

University for Business and Technology in Kosovo

UBT Knowledge Center

UBT International Conference

2016 UBT International Conference

Oct 28th, 9:00 AM - Oct 30th, 5:00 PM

Traditional mathematics and new methods of teaching through programming together with students

Robert Kosova

Aleksander Moisiu University, robertko60@yahoo.com

Teuta Thanasi

Aleksander Moisiu University, teutamatematika@hotmail.com

Lindita Mukli

Aleksander Moisiu University, linditamukli@gmail.com

Loreta Nakuçi Pëllumbi

Harry T. Fultz Institute, loreta.nakuci@gmail.com

Follow this and additional works at: <https://knowledgecenter.ubt-uni.net/conference>



Part of the [Databases and Information Systems Commons](#), and the [Information Security Commons](#)

Recommended Citation

Kosova, Robert; Thanasi, Teuta; Mukli, Lindita; and Pëllumbi, Loreta Nakuçi, "Traditional mathematics and new methods of teaching through programming together with students" (2016). *UBT International Conference*. 4.

<https://knowledgecenter.ubt-uni.net/conference/2016/all-events/4>

This Event is brought to you for free and open access by the Publication and Journals at UBT Knowledge Center. It has been accepted for inclusion in UBT International Conference by an authorized administrator of UBT Knowledge Center. For more information, please contact knowledge.center@ubt-uni.net.

Traditional mathematics and new methods of teaching through programming together with students

Robert Kosova¹, Teuta Thanasi², Lindita Mukli³, Loreta Nakuçi Pëllumbi⁴

^{1,2,3} University „Aleksander Moisiu“. Faculty of Information Technology
Department of Mathematics. Durrës. Albania

⁴ Institute „Harry T. Fultz“. Tirana. Albania.

{robertko60¹, teutamatematika²}@hotmail.com,
{linditamukli³, loreta.nakuci⁴}@gmail.com

Abstract. We are used to the traditional methods of teaching mathematics. The textbook, the blackboard and a chalk have been for centuries a wonderful part of teaching. And, they always will be. Traditional teaching methods of mathematics are a wonderful legacy of our educational system that have educated generations of teachers, engineers, administrators, managers, leaders, and economists. American universities websites, the video- lectures of the best professors of well-known disciplines such as statistics, operational research, number theory, algebra, game theory, show impressing large blackboards, all over the auditor's walls. We always will need and admire traditional mathematics. But, beyond the lessons, especially in applied disciplines, students need to learn the software created for discipline applications. Making IT students to learn and, more important, to create applications and programs following the theoretical lessons is a useful way to help them understand better the theoretical knowledge. Similarly, other “theoretical” math disciplines such as number theory, game theory, geometry can be used to. The idea of making a group of IT students and the professor to create a package of applications for the math courses is a very useful process to a more advanced stage of teaching.

Keywords: education, tradition, technology, mathematics, programming.

1. Introduction:

The information nowadays grows exponentially. School and university programs can't change so quickly. It is easier to learn from the internet or another media, than from the school. New technologies give us the information in such way, which is more attractive for the students. On the other hand, the school offers traditional methods in providing knowledge. They seem to be boring, difficulties, etc. The problem is: can we use the new technologies to increase the interest of the students about the scientific subjects? How can we include the students in the process of the teaching? How can we make them to feel better and more appreciated? We can use the computer in the electrical, electronic `s field in two general directions: computer languages or software packages. However, we can apply one of them, to do the analyses of an electrical circuit. For example, program languages like C++, C#, Java, or one of the software packages like PSpice, Multisim, MatLAB etc. For instant, if we want to explain

the solution of the differential equation second- order, which describes a circuit containing storing elements on it, we applied program language C++, C#, and Mat LAB. Everything was prepared by the students. This makes the student to learn more in the same time increase the reliability in work, self-confidence etc. On the other hand the analyses of the phenomena performed better and faster.

2. Solving Differential equations second-order.

The respond of a linear electrical circuit, which containing two storage elements, is described by differential equation second -order. Typical circuits are RLC circuits, (Resistor, Inductor and Capacitor). If we want to obtain the differential equation for a circuit than we can apply, Kirkoff's Law. We determine the response $x(t)$ (which may be voltage or current) by taking the following steps.

1. We first determine the initial conditions $x(0)$ and $x'(0)$
2. We find the natural response $x_n(t)$ by turning off independent sources, this means the right side of the equation becomes zero.
3. We determine its characteristic roots.
4. Depending on whether the response is over damped, critically damped, or under damped, we obtain $x_n(t)$ with two unknown constants.
5. We finally determine the constants associated with the natural response by imposing the initial conditions $x(0)$ and $x'(0)$, determined in step 1.

The differential equation second -order is given by expression:

$$a \frac{d^2 x_c}{dt^2} + b \frac{dx_c}{dt} + cx_c = f(t) \quad (1)$$

We substitute $d/dt = \alpha$, then we have: $a\alpha^2 + b\alpha + c = f(t)$ (2)

If $f(t) = 0$, then $a\alpha^2 + b\alpha + c = 0$, it is called characteristic equation for the circuit;

$$a\alpha^2 + b\alpha + c = 0 \quad (3)$$

The roots of the equation α_1, α_2 , determine the behavior of the circuit. These roots are nominated as natural frequency of the circuit.

