

Mathematica-aided Education of Science-major Students

K. Nakagami, F. Takeuchi, F. Ushitaki and M. Yasugi
Faculty of Science, Kyoto Sangyo University, Kyoto, Japan
<http://www.kyoto-su.ac.jp/~yasugi>

We will make a comprehensive report of our experience with Mathematica-aided education. Some students belonging either to Mathematics or Computer Science Department do graduation works using Mathematica as a language. We take up three subjects in those works.

■ Helping students to understand mathematics.

We have seen there are three ways of helping students to understand mathematics by Mathematica: (1) graphics, (2) reading and writing programs, and (3) experiments.

(1) The converging process of intervals to Cantor set and linear transforms of figures in the plane and the space by matrices and an initial value problem using the conservation law: With graphics, one can obtain a vivid picture of a limiting process, can get an idea of what a matrix is and dependence of a n initial value on conservation quantity.

(2) Mathematical property of Cantor set and the condition of a Mersenne number to be a prime: By reading and writing programs, which gives procedural aspect of the mathematical notions, students understand easily those properties and proofs.

(3) The fractional expression of left out points of Cantor set: We present a conjecture on the form of such fractions, and let the students experiment on it. They then wrote a program of "quasi Cantor set" (dividing the unit interval into four equal parts and eliminating the second and the fourth subintervals).

■ Making teaching materials.

Students can also help highschool students and freshmen at the university by inventing teaching materials with Mathematica. Graphics in linear algebra as in I above can be used also for this purpose, and a systematic drill for matrix computations is being programmed. In order to help understanding of quadratic functions, for example, one can let children draw a picture of a face with three parabolas. Through such works, with trial and error, they learn some properties of quadratic functions such as the fact that the shape of a parabola is determined by the coefficient of its highest order. In those trials, non-trivial devices were required in coloring by parts and solving inequalities.

■ Programming to solve puzzles and games.

We let students solve logical puzzles or make up strategies for winning a game by Mathematica. The principal objective of this activity is to help them analyze the puzzle or game, and materialize the algorithm in a relatively transparent way using Mathematica.

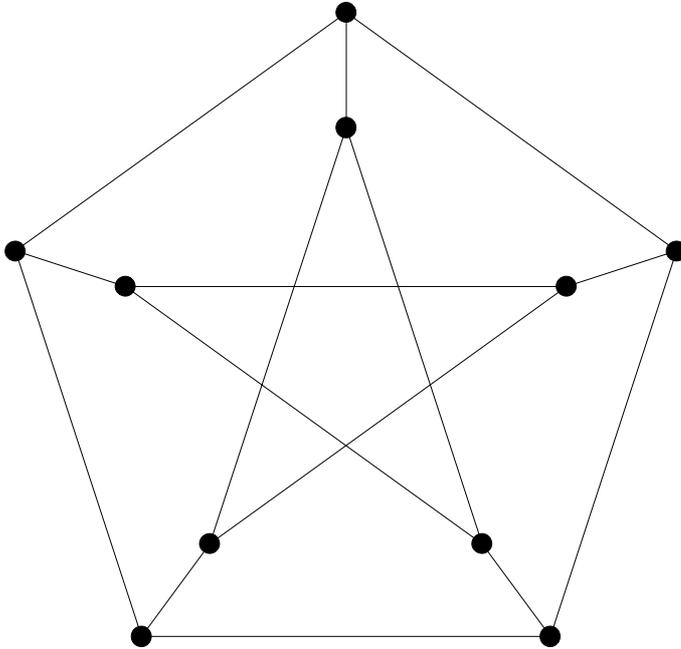
After one year of training of using Mathematica as a programming language, each student chooses a puzzle or a game of appropriate difficulty for him/her to study for another year, assisted by the teacher.

The advantages of Mathematica as a tool are, its well thought-out structure as a programming language and the pattern matching feature of the language which is extremely useful for solving this kind of problems. Students are often amazed by the simplicity of the coding using this feature.

