

Introduction to the Infomath function in mathematics and its application in electrical and electronic engineering

CLAUDE ZIAD BAYEH^{1, 2}

¹Faculty of Engineering II, Lebanese University

²EGRDI transaction on mathematics (2002)

LEBANON

Email: claude_bayeh_cbegradi@hotmail.com

Abstract: -The *InfoMath Function* is a new function introduced by the author into mathematical domain in 2002 and it can describe an infinite number of functions, and can unify many expressions and formulae into only one expression or formula. It is introduced in the mathematical domain to facilitate and reduce many expressions and forms to a simple expression that has the same characteristics as the precedents. For example in the power electronics, one can join characteristics of the voltage applied on the resistor, capacitor and coil into only one expression without losing any data or any characteristic of these elements, this is not the case of any other function. The *InfoMath function* is simple to use and to introduce in any type of expressions as any other functions for example: cosine, Logarithm; exponential... In another hand it can replace any other function by varying some parameters. This function is composed into 3 parts: *input*, *condition* and *output*. By varying parameters of these parts, one can describe an infinite number of forms and functions. E.g: one can describe a square waveform, rectangular function, triangular function, pulse of Dirac, Hysteresis curve, sign function, angular function, rectangle, circle, ellipse, trigonometry functions and many others functions. The *input* can be a constant number, scalar, parameter, function, vector, variable, matrix ... The same for the *output*. The form of the *InfoMath function* is: *inputs; conditions; output*. One can have many inputs but in this study the author will show only the case of one input. On the other hand, only one output is defined. Finally the *InfoMath function* has enormous variety of applications in all scientific fields and especially in mathematics, electrical engineering, power electronics, physics and chemistry. In this paper, the author will discuss the application of the *InfoMath function* into the electrical and electronic domains. Many examples are given in order to have a good idea about how to use the *InfoMath function* in any domain.

Key-words: - Mathematics, General function, Unification, Logic, Computer language.

1 Introduction

Mathematics is often defined as the study of topics such as quantity, structure, space, and change. Another view, held by many mathematicians, is that mathematics is the body of knowledge justified by deductive reasoning, starting from axioms and definitions [3-18].

Practical mathematics, in nearly every society, is used for such purposes as accounting, measuring land, or predicting astronomical events. Mathematical discovery or research often involves discovering and cataloging patterns, without regard for application. The remarkable fact that the "purest" mathematics often turns out to have practical applications is what Eugene Wigner has called "the unreasonable effectiveness of mathematics". Today, the natural sciences, engineering, economics, and medicine depend heavily on new mathematical discoveries.

In the electrical and electronic engineering, the mathematics plays a big role in introducing new formulae and functions that are extremely important in producing new and usable signals as sinusoidal signal, rectangular signal, Dirac function and so on... Till now, scientists introduce new formulae when there is a leakage in demonstrating a certain phenomena in the nature or if the new introduced formulae will help them to create a new technology or a new concept. For example the step function plays a big role in the signal theory and power electronics in which it is used to create two states (0 and 1) or (On and Off), These states have big influence in developing the semiconductors which have the same principal (On and Off) of the step function.

Briefly, each introduced function or equation in mathematics can be used in electrical and electronic engineering especially when it is related to the time factor. In this paper, many of these functions are presented with their equivalent function into the *InfoMath function*. The *InfoMath function* is very

important in the electrical and electronic engineering domains because it can describe any other function or expression. Moreover, one can imagine the importance of one single function (Infomath function) that describe an infinite number of functions and expressions as Dirac function, Sign function, saturation function, Step function, Rectangular function...

Moreover, the author introduced new and original formulae that might be used in the future in the electrical and electronic domains. The importance of each new function introduced is to facilitate the manipulation of similar big number of formulae and expressions. For example, do you know that a single function can describe a chaotic Hysteresis curve? In reality it is impossible to do it in the traditional ways, but this mystery can be done using *Infomath function*. Some other important things to do are to unify many different characteristics in the nature such as resistor, capacitor and Coil into one formula. One can imagine that it would be impossible to join or unify many items with different characteristics in the nature into only one item or formula; this will be much easier to do by using the *Infomath function*.