There are three cases:

- a. $D > 0$, we have two different roots α_1, α_2 .
The solution is: $x_n(t) = A_1 e^{\alpha_1 t} + A_2 e^{\alpha_2 t}$ over damped.
- b. $D = 0$, we have two roots, which are the same $\alpha_1 = \alpha_2 = a$.
The solution is: $x_n(t) = (A_1 + A_2 t) e^{at}$ critically damped.
- c. $D < 0$, we have two complex roots, $\alpha = \delta \pm j\omega$
The solution is $x_n(t) = e^{\delta t} (A_1 \cos \omega t + A_2 \sin \omega t)$ underdamped. The sign of D is depended from the value of coefficients a, b, c .

The steps to solve the equation:

Let have $a, b, c, x_c(0), x'_c(0)$;

1. We calculate $D = b^2 - 4 * a * c$
2. We calculate the roots α_1, α_2
3. For $D > 0$, we have $\alpha_1 \neq \alpha_2$, $x_c(t) = A_1 e^{\alpha_1 t} + A_2 e^{\alpha_2 t}$
4. For $D = 0$, we have $\alpha_1 = \alpha_2$, $x_c(t) = (A_1 + A_2 t) e^{\alpha t}$
5. For $D < 0$, we have $\alpha = \delta \pm j\omega$, $x_c(t) = e^{\delta t} (A_1 \cos \omega t + A_2 \sin \omega t)$
6. We calculate the coefficients A_1, A_2 .

For $t = 0$ we have: $x_n(0) = A_1 + A_2$ and $x'_n(0) = \alpha_1 A_1 + \alpha_2 A_2$, $D > 0$,

3. Programming with C++ language.

```
#include <iostream>
#include <math.h>
#include <limits>
using namespace std;
namespace {
    unsignedconst e = 2.7182818284590452353602874713527;
};
int main()
{
    std::numeric_limits<double>::quiet_NaN();

    // inputet
    double a, b, c, x1, x2;
    cout<< "Jepvariablin A: ";
    cin>> a;
    cout<< "Jepvariablin B: ";
    cin>> b;
    cout<< "Jepvariablin C: ";
    cin>> c;
    cout<< "Jepvariablin X(0): ";
    cin>> x1;
    cout<< "Jepvariablin X'(0): ";
    cin>> x2;
    cout<< "\n";
}
```

Programming with C# language

```
using System;
using System.Windows.Forms;
using OxyPlot;
using OxyPlot.Series;
using System.IO;
using OxyPlot.WindowsForms;

namespace ElektronikeEk
{
    publicpartialclass Form1 :Form
    {
        publicconstdouble e = 2.7182818284590452353602874713527;

        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```
{
InitializeComponent();
}

private void TextBox_KeyPress(object sender, KeyPressEventArgs e)
{
    errorProvider1.Clear();
    TextBox txt = (TextBox)sender;
    // allows 0-9, backspace, and decimal
    if(!char.IsControl(e.KeyChar) && (!char.IsDigit(e.KeyChar)) && (e.KeyChar != '!') && (e.KeyChar
    != '-'))
    {
        errorProvider1.SetError(txt, "Only numeric characters allowed.");
        e.Handled = true;
        return;
    }

    if (e.KeyChar == '-' && !((sender as TextBox).SelectionStart == 0) && (sender
    as TextBox).Text.Contains("-"))
    {
        errorProvider1.SetError(txt, @"Minus only allowed at the begining.");
        e.Handled = true;
    }

    // checks to make sure only 1 decimal is allowed
    if (e.KeyChar == '.')
    {
        if (e.KeyChar == '.' && (sender as TextBox).Text.IndexOf('.') > -1)
        {
            errorProvider1.SetError((sender as TextBox), "Only one decimal point allowed.");
            e.Handled = true;
        }
    }
}

.....
// Plot the Graph
```



Figure 1. Plots of the solution of diff. eq. sec.-order.
Over damped, critically damped, underdamped.

4. Number Theory and Programming.

Number theory has a lot of interesting theories and proofs to work for. The figures show the calculation result of some programming such as;

1. Verify if number is prime or composite,
2. Find the prime numbers up to a given number,
3. Find Mersenne numbers, verify if they are prime or not,
4. Generate Pythagorean triples,
5. Generate twin primes,
6. Find LCM and GCD, etc.
7. Find all the divisors of a given number,
8. Find two numbers when you have their LCM and GCD, etc.