We would like to present here, as an example, two subjects chosen in the past by students. The first one is a game and the second a puzzle.

Yashima-game

Two players play alternately. On a sheet of paper, the players draw arbitrary number of nodes as points (meaning islands), and then connect arbitrarily the nodes by straight lines (meaning bridges). An example is shown in the following figure.



After that, each player chooses his/her own starting node, and put a bead on those nodes. First player now moves from his/her node to another along one of the straight lines, and then deletes the strait line used (burns the bridge he/she just crossed). Second player does the same. The objective of the game is to make the opponent frozen in an island so that no straight line is existing from the node where the opponent is.

This game, although simple, is fairly amusing when played. To analyze this game, the student who chose this subject has given a unique name to each node. Then she made a matrix of N by N whose elements are either True or False. N is the number of nodes, and the element True means two nodes are still connected with a straight line.

Then she made a program using the backtrack algorithm. She has started from the Backtrack function in the Skiena's book. She changed the function such that the length of the total solution is not a fixed number. She could then tell that given the initial board, she can win even the opponent plays flawlessly, or she can loose if the opponent plays flawlessly.

In solving this problem with Mathematica, she was impressed by the flexibility of Mathematica. The number of nodes as well as the number of lines can be anything. Also she could follow the movement of backtracking function at work in a very transparent way. Unless the board is very complicated, the speed of Mathematica is enough so that it is fun to play against the computer.

Nonogram

The second example we show here is a puzzle called Nonogram. A very simple example is shown in the next figure. This example is entitled as Puppy.

problem

3					
2					
4					
3					
1 . 1					
	2	5	1	3	1
			2		

solution

3	■	■	■	■	■
2	■	■	■	■	■
4	■	■	■	■	■
3	■	■	■	■	■
1 . 1	■	■	■	■	■
	2	5	1	3	1
			2		

A rectangle of 5 by 5 squares is surrounded by series of numbers called hints. The player should either draw in a dot in the square or paint it black. The hint like 1 . 1 in the bottom row means in that row of two non-adjacent single squares somewhere should be painted black. The dot in the hints means that those painted squares are non adjacent. The hint of the left-most column is just 2. This means in this column, somewhere two consecutive squares should be painted black, and the rest (3 squares) should carry each a dot.

The rule is simple. Problems are very carefully prepared so that the player can reach the unique final solutions without cut-and-try method.

The real problems are a little more complicated with larger number of squares. We show here one of the problems which is entitles as Crawl.

■ Acquiring mathematical concepts and skills

In this section, we report some effects of graduation works by students of the third year and the fourth year. They do the work in continuation for two years. Our students, being computer students, do not have much chance of acquiring mathematical background, but with some help of Mathematica, they were able to make progress without pain.

Mersenne numbers:4 students

By reading a program of finding Mersenne numbers, the students learned elements of number theory.

This work was motivated by experiments on the distribution of prime numbers in the set of natural numbers. By using "ListPlot" one can plot the distribution of prime numbers up to the

100th, 1000th, 100000th, ...

primes, and see that the frequency decreases. The students, on the other hand, know that there are infinitely many prime numbers by the method of Euclid.

They then became interested in possible algorithms of finding large prime numbers, and found a well-known program named "MersenneExponentQ" in a textbook. It is no easy job to justify the "Do" part of the program, which stands

```
Do[r=Mod[r2-2,N],{p-2}]
```

where the program tries to judge whether $2^p - 1$ is a prime number for a prime number p . The output is "True" if $r=0$ when the "Do" part is repeated $(p-2)$ times, and is "False" otherwise.

Upon suggestion in the text, they began to challenge the proof of an old theorem (1935) by D.H.Lehmer, in Journal of the London Mathematical

Society. It is a short article which assumes deep knowledge in number theory, and so, in order to give a detailed proof, the students first studied elements of number theory and then completed the proof.

Lehmer's theorem states as follows.

