

# An educational program of teaching mathematics with computer music tasks

Itaru Kaneko\*

**Abstract:** This paper reports on an experimental program for high school students that is aimed at encouraging them to use mathematics by offering the experience of synthesizing mathematically generated music signals using high school math. The program consists of two parts, the lecture and workshop, runs for a total of six hours. The first part explains the basic ideas of making simple tones and sounds using mathematical functions and expressions to construct simple tone scales, rhythm, and harmony. The second part is a workshop. Students may freely create their own music, using the technique they learn in the lecture or any other equations they write on their own. We taught the course more than eight times and observed strong interest among the high school participants. In addition, we are providing a publicly accessible website that shows the complete lecture program and collections of works created by the high school students.

## INTRODUCTION

Mathematics education is important in all modern civilizations. In Japan, high school students make reasonably high achievements in this subject. However, some aspects of mathematics education need further improvement. Overall, Japanese students performed well according to a 2006 study conducted by the Programme for International Student Assessment (PISA)[1].

However, according to the same report, Japanese high school students are less self-confident using of mathematics in their future careers. This is not because they do not understand the importance of mathematics. Indeed, the students are strongly motivated to study mathematics, and there is a high demand in Japanese industry to hire students with good mathematical skills.

One mission of the Dept. of Applied Computer Science of the Faculty of Engineering, Tokyo Polytechnic University is to provide excellent education for students seeking to enter the Japanese computer industry. Mathematics is one of the important skill for software engineers working in computer industry. With this reason, the department also encourages high school students to study mathematics.

The department has proactively used information systems for education since its inception[2]. In addition, the author performs research aimed at achieving an effective approach to education using information technology in various ways[3][4][5].

High school students may lack a clear understanding of when and why their knowledge will be valuable and necessary. At this stage of education, their learning experiences are not driven by strong passions.

To encourage high school students to develop a passion for studying mathematics, we held an experimental workshop in 2008. The workshop promoted the study of mathematics through music synthesis. The author participated in the standardization of MPEG-4 Structured audio[6], which is based on Csound[8]. Both of those are software standard to develop a computer music and author intensively worked for computer music technology at that time. Those experiences in that standardization project also helped this development.

The workshop courses were offered eight times. The participants included both high school and university students. Most of them showed a strong interest and increased passion for mathematics following participation in the workshop. We believe that the workshop had a notable effect on their motivation to learn about mathematics and computers.

## BASIC CONCEPT

To select the tool used in the workshop, we considered the following three criteria.

- (1) Students can create music and enjoy the workshop.
- (2) By participating in the workshop, students can learn how mathematical computations are used.
- (3) Students learn how computers synthesize music.

Table 1 shows the possible tool choices. Among the various tools that synthesize music are DTM software, Common Lisp Music[7], and Csound[8]. However, we rejected these possibilities because, even though each one is able to synthesize music,

---

\* Associate Professor, Applied Computer Science, Tokyo Polytechnic University  
Received Sept. 29, 2014

it is difficult to explain the calculations used by the system. The tool thus appears as a “black box” to students; it does not allow them to understand how music signals can be synthesized using mathematics.

Another possibility was to use a programming language and some built in functions such as MIDI-synthesizer APIs. However, we also rejected this choice, because then the mathematical process that generates sound would be hidden in the MIDI library.

Table 1: Comparison of various tools.

Tool	Pros	Cons
DTM software	Rich sound and easy-to-use interface.	Student can use the functions but cannot see the relation between the function and mathematics. Student cannot extend the function.
Lisp+CLM	Common Lisp based. Very powerful.	Too difficult to teach Lisp in a very short time, e.g., 4 hours.
C-sound	C-like language. Very powerful.	Too hard to teach C language in very short time, e.g., 4 hours.
Various programming languages	Theoretically possible to synthesize any kind of music.	Too difficult and time-consuming to teach a programming language to high school students.
Programming language with MIDI	MIDI makes sound synthesis easier.	Most parts of the sound synthesis are hidden to students.
MATLAB or Scilab	Possible to create music in a 10–20 line program	Difficult to write a program to synthesize music.