In this paper, the *InfoMath function* is introduced by the author into the mathematical domain that can be categorized as new mathematical discovery or new mathematical invention of new function in which it has enormous application into all scientific domains [1-2] and [19-23]. The main goal of introducing this new function is to facilitate the manipulation of many functions, to unify many expressions and equations into only one function, and to simplify complicated expressions.

The *Infomath function* is a mathematical function with a computer language base, in which it works as computer software that receives inputs, analyses these inputs by conditions steps and finally gives an output.

Logical expressions are used to form a function that gives the answer and expression needed.

e.g.: $y = x; > 0; (1)/(0)$ is an *Infomath function* that is formed by three parts (*inputs; conditions; output*). The first part which is “ x ” is the input of the function.

The second part which is “ > 0 ” is the condition of the function that verify the input with a logical expression.

The third part which is “ $(1)/(0)$ ” is the output of the function in which if the conditions are verified the output value is equal to “1”, else if the conditions are not verified therefore the output is equal to “0”.

- Analyzing the expression $y = x; > 0; (1)/(0)$

This expression means *If $x > 0$ then $y = 1$ else $y = 0$* . This function is composed into three parts and each part has its own characteristics:

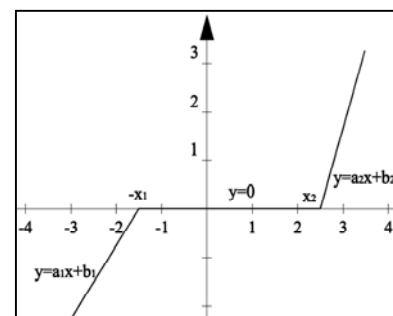
- 1- The first part is the input: it can be one or more. It is separated by a (;) from the second part.
- 2- The second part is the condition applied on the input to perform a logic expression that gives an output in a specific target. It is separated by a (;) from the third part.
- 3- The third part is the output: the value of the output will replace the InfoMath function in any expression e.g.: for the previous example $y = x; > 0; (1)/(0)$, *If $x > 0$* the expression $x; > 0; (1)/(0)$ is replaced by the output (1) and the expression will be $y = 1$. *If $x \leq 0$* the expression $x; > 0; (1)/(0)$ is replaced by the output (0) and the expression will be $y = 0$. In which it replaces the following expression

$$y = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \Leftrightarrow y = x; > 0; (1)/(0)$$

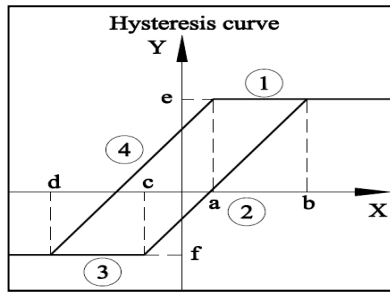
In this section a simple example is discussed, but in reality the *Infomath function* is much more complicated and it can be more developed in a way to replace any other existing function. In the following sections the *Infomath function* is developed by using some applications in mathematics, electrical engineering, power electronics and chemistry.

- Example of the application in mathematics:

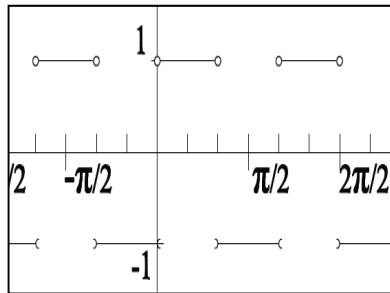
In this part, few figures are shown that can be formed by the *Infomath function*