For any prime number n , put

$$N=2^n - 1, S_1 = 4, S_k = S_{k-1}^2 - 2$$

Then N divides S_{n-1} if and only if N is a prime number.

In the program, the condition that N divides S_{n-1} is reduced to a computation of $\text{Mod}[S_{n-1}]$ modulo $2^n - 1$. The students were able to mathematically justify this reduction of computation.

All in all, starting with experiments with plotting, the students found pleasure in number theory and drew out their intelligence for it. They also learned that an old mathematical theorem can contribute to programming a mathematical algorithm, and that a mathematical theorem has to be adjusted to efficient computations in writing a program.

Cantor set:6 students

The purpose of assigning this theme to the students was to let them understand ternary expressions of real numbers as well as the limiting process of sets.

By plotting the construction process of the Cantor set using graphics, and doing experiments, the students were able to prove some mathematical properties of the points removed and the points unremoved as well as the length and cardinality of the Cantor set.

Let $J_{2,2}$ denote the second removed subinterval at the second step.

$$x \in J_{2,2} \iff x = 2/3 + 1/3^2 + \sum_{n=3}^{\infty} \epsilon_n / 3^n < 2/3 + 2/3^2$$

where $\varepsilon_i = 0,1,2$, is but one example. (Before the experiments, my lecture had passed over their heads!) The students next attempted to write a program of the following. Instead of dividing into three parts, divide the closed interval $[0,1]$ into four equal parts and remove the second and the fourth subintervals; repeat the same process to all the remaining subintervals. They were able to set up some propositions on the points in the intervals and characterize a Cantor-like set thus created.

With programming of the construction process of this set, they found one snag, that is, in this case, one has to refer to the interval $[3/4,1]$. Mathematica regards 1 as "Integer", but not as "Rational", and a uniform expression of the removed intervals as in the case of the Cantor set breaks down. One has to make this initial step as an extra case. In spite of lack of training in the limit processes, especially of that of sets, the students thus understood the essence of the Cantor set stimulated by a programming of the Cantor-like set.

Educational materials by Mathematica:8 students

Those students wanted to take the advantage of Mathematica in helping highschool and freshman level students in linear algebra by presenting experimental tools and self-training tools by Mathematica. They have written the following programs.

Showing the effect of a matrix or a successive application of matrices by animation. The motion of a point or a set of points in the plane when applied a matrix or several matrices in succession is animated, so that a learner will understand what kind of transformation a matrix does to the plane.

Self learning kits for matrix translations and solving simultaneous linear equations.

This project aimed at making self-learning kits on translations of two by two or three by three matrices and solving simultaneous linear equations. The program first creates randomly matrices either in integers or in reals and then let a learner choose a problem. If a student does the problem, then the program will output "True" or "False" and, if "False" then a correct answer is output.

The students not only came up with many ideas on programming, but also learned linear algebra a lot better than before. They also became deeply concerned with what be good educational materials. We would like to do experiments on the effect of learning by using these kits with highschool students and freshmen.

On initial value problem of the KdV-equation and Mathematica

We consider the KdV equation

$$u_t + 6u_x u + u_{xxx} = 0, \quad (1)$$

which is known as a nonlinear differential equation with solitary wave solutions. We rewrite equation (1) in the form

$$u_t + u_x (3u^2 - u_{xx}) = 0. \quad (2)$$

Under the assumption that $u, u_x, u_{xx}, u_{xxx}, \dots \rightarrow 0$ as $|x| \rightarrow \infty$, we integrate both sides of (2) with respect to x . Then we can see that the integral $I_1 = \int u dx$

is a time independent quantity. Thus the KdV equation has a conservation law.

Similarly multiplying u to both sides of (2), we can see that the integral

$I_2 \int u^2 dx$ is the time independent quantity. This is also a conservation law. We can repeat the similar argument to find more conservation laws. The integral I_n is called the n-th order conserved quantity. The solitary-wave solution is of the form

$$u(x, t) = A \operatorname{sech}^2 \sqrt{A/2} (x - 2At - x_0), \quad (3)$$

where A is an amplitude. Thus the solitary-wave solution is determined by its amplitude A except for the phase x_0 .