The only remaining choice was to use a programming language to write programs that synthesize sound and music, which is MATLAB or Scilab. In our opinion, those two has almost equivalent function. And we chose Scilab because it is Freeware so that student can use Scilab at home. After choice of the programming language, next question was how complex the program would be and whether it would be possible to teach the program to high school students in a few hours.

## LECTURE

The lecture consists of two parts and runs for a total of six hours maximum. The first part explains the basic method of making signals, from simple tones to more complex music. The second part is the workshop, which allows the students to freely create their own music. The lecture uses Scilab for signal music synthesis.

Figure 1 shows a web page from the first part. The page animates the propagation of wave forms and explains the relations among wave lengths, speeds of wave, and frequencies.

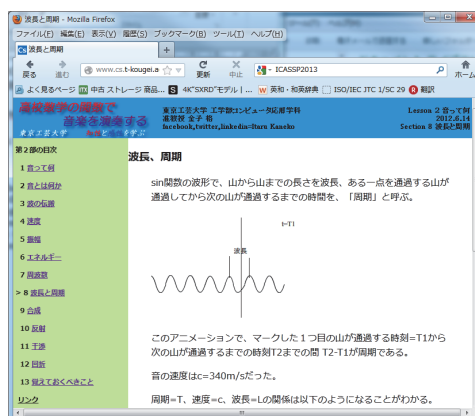


Figure 1: A page which shows an animated figure explaining wavelength and frequency.

Figure 2 shows the next step, Introduction of the sine function. After giving a brief explanation of sound, the lecturer immediately begins using Scilab to create the sound of the sine function and allows the students to create signals with different frequencies and loudness. By using mathematical expressions and by interactive experience, the students can quickly understand the relation between the expression and magnitude and frequency of generated sound.

Trigonometric functions (e.g., sin, cos ...) are already taught as part of the curriculum at a majority of Japanese high schools. Around 50% of students who participated in the workshop chose to attend the course that includes the sine function. However, it was necessary to give a basic explanation of this function.

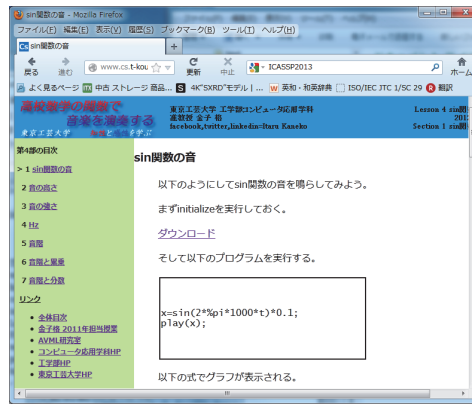


Figure 2: Introduction of the sine function and tone “A” using Scilab.

After teaching the basic usage of Scilab, the lecturer then explains how to generate a chromatic scale using Scilab, as shown in Figure 3.

“For” statements are used in these program. For high school student, all of the control statement, e.g. “For” loop statement and “If” conditional statement are difficult concept. However we believe that “For” statements are relatively easier to be understood because high school students already learned “series” which is included in the lecture of mathematics. “For” statements can be understood as alternative notation of accumulation of series or summation. Contrary to it, students need to understand clear concept of step-by-step execution and conditional execution, to understand “If” statements. For high school students who have no previous training of programming, control statements such as if – then – else are too difficult to be understood, especially within the very short time allotted for the workshop. Therefore, only “For” statements and matrices are introduced. These are relatively easy to understand because they can be explained as different forms of finite series. In our experience, high school students can understand and manage this level of programming, even with no programming background.

Next, the lecturer introduces an expression to map integers onto notes in the C-major scale. At first, simple program shown in Figure 9 which play chromatic scale is given as an example. The method to map integer “i”, which corresponds to the C-major scale, to another integer “n”, which corresponds chromatic scale, is shown in Figure 3. And the Scilab program to do this mappic is shown in Figure 10. Integer values “n” and “i”, shown in the Figure 3, corresponds to the variable having same letter in the program which is shown in the Figure 10.

