2}&\frac{9}{2}\end{pmatrix}, \begin{pmatrix}1&0&0\\0&\frac{9}{2}&\frac{7i}{2}\\0&-\frac{7i}{2}&\frac{9}{2}\end{pmatrix}, \begin{pmatrix}1&0&0\\0&5&3i\\0&-3i&5\end{pmatrix},$$

$$\begin{pmatrix}1&0&0\\0&5&4i\\0&-4i&5\end{pmatrix}, \begin{pmatrix}1&0&0\\0&\frac{11}{2}&\frac{7i}{2}\\0&-\frac{7i}{2}&\frac{11}{2}\end{pmatrix}, \begin{pmatrix}1&0&0\\0&\frac{11}{2}&\frac{9i}{2}\\0&-\frac{9i}{2}&\frac{11}{2}\end{pmatrix}, \begin{pmatrix}1&0&0\\0&6&4i\\0&-4i&6\end{pmatrix}, \begin{pmatrix}2&0&0\\0&\frac{3}{2}&-\frac{i}{2}\\0&\frac{i}{2}&\frac{3}{2}\end{pmatrix}, \begin{pmatrix}2&0&0\\0&2&i\\0&-i&2\end{pmatrix},$$

$$\begin{pmatrix}2&0&0\\0&\frac{5}{2}&\frac{i}{2}\\0&-\frac{i}{2}&\frac{5}{2}\end{pmatrix}, \begin{pmatrix}2&0&0\\0&\frac{5}{2}&\frac{3i}{2}\\0&-\frac{3i}{2}&\frac{5}{2}\end{pmatrix}, \begin{pmatrix}2&0&0\\0&3&i\\0&-i&3\end{pmatrix}, \begin{pmatrix}2&0&0\\0&3&2i\\0&-2i&3\end{pmatrix}, \begin{pmatrix}2&0&0\\0&\frac{7}{2}&\frac{3i}{2}\\0&-\frac{3i}{2}&\frac{7}{2}\end{pmatrix}, \begin{pmatrix}2&0&0\\0&\frac{7}{2}&\frac{5i}{2}\\0&-\frac{5i}{2}&\frac{7}{2}\end{pmatrix},$$

$$\begin{pmatrix}2&0&0\\0&4&2i\\0&-2i&4\end{pmatrix}, \begin{pmatrix}2&0&0\\0&4&3i\\0&-3i&4\end{pmatrix}, \begin{pmatrix}2&0&0\\0&\frac{9}{2}&\frac{5i}{2}\\0&-\frac{5i}{2}&\frac{9}{2}\end{pmatrix}, \begin{pmatrix}2&0&0\\0&\frac{9}{2}&\frac{7i}{2}\\0&-\frac{7i}{2}&\frac{9}{2}\end{pmatrix}, \begin{pmatrix}2&0&0\\0&5&3i\\0&-3i&5\end{pmatrix}, \begin{pmatrix}2&0&0\\0&5&4i\\0&-4i&5\end{pmatrix},$$

$$\begin{pmatrix}2&0&0\\0&\frac{11}{2}&\frac{7i}{2}\\0&-\frac{7i}{2}&\frac{11}{2}\end{pmatrix}, \begin{pmatrix}2&0&0\\0&\frac{11}{2}&\frac{9i}{2}\\0&-\frac{9i}{2}&\frac{11}{2}\end{pmatrix}, \begin{pmatrix}2&0&0\\0&6&4i\\0&-4i&6\end{pmatrix}, \begin{pmatrix}3&0&0\\0&\frac{3}{2}&-\frac{i}{2}\\0&\frac{i}{2}&\frac{3}{2}\end{pmatrix}, \begin{pmatrix}3&0&0\\0&\frac{3}{2}&\frac{i}{2}\\0&-\frac{i}{2}&\frac{3}{2}\end{pmatrix}, \begin{pmatrix}3&0&0\\0&\frac{5}{2}&\frac{3i}{2}\\0&-\frac{3i}{2}&\frac{5}{2}\end{pmatrix},$$

Fig. 4. Generation of Hermitian matrices from one unitary matrix by changing eigenvalues

Some teachers criticized that irrational numbers such as $\sqrt{5}$ and $\sqrt{7}$ look ugly as exercise problems. Therefore we chose only matrices which include Q, $Q(\sqrt{2})$ and $Q(\sqrt{3})$. Finally, we got 174 tractable 3-dimensional unitary matrices after changing some parameters. These predefined unitary matrices include other variable number field such as $Q(\sqrt{10})$ and $Q(\sqrt{13})$ because these unitary matrices often generate nice looking unitary matrices regardless of including such irrational numbers.

The number of finally obtained matrices is 174. Although this number seems smaller than what we obtained in our previous work, we can still get virtually infinite Hermitian matrices because of virtually infinite assignments of eigenvalues to eigenspaces. See Fig. 4. These Hermitian matrices are generated by using one unitary matrix by changing a target diagonal matrix.

Learners also felt excessive difficulty because the number fields varied from ones that we specified. The number field varied because of procedure of orthogonalization. Learners should start orthogonalization of vectors in ascending order of the norms. When they refer to the generated model answers, they may learn the importance of the procedure of orthogonalization.

Some parts of the automatic generation system of row-operations problems, our early work, can be reused in the generation of eigenvalue problems. Calculation of eigenvectors involves row-operations.

After this refinement, we conducted a second experiment in which we asked the same CS students to solve the generated problems again. See a Tab. IX. Students required 2 min to calculate eigenvalues and 5 min to construct a unitary matrix. And also, teachers satisfied the complexity-controlled problems.