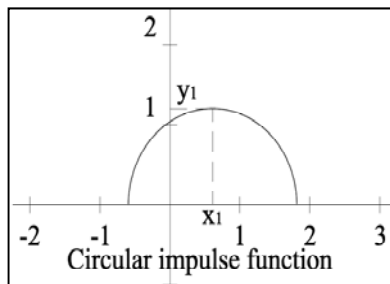
a) Dead zone function



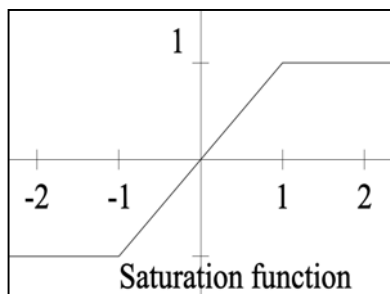
b) hysteresis curve



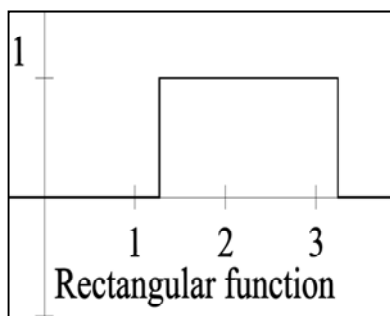
c) Angular function



d) Circular impulse function



e) Saturation function



f) Rectangular function

Fig. 1.a to 1.f: represent particular cases of the InfoMath Function (input;conditions;output)

The *InfoMath function* can describe any other function whatever is its complexity and length.

2 Definition of the *Infomath function* in Mathematics

The *Infomath function* is a mathematical function with a computer language base, in which it works as a computer software that receives inputs, analyses these inputs by conditions steps and finally to give an output. The main form of this function is "inputs;conditions;output"

- 1- The *input*: it can be one or more. It is separated by a (;) from the conditions. The *input* can be a constant number, scalar, parameter, function, vector, variable, matrix...
- 2- The *conditions*: the condition can be one or more logical expression. It has the role to verify the input if it applies the current conditions or not. In the following section, a table of the most important conditions is presented to use as conditions.
- 3- The *Output*: it can be one or more. It is separated by a (;) from the conditions. The *output* can be a constant number, scalar, parameter, function, vector, variable, matrix... there is two kind of output, one if the condition is verified and the other if the conditions is not verified. If the first one is written only, and if the conditions are not verified therefore the output takes a "1" as an initial value (by default).

2.1 Table represents some operators

The following table represents different type of operators used to form the condition part of the *InfoMath function*. This is a non exhaustive list, it gives an idea about the condition type and how to use it, but one can add and use an infinite number of operators.

2.2 How to use the *InfoMath function* in an expression

The *InfoMath function* can be used in many ways into an equation or expression; the following cases will be presented as a default value.

If conditions are not verified, the *InfoMath function* is replaced automatically by a value that doesn't affect the form of the expression which is "1".

In a general case, if conditions are not verified, the *InfoMath function* is replaced automatically by the value "1" as a default value e.g.: input;cond;(output) is the same as