The program uses only the “int” function, which is not difficult for high school students to understand.

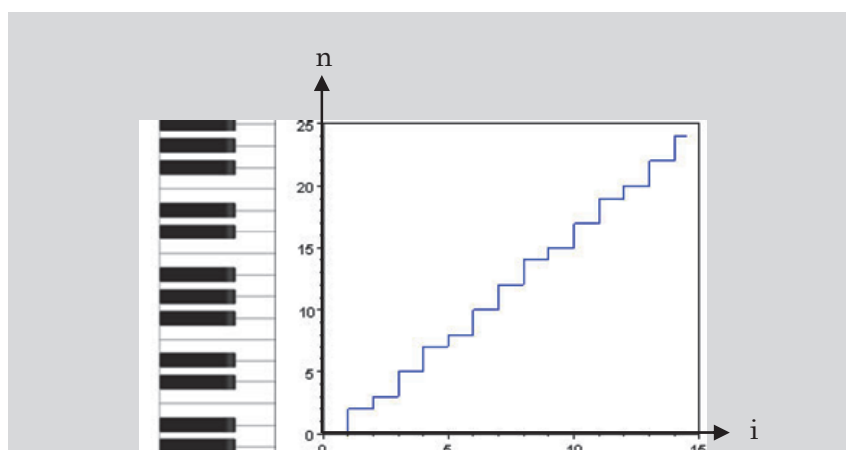


Figure 3: Introduction of the C-major scale. n and i corresponds to the variable in the program in Figure 12

After showing this example to the students, we also give some practices. For an example, by changing program to play 24 tone scales in 1000 beats per minutes, students can experience dramatically change of the music by the change in small part of the program. We give similar practice after showing all of following examples. Therefore, students can understand and confirm the relation between program and music.

After this, the students are given a simple program to generate two melodies and asked to combine them using an algorithmic method. The program in Figure 11 demonstrates how to play two voices. In this program, line 06 to 13 will produce one voice and line 14-20 will produce another voice. Because mathematical expressions to compute value  $n$  are different each other, this program generate two different melodies for two voices.

The students are now ready to learn about timbre. The lecturer provides sample programming to control the envelope using the sine function and exponential function, and also explains the expression for the modulation of frequency. The combination of modulation and envelope is controlled in a very straightforward and effective way. This method is not advanced, but it is intended to teach students a simple and easy way to control timbre using mathematic expressions.

At the end of Part 1, five voice music syntheses are demonstrated in the course. Because everything is controlled numerically by mathematical expressions, the program to play five different timbres in five different melodies is quite simple and short. Usually, the students are surprised and excited to find that they understand the mathematical expressions and will listen attentively to the complex music played by such simple expressions.

Figure 13 shows an example of one such program. The music consists of five parts and is around 20 seconds long. Music can be synthesized in this way by combining the technique already presented. We emphasize that the program does not contain any complex expressions or program code, as these might be too difficult for high school students. Even using simple calculations, it is possible to create fairly complex music.

## WORKSHOP

After the lecture, the students attend a 2-hour workshop followed by a mini-concert session. After the mini-concert session, we ask the students for feedback on the workshop.

The student feedback is generally positive. It is rare that a student will complain about the difficulty of writing mathematical equations to create music content. The students spend one hour learning how to write music using mathematical expressions. From there, they gradually improve their expressions to make the music better.

Although each program uses a similar basic structure, the resulting sounds are all unique and creative.