Suppose that the initial wave function is given by

$$u(x, t = 0) = A_0 \operatorname{sech}^2(x/\lambda). \quad (4)$$

In the following, we restrict ourselves to the case where $0 < A < 2$. Then the initial wave function tends to a solitary wave solution as $t \rightarrow \infty$. For each conserved quantity I_n , we can compute the amplitude A of the solitary wave solution. Each conserved quantity I_n for the initial wave function and the solitary wave solution coincide. Hence we have

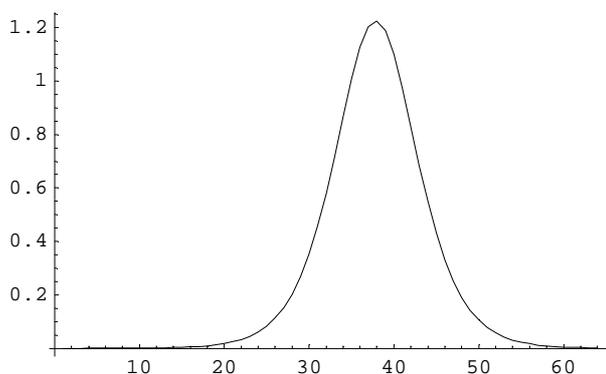
$A = (A_0)^2/2$ from the conserved quantity I_1 . Similarly we have $A = (A_0^2 \sqrt{2})^{2/3}$ and

$$A = \left\{ \sqrt{2} (2 A_0^3 - A_0^2/\lambda) / 3 \right\}^{2/5} \quad (5)$$

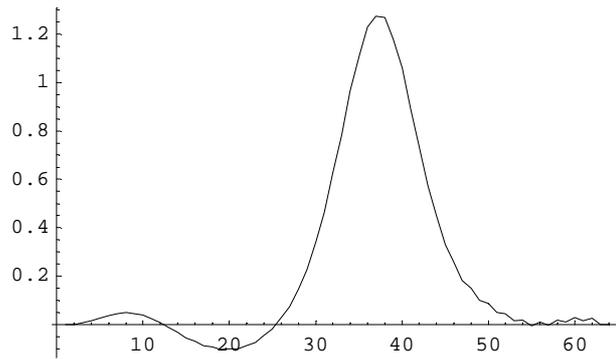
from the conserved quantity I_2 and I_3 , respectively. However, in order to get more precise amplitude A , it is better to use higher order conserved quantity. Now, we will use the amplitude A obtained by formula (5), and make a graph, called by Graphics1, of the solitary wave solution with the initial wave function

$$u(x, t = 0) = 1.4 \operatorname{sech}^2 x. \quad (6)$$

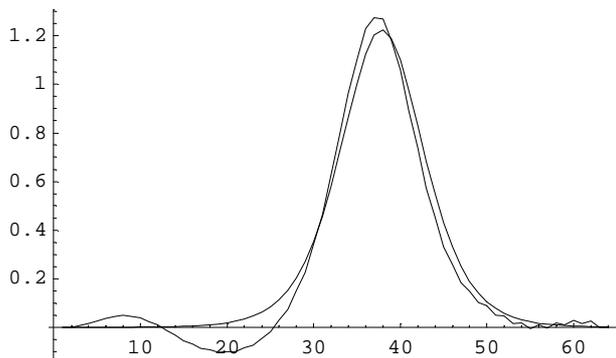
On the other hand, by using the difference method, we obtain a numerical solution of the KdV-equation (1) with the initial wave function (6) and make a graph, called by Graphics2, of the approximating solution. These two graphs for the solitary wave solution almost coincide. In such a way, we succeed in making students understand visually that the initial value problem can be solved by making use of the conservation laws.