4.1. Triple Pythagorean Numbers.

A "Pythagorean Triple" is a set of positive integers a, b, c that fits the rule: $a^2 + b^2 = c^2$. There are an infinite such numbers. We can prove this with the help of the first Pythagorean Triple (3, 4, and 5), because the triples of forms $3n, 4n, 5n$ are also a set of Pythagorean Triple. This is true because: $(3n)^2 + (4n)^2 = (5n)^2$.

```
CHOISE LIST:
VERIFY A PRIME NUMBER = 1
FIND PRIME NUMBERS UP TO NUMBER = 2
FIND PRIME NUMBERS BETWEEN TWO NUMBERS = 3
FIND MERSENNE NUMBERS (2^n)-1 = 4
FIND PYTHAGORIAN TRIPLES = 5
FIND TWIN PRIMES UP TO A NUMBER = 6
FIND GCD AND LCM= 7
FIND TWO NUMBERS WHEN YOU HAVE GCD AND LCM= 8
END THE PROGRAM = 0
-----
ENTER A NUMBER FROM THE CHOISE LIST? 5
ENTER A NUMBER TO GENERATE PYTHAGORIAN TRIPLES? 200
PYTHAGORIAN TRIPLES UP TO THE NUMBER 200 ARE;
3, 4, 5
5, 12, 13
6, 8, 10
7, 24, 25
8, 15, 17
9, 12, 15
9, 40, 41
10, 24, 26
11, 60, 61
12, 16, 20
12, 35, 37
13, 84, 85
14, 48, 50
15, 20, 25
15, 36, 39
15, 112, 113
16, 30, 34
16, 63, 65
17, 144, 145
18, 24, 30
18, 80, 82
```

Figure. 2. Triple Pythagoreans numbers, generated by the program Just Basic.

4.2. The Fibonacci sequence;

The formula to generate the Fibonacci sequence is $F_n = F_{n-1} + F_{n-2}$, where;

F_n – is the term number "n",

F_{n-1} – is the term number "n – 1",

F_{n-2} – is the term number "n – 2",

The first 20 Fibonacci numbers are;

0,1,1,2,3,5,8,13,21,34,55,89, 144,233,377,610,987,1597,2594,4181,6765, ...

4.3. Mersenne numbers;

The Mersenne numbers are of form; $M_n = 2^n - 1$, for some integer n.As of January 2016, 49 Mersenne primes are known. The largest known prime number $2^{74\ 207\ 281} - 1$ is a Mersenne prime.

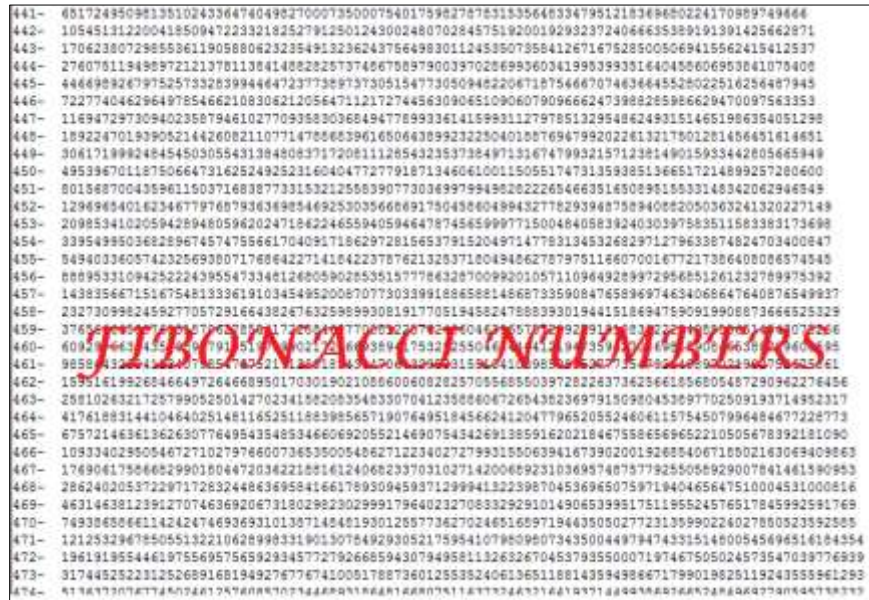


Figure 3. Fibonacci Numbers, generated by the program Just Basic.

Conclusions:

Students, working in groups, prepared the material, in C++, C#, Just Basic, the other calculated formulas in Mat LAB. Teacher and students discussed together and they observed that:

1. It is made clearer the effect of the value of a, b, c coefficient and the initial conditions $x(0), x'(0)$ to the solution of differential equation second –order. This solution represents the natural respond of a circuit containing storage elements. The values of coefficients a, b, c, are depended on circuit parameters R, L, C and how they are connected.
2. The students' interest is increased,
3. Other students' motivation is greater,
4. They can use better computer methods,
5. They can compare the computer methods,
6. They will be able to choose the computer method, etc.
7. Number theory problems and proofs become more interesting and attractive by programming, working with students.
8. Programming will always will be used to work with unsolved problems in number theory, such as; finding the largest prime number, finding the largest Mersenne number, etc.

References:

1. David E. Johnson, John L. Hilburn, Johnny R. Jonnson; "Basis Electric Circuit Analysis"; (Prentice Hall), 1980, fourth Edition, Inst.762.
2. Jeanine Meyer, Marty Lewinter." Elementary Number Theory with Programming. ISBN-10: 1119062674.
3. Robert. L. Boylestad; "Introductory circuit analysis". (Prentice Hall), twelve edition, 2010, Inst.1188.