Readers may listen to the music created by the high school students during the workshop at the following URL. The page is also shown in Figure 4

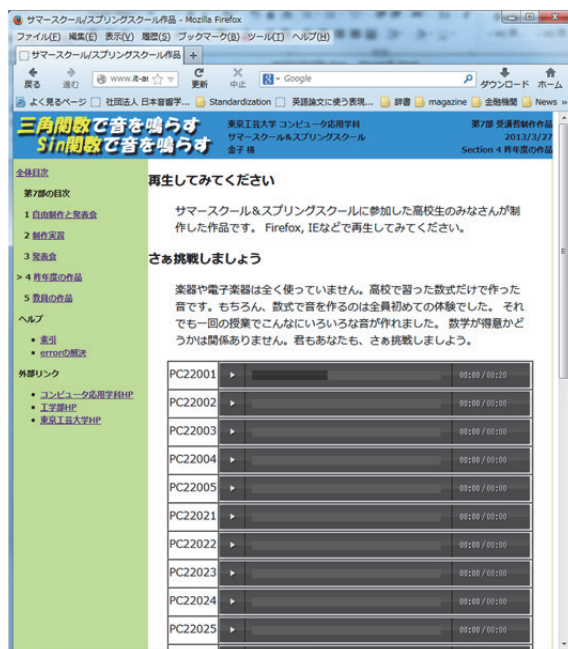


Figure 4: Student works created using Scilab. Usually, all the students can create unique music sounds using Scilab. The web page is publicly accessible at <http://www.it-aru.com/x/11/summerschool/7/>.

## RESULTS

We used questionnaire to evaluate effectiveness of our workshop. Figure 5 shows the questionnaire given to the high school students. The questionnaire consists of five questions. The students were asked to mark one of five levels according to their level of support to the opinion in each subject. The questionnaire asked to the participants twice in the each workshop, before and after the workshops. Since 2008, we periodically repeat our workshop and questionnaire and accumulate the results. Total number of participants is more than 50.

Figure 6 shows the results of the questionnaire. The inquiry compares the participants' answers to the same questions answered before and after the workshop. After the workshop, their interest in music and workshop is clearly increased. However, number of sample is around 8, which is still not sufficient for statistical validation.

Figure 7 shows the questionnaire results for the undergraduates. Their interest in mathematics greatly improved following the workshop.

The high school students' motivation to study mathematics was already high before the workshop, because math skills are essential for being accepted to a good university. However, the undergraduates in computer science courses had low initial motivation to study mathematics, because math skills are not very important in many of computer science course lectures. For an example, higher order equations are not essential to understanding programming. Therefore, it is understandable that the undergraduates showed more improvements in response to this question.

However, because either interest in mathematics or workshop is clearly increased following the workshop, we believe the workshop was received positively by the participants. Even though the high school students did not increase their interest in mathematics, we believe they found another reason to study it.

We also tried shorter version of work shop includes 20 minutes 20 lectures and 10 minutes exercises. Interestingly we could better result for the all of the subjects as shown in Figure 8.

## CONCLUSION

In this paper, we described an experimental workshop that encourages high school students to use mathematics to synthesize musical sounds and music. It is feasible and practical to give the proposed workshop in two hours. Scilab is an appropriate tool for the workshop activities. In addition, it is also feasible to let students create their own music. For many students, the workshop is their first experience running such massive calculations, e.g million times of multiplication and addition. Thus, the workshop activities will help them understand how such massive number of computation will result some useful results (e.g. music synthesis).

There are many ways to encourage young people to study mathematics. Obviously, music is not a very important target compared to other industrial applications. However, using music for education is also beneficial because it stimulates students' interest.

	No← level →Yes				
	1	2	3	4	5
Interest in mathematics					
Interest in computer music					
Interest in workshop					
Difficulty of workshop					
Interest in programing					

Figure 5: The questionnaire used to evaluate effect of workshop.

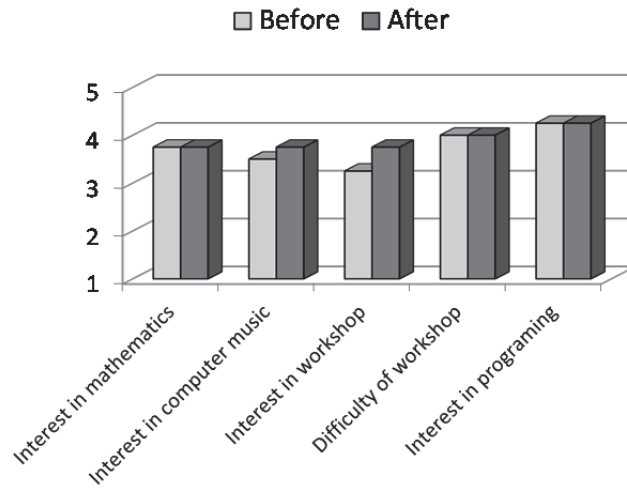


Figure 6: Questionnaire responses of the high school students.