TABLE IX. THE NUMBER OF THE STUDENTS BELONG TO EACH TIME (MIN) FOR SOLVING THE GENERATED PROBLEMS.

	Eigenvalues			Eigenvectors
1s	1 student		4s	4 students
2s	5 students		5s	4 students
3s	4 students		6s	2 students

TABLE X. THE NUMBER OF THE STUDENTS BELONG TO EACH TIME (MIN) FOR SOLVING THE GENERATED EQUATIONS.

	Equation 19	Equation 20	Equation 21
Student 1	20s	23s	27s
Student 2	25s	28s	33s
Student 3	15s	18s	19s
Student 4	21s	23s	25s
Student 5	22s	25s	31s

C. Demonstration of differential equations

We show some examples of generation of differential equations.

Here, we focus on the computational complexity of separable differential equations. $y' = (ax + xy + c)^2$ belongs to separable equations, where $u = ax + by + c$. The ansewer of the problem is below.

$$y = \frac{1}{b}\{\sqrt{\frac{a}{b}}\tan(\sqrt{ab}x + K) - ax - c\}, \tag{18}$$

where K is constant. Therefore, the values a and b are important for controlling the complexity. For example,

$$y' = (x + y - 3)^2, \tag{19}$$
$$y' = (9x + 4y - 1)^2, \text{and} \tag{20}$$
$$y' = (3x + 5y + 3)^2, \tag{21}$$

were generated by our system and the parameters of algebraic number fild are set as follows. Equation 19 belongs to Q and has 16 calculation steps. Equation 20 belongs to Q and has 18 calculation steps. Equation 21 belongs to $Q(\sqrt{3}, \sqrt{5})$ and has 18 calculation steps.

D. Discussion of Differential Equations

We asked 5 CS students to solve the separable equations which were Eq.19, Eq.20, and Eq.21. The 5 students knew how to solve such separable equations like $y' = (ax + by + c)^2$ which reqire change of variables $u = ax + by + c$, but the time measurements of their solving each equation were little different. See Tab. X. They took 20 min for Eq. 19, 23 min for Eq. 20, and 27 min for Eq. 21. They told us that the problem of $Q(\sqrt{3}, \sqrt{5})$ was so much difficult even though the number of the calcualtion steps of each problem did not vary. In other words, the integration of $1/(5u^2 + 3)$ confused the learners to solve Eq. 21. The result shows exsitence of learners' perspective complexity which takes them much time to calcualte aside from the true nature of the problems.

These results demonstrate the validity of our proposed framework. Large-scale verification experiments will be conducted out with the cooperation of university teachers.

VI. CONCLUSION

Constructing problems with sufficient computational complexity is essential for maintaining learners' motivation. In this paper, we presented a new framework for evaluating and controlling computational complexity of mathematics problems from the learners' perspective. In addition, we developed an automatic generation system for eigenvalue problems and differential equations based on our framework. The automatic generation of complexity-controlled eigenvalue problems and differential equations is one of the sample implementations that validate our framework. Controlling the complexity of mathematics problems involves restricting the number of calculation steps, the heights of the involved rationals, and the algebraic number field appearing in a model solution. In small-scale preliminary experiments, we could decrease the computational complexity and answering time.

The system can generate virtually an infinite number of exercise problems. Our framework helps teachers prepare learning materials and thereby save time for mentoring students.

We expect our technique to be applicable to other subjects in linear algebra and analysis. Our system paves a path to strict quantitative control on various aspects of complexity from the learners' perspective. Future work will include the application of the row-operations system to the evaluating the complexity of a constructing a unitary matrix. And also we will apply our methods to other areas such as number theory. We also would like to show that this system has a positive effect on college-level engineering education. Therefore, demonstration experiments in classroom will be the next step of our research.

REFERENCES

[1] Wolfram Research (2012), Wolfram Problem Generator, ⟨http://www.wolframalpha.com/problem-generator/⟩, (Access: 2013/4/17).

[2] David Pardue and Ly Nguyen (2007), Davitily Math Problem Generator, ⟨http://www.mathproblemgenerator.com/⟩, (Access: 2013/4/17).

[3] Andrew Maisel (1996), SuperKids Math Worksheet Creator, ⟨http://www.superkids.com/aweb/tools/math/⟩, (Access: 2013/4/17).

[4] LLC (2000), Math Fact Cafe, ⟨http://www.mathfactcafe.com/worksheet/buildit/⟩, (Access: 2013/4/17).

[5] Michael Sipser (1997), Introduction to the Theory of Computation, PWS PUBLISHING COMPANY.

[6] Wolfram Research (2009), Wolfram Mathematica 9 Documentation, ⟨http://reference.wolfram.com/language/⟩, (Access: 2013/4/20).

[7] Geza Schay (2012), A Concise Introduction to Linear Algebra, Brikhauser; 2012 edition.

[8] Henry Ricardo (2009), A Modern Introduction to Linear Algebra, Chapman and Hall/CRC

[9] Paul J.McCarthy (1996), Algebraic Extensions of Fields, Dover Publications, Inc., New York.

[10] James.H.Wilkinson, (1988), The Algebraic Eigenvalue Problem (Monograph on Numerical Analysis), Oxford University Press; 1 edition

[11] John William and Scott Cassels (2010), Algebraic Number Theory, London Mathematical Society

[12] Sandy Kemsley (2000), abc teach, ⟨http://www.abcteach.com/⟩, (Access: 2014/6/21).

[13] Ben Taylor Langton (1999), Quick Math, ⟨http://www.quickmath.com/⟩, (Access: 2014/6/21)