Symbols	instruction name	example	Explanation
Logical operators:			
	XOR	$x:=a b;(y)$	if $x=a$ XOR $x=b$ is true \rightarrow put (y)
		$x:=(a b);(y)$	if $x=(a$ XOR $b) \rightarrow$ put (y)
	OR	$x:=a b;(y)$	if $x=a$ OR $x=b \rightarrow$ put (y)
		$x:=(a b);(y)$	if $x=(a$ OR $b) \rightarrow$ put (y)
&	AND	$x:=a \& b;(y)$	if $x=a$ AND $x=b \rightarrow$ put (y)
\uparrow or !&	NAND	$x:=a \uparrow b;(y)$	if $x=a$ NAND $x=b \rightarrow$ put (y)
/	ELSE (if not)	$x:= a: (y) / (z)$	if $x=a \rightarrow$ put (y) , if not ($x \neq a$) \rightarrow put (z)
!	NOT	$x:= ! a: (y)$	if $x=$ not a \rightarrow put (y)
Relational Operators:			
>	greater than	$x;>a;(y)$	if $x > a \rightarrow$ put (y)
<	less than	$x;<a;(y)$	if $x < a \rightarrow$ put (y)
>= or (\geq)	greater than or equal to	$x;>=a;(y)$	if $x \geq a \rightarrow$ put (y)
<= or (\leq)	less than or equal to	$x;<=a;(y)$	if $x \leq a \rightarrow$ put (y)
\neq	Not equal or (different)	$x;\neq a;(y)$	if $x \neq a \rightarrow$ put (y)
=	equal	$x;=a;(y)$	if $x = a \rightarrow$ put (y)
Intervals:			
[]	closed interval	$x:[a,b];(y)$	if x is a part of the closed interval [a,b] \rightarrow put (y)
] [opened interval	$x:]a,b[;(y)$	if x is a part of the opened interval]a,b[\rightarrow put (y)
[[or]]	semi-open interval	$x:[a,b];(y)$	if x is a part of the semi-opened interval [a,b[\rightarrow put (y)
a \rightarrow b	loop from a to b (if a<b increase)	$x;1\rightarrow 6;(y * x)$	if x vary from 1 to 6 \rightarrow put (y*x) for each variation
	or (if a>b decrease)	$x;7\rightarrow 2;(y * x)$	if x vary from 7 to 2 \rightarrow put (y*x) for each variation
Relation with groups or elements:			
\in	Element of	$x;\in [a,b];(y)$	if $x \in [a,b] \rightarrow$ put (y)
		$x;\in [a,b]]s,c[;(y)$	if $x \in [a,b]$ or $x \in]s,c[\rightarrow$ put (y)
\subset	Subset of	$x;\subset [a,b];(y)$	if $x \subset [a,b] \rightarrow$ put (y)
\cup	Union	$x;\in E(a) \cup E(b);(y)$	if x is an element of the 2 Ensembles a \cup b \rightarrow put (y)
\cap	Intersection	$x;\in E(a) \cap E(b);(y)$	if x is an element of the 2 Ensembles a \cap b \rightarrow put (y)
Domain of the number:			
\mathbb{D} or D	Set of decimal Nb.	$x;\mathbb{D};(y)$	if x is a decimal nb. \rightarrow put (y)
\mathbb{N} or N	Natural numbers	$x;\mathbb{N};(y)$	if x is a Natural nb. \rightarrow put (y)
\mathbb{Q} or Q	Set of rational Nb.	$x;\mathbb{Q};(y)$	if x is a rational nb. \rightarrow put (y)
\mathbb{R} or R	Set of real numbers	$x;\mathbb{R};(y)$	if x is a real nb. \rightarrow put (y)
\mathbb{Z} or Z	Set of integers	$x;\mathbb{Z};(y)$	if x is an integer nb. positive or negative \rightarrow put (y)
\mathbb{C} or C	Set of complex Nb.	$x;\mathbb{C};(y)$	if x est un nb complexe on met (y)
Re	Real part of a Complex nb.	$Re(x);=a;(y)$	if the real part of x is = a \rightarrow put (y)
		$x;Re;(y)$	if x is a real number \rightarrow put (y)
Im	Imaginary part of a Complex nb.	$Im(x);=a;(y)$	if the imaginary part of x is = a \rightarrow put (y)
		$x;Im;(y)$	if x is an Imaginary number \rightarrow put (y)