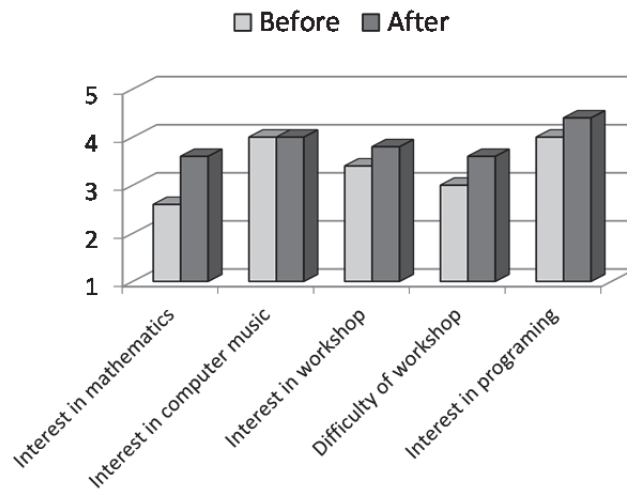


Figure 7: Questionnaire responses of the undergraduate students.

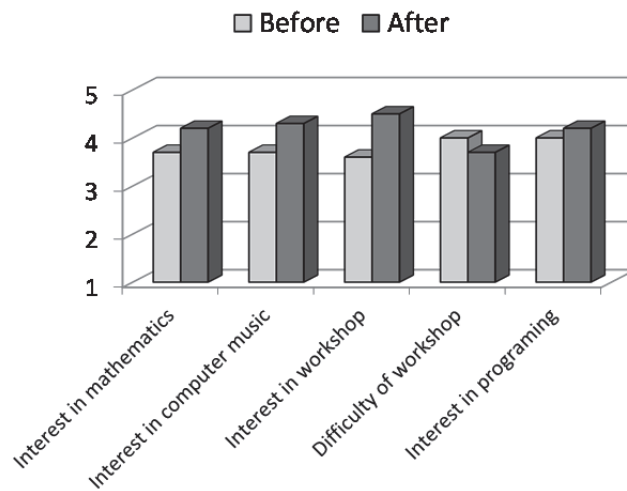


Figure 8: Questionnaire responses of the high school students participated to shorter version of workshop.

## ACKNOWLEDGEMENT

We thank Tokyo Polytechnic University very much for providing a facility and enabling us to hold this workshop. We also thank all of students who participated in the workshop.

## References

- [1] OECD, 2011. PISA - Quality time for students: Learning in and out of school, OECD Publishing, France
- [2] Sone, J. et al., 2006. Construction of the e-learning system in Tokyo Polytechnic University, The academic reports, the Faculty of Engineering, Tokyo Institute of Polytechnic, 29(1), pp.39-42.
- [3] Kaneko, I., 2008. Consideration of a future educational information system for specialized education, The academic reports, the Faculty of Engineering, Tokyo Institute of Polytechnic, 31, pp.107-117.
- [4] Kaneko, I., 2011. Music synthesis using high-school mathematics for experiential learning in computer programming [in Japanese], Forum on Information Technology 2011, pp.299-300.
- [5] Kaneko, I., 2012. Creating music using the trigonometric function, <http://www.cs.t-kougei.ac.jp/av-media/lectures/2012/cs/x/summerschool/7/page050.html>.
- [6] Scheirer, E.D., 1998. THE MPEG-4 STRUCTURED AUDIO STANDARD, Proc. IEEE ICASSP 1998, vol. 6, pp 3801-3804.
- [7] Taube, H. 1991. Common Lisp Music: A music composition language in Common Lisp and CLOS, Computer Music Journal, Vol. 15, No. 2, Summer, pp.21-32, MIT Press.
- [8] Vercoe, B. 1986. Csound Manual, <http://www.csounds.com/manual/html/>.