Graphics1



Graphics2



Graphics1, Graphics2

Teaching parabola by drawing faces

Since it is quite easy to plot the graphs of functions with *Mathematica* in teaching mathematics, we can use it for drawing some figures of curves or surfaces which are difficult to draw by ourselves. As a result of such drawings, students learn various properties of the functions, and it enables them to comprehend some important concepts in mathematics.

On the other hand, *Mathematica* graphics is also useful in teaching elementary mathematics. In this section, we propose a device for it. Of course, it is meaningless to plot the graphs of functions given as exercises in a text-book. Our idea is to let highschool students draw some simple pictures on the coordinate plane by arranging some parts of the graphs of elementary functions.

They become engrossed in this kind of work because it is a creative work. They thus become familiar with some mathematical notions without even realizing it.

As an example, we will explain how to use *Mathematica* for teaching the theory of quadratic functions and parabolas. With our device, students can help highschool children to understand essence of these functions.

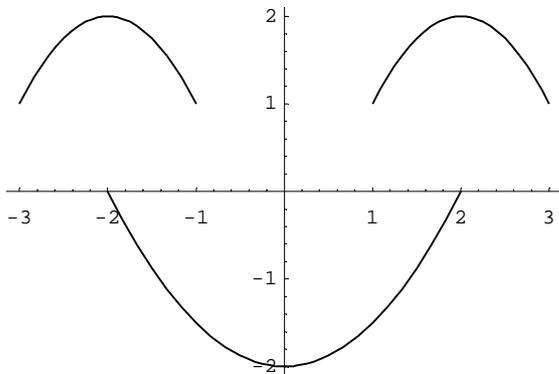
Quadratic function is one of the most basic functions for learners of mathematics. Its graph parabola is also important. However, when learning it in school, they must plot a lot of graphs from given functions or conditions, or solve various exercises in order to acquire basic properties of such a function. Instead of such a traditional method, let us use *Mathematica*.

Only by drawing pictures of human faces with three parabolas as follows, they can learn most of the properties of quadratic functions and parabolas.

```

righteye = Plot[-(x - 2)^2 + 2, {x, 1, 3}];
lefteye = Plot[-(x + 2)^2 + 2, {x, -3, -1}];
mouth = Plot[1/2 x^2 - 2, {x, -2, 2}];
Show[{righteye, lefteye, mouth}, AspectRatio -> Automatic]

```

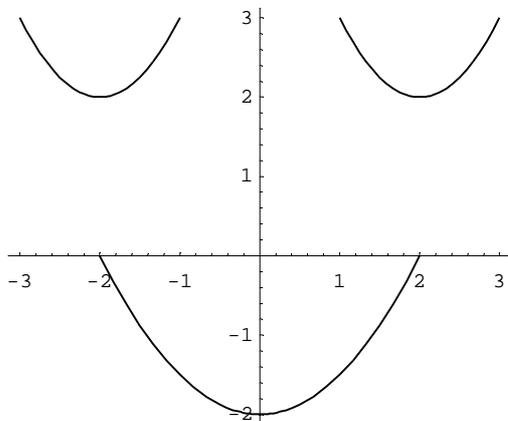


Of course, it is difficult for highschool students to code like this. It is necessary for a teacher or a tutor to show and explain the program written in *Mathematica*. After explanation, let them change the coefficients in the functions one by one as in the following example. They observe how the graph changes accordingly. Through such works, with trial and error, they discover some characteristics of quadratic functions such as the fact that the shape of a parabola is determined by the coefficient of its highest order.

```

righteye = Plot[(x - 2)^2 + 2, {x, 1, 3}];
lefteye = Plot[(x + 2)^2 + 2, {x, -3, -1}];
mouth = Plot[1/2 x^2 - 2, {x, -2, 2}];
Show[{righteye, lefteye, mouth}, AspectRatio -> Automatic]

```



If a highschool student desires to plot a more complicated picture, he/she needs to study both mathematics and *Mathematica*. However, he/she does it because he/she wants to complete the work. For example, in the process of drawing the following picture, he/she learns the relation between two parabolas, how to compute the coordinate of the intersection point of two parabolas, and the meaning of the region which is surrounded by several curves. Moreover, if he/she thinks that the outline of the face does not look like the human one, he/she is invited to study more advanced functions.

```
Solve[-5 (x-3/2)^2 + 3 == (x-2)^2 + 7/4, x]
```

```

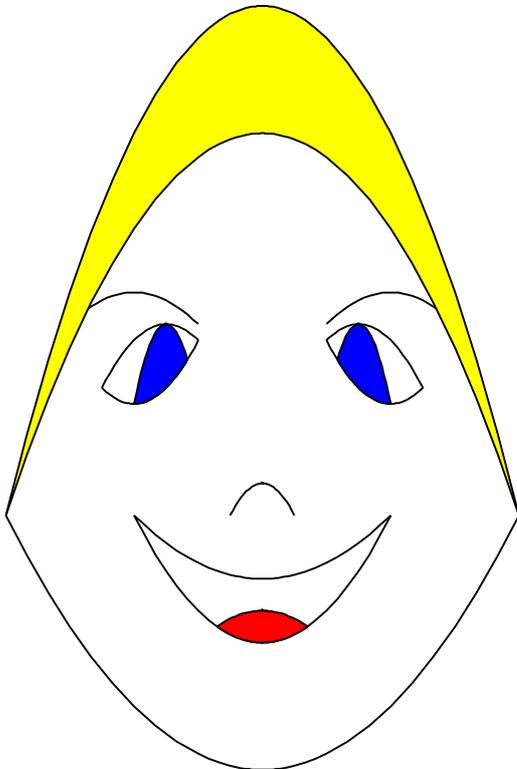
7
{{x -> -}, {x -> 2}}
6

```

```
Solve[1/2 x^2 - 2 == -1/2 x^2 - 3/2]
```

```
{{x -> -(1/Sqrt[2])}, {x -> 1/Sqrt[2]}}
```

```
Needs["Graphics`FilledPlot`"];
hair=FilledPlot[{-1/2 x^2 + 8,-3/8 x^2 + 6}, {x, -4, 4},
  Fills->Hue[1/6]];
righteye1=Plot[(x-2)^2 + 7/4, {x, 1,5/2}];
righteye2=Plot[-(x-3/2)^2 + 3, {x, 1,5/2}];
righteyeball=FilledPlot[{-5 (x-3/2)^2 + 3, (x-2)^2 + 7/4},{x,7/6,2},
  Fills->Hue[2/3]];
righteyebrow=Plot[-1/2 (x-2)^2 + 7/2, {x, 1,3}];
lefteye1=Plot[(x+2)^2 + 7/4, {x, -5/2,-1}];
lefteye2=Plot[-(x+3/2)^2 + 3, {x, -5/2,-1}];
lefteyeball=FilledPlot[{-5 (x+3/2)^2 + 3, (x+2)^2 + 7/4},{x,-2,-7/6},
  Fills->Hue[2/3]];
lefteyebrow=Plot[-1/2 (x+2)^2 + 7/2, {x, -3,-1}];
nose=Plot[-2 x^2 + 1/2, {x, -1/2, 1/2}];
mouth1=Plot[1/2 x^2 - 2, {x, -2, 2}];
mouth2=Plot[1/4 x^2 - 1, {x, -2, 2}];
tongue=FilledPlot[{-1/2 x^2 - 3/2, 1/2 x^2 - 2},
  {x, -1/Sqrt[2], 1/Sqrt[2]},Fills->Hue[0]];
chin=Plot[1/4 x^2 - 4, {x, -4, 4}];
Show[{righteye1, righteye2, righteyeball, righteyebrow,
  lefteye1, lefteye2, lefteyeball, lefteyebrow,
  nose, mouth1, mouth2,tongue, hair,chin},
  AspectRatio->Automatic,
  PlotRange->All,Axes->False]
```



We believe that a student gets interested in mathematics through such works.