## Appendix : Examples of Scilab program for music synthesis

```

01 // Program 1: Chromatic scale
02 stacksize(1e7);
03 fs=44100;x=zeros(1:20*fs);
04 w=0:4410; t=w/fs;
05 for n=0:12
06     p=n*fs/4+w+1;
07     f=440*2^(n/12);
08     a=0.1;
09     x(p)=x(p)+sin(2*pi*f*t)*a;
10 end
11 playsnd(x, fs);

```

Figure 9: The Scilab program for a chromatic scale.

```

01 // Program 2: C-Major scale
02 stacksize(1e7);
03 fs=44100;x=zeros(1:20*fs);
04 w=0:4410; t=w/fs;
05 for i=0:14
06     p=i*fs/4+w+1;
07     n=int((i+2)/7*12+.4);
08     f=440*2^(n/12);
09     a=0.1;
10     x(p)=x(p)+sin(2*pi*f*t)*a;
11 end
12 playsnd(x, fs);

```

Figure 10: C-Major scale.

```

01 // Program 3: Duet
02 stacksize(1e7);
03 fs=44100; x=zeros(1:20*fs);
04 w=0:4410;t=w/fs;
05 for i=0:64
06     // voice 1
07     p=int(i*fs/4)+w+1;
08     k=i-int(i/4)*4;
09     m=k*2;
10     n=int(m/7*12 + 0.4);
11     f=440*2^(n/12);
12     a=0.1;
13     x(p)=x(p)+sin(2*pi*f*t)*a;
14     // voice 2
15     k=i-int(i/8)*8;
16     m=int(k/2)*3+7;
17     n=int(m/7*12 + 0.4);
18     f=440*2^(n/12);
19     a=0.1;
20     x(p)=x(p)+sin(2*pi*f*t)*a;
21 end
22 playsnd(x,fs);

```

Figure 11: Program 4 - Two voices.

```

01 //Program 4:Different timbres
02 fs=44100; x=zeros(1:20*fs);
03 w=0:fs; t=w/fs;
04 a=(1-exp(-t/0.01)).*exp(-t/0.3)*.2;
05 for i=0:63
06     // timbre 1
07     p=int(i*fs/6)+w+1;
08     k=i-int(i/4)*4;
09     j=k*2;
10     n=int(j/7*12 + 0.4);
11     f=440*2^(n/12);
12     m=sin(2*pi*f*t).*sin(20*t)*2;
13     x(p)=x(p)+a.*sin(2*pi*f*t+m)*.2;
14     // timbre 2
15     k=i-int(i/8)*8;
16     j=int(k/2)*3+7;
17     n=int(j/7*12 + 0.4);
18     f=440*2^(n/12);
19     m=sin(2*pi*f*t).*sin(20*t)*2;
20     x(p)=x(p)+a.*sin(2*pi*f*t+m);
21 end
22 playsnd(x,fs);

```

Figure 12: Program 5 – Different timbres.



```
01 // Program 5: Quintet
02 stacksize(1e7);
03 fs=44100; x=zeros(1:20*fs);
04 w=0:fs; t=w/fs;
05 a=(1-exp(-t/0.001)).*exp(-t/0.3)*.1;
06 for b=0:95
07     for v=1:5
08         p=int(b*fs/6)+w+1;
09         j=b-int(b/4/v)*4*v;
10         k=int(j/v)*(2+v)-v*7+14;
11         n=int(k/7*12 + 0.4);
12         f=440*2^(n/12);
13         m=sin(2*pi*f*t).*sin(20*t)*v;
14         x(p)=x(p)+a.*sin(2*pi*f*t+m);
15     end
16 end
17 playsnd(x, fs);
18
```

Figure 13: Quintet.