Characteristic and nature of the number:			
P or Ev	even number	$x;P;(y)$	if x is an even number → put (y)
I or Od	odd number	$x;I;(y)$	if x is an odd number → put (y)
Pr	Prime Number	$x;Pr;(y)$	if x is a Prime Number → put (y)
Bg(n)	beginning of a number by (n)	$x;Bg(3);(y)$	if x begin with the digit 3 → put (y), e.g.: 342 begin with the digit 3
T(n)	terminal of a number by (n)	$x;T(2);(y)$	if x is terminated with the digit 2 → put (y), e.g.:542 terminate by 2
		$x;T(52);(y)$	if x is terminated with digits 52 → put (y), e.g.:552 terminate by 52
Cd(n)	Number contain a digit (n)	$x;Cd(6);(y)$	if x contain the digit 6 → put (y), e.g.:34627 contain 6
M(n)	multiple of a number by (n)	$x;M(3);(y)$	if x is a multiple of (3) → put (y) , e.g.: 39 is a multiple of 3
D(n)	divisor of a number by (n)	$x;D(2);(y)$	if x is a divisor of (2) → put (y), e.g.: 8 is a divisor of 2
E(n)	Ensemble or group that have the same type or nature	$x;E(A);(y)$	$A=\{1,3,5,8,9,13,15,34,65\}$ is an Ensemble of integer numbers, if x is a part of the ensemble A → put (y)
Fb	Fibonacci Series	$x; \in Fb;(y)$	If x is an element of Fibonacci Series → put (y)

Table 1: represents some operators used to form the condition part of the InfoMath function.

Remark: x is the input and y is the output

$input; cond; (output)/(1)$ that is mean if the condition is not verified the InfoMath function is replaced by “1” therefore $input; cond; (output) = 1$. If we want the default value equal to zero we add a (0) at the beginning of the first parenthesis of the output. e.g.: $input ; cond; 0(output)$ is the same as $input ; cond; (output)/(0)$ that is mean if the condition is not verified the InfoMath function is replaced by 0 therefore $input; cond; 0(output) = 0$.

The same thing can be applied to any value at the output for example: $input ; cond; (output)/(A)$ that is means if the condition is not verified the InfoMath function is replaced by "A" which can be a scalar, parameter, vector, matrix... Therefore $input; cond; (output)/(A) = A$.

•The InfoMath function can be used with all kind of operators for example:

2.2.1 Multiplication and division

e.g.: $f(x) = a.x ; cond; (y)$ in this case if the condition is verified, the InfoMath function is replaced by the output “y” and the answer will be $f(x) = a.y$. In other hand, if the condition is not verified, the InfoMath function is replaced by “1” in

a way that will not affect the expression, the answer will be $f(x) = a$

So as a result

$$f(x) = \begin{cases} a.y & \text{if } x \text{ verify the condition} \\ a & \text{if } x \text{ doesn't verify the condition} \end{cases}$$

2.2.2 Addition or Subtraction

e.g. 1: $f(x) = a + x ; cond; (y)$ in this case if the condition is verified, the InfoMath function is replaced by the output “y” and the answer will be $f(x) = a + y$. In other hand if the condition is not verified, the InfoMath function is replaced by “1” as a default value, the answer in this case will be $f(x) = a + 1$.

So as a result

$$f(x) = \begin{cases} a + y & \text{if } x \text{ verify the condition} \\ a + 1 & \text{if } x \text{ doesn't verify the condition} \end{cases}$$

e.g.2: $f(x) = a + x ; cond; 0(y)$ in this case if the condition is not verified, the InfoMath function is replaced by “0” as a default value, the answer in this case will be $f(x) = a + 0 = a$.

So as a result

$$f(x) = \begin{cases} a + y & \text{if } x \text{ verify the condition} \\ a & \text{if } x \text{ doesn't verify the condition} \end{cases}$$

2.2.3 Power (or superscript x^m)

e.g.: $f(x) = A^{x;cond;(y)}$, in this case if the condition is verified, the *InfoMath function* is replaced by the output “y” and the answer will be $f(x) = A^y$. In other hand if the condition is not verified, the *InfoMath function* is replaced by “1”, the answer will be $f(x) = A$.

So as a result

$$f(x) = \begin{cases} A^y & \text{if } x \text{ verify the condition} \\ A & \text{if } x \text{ doesn't verify the condition} \end{cases}$$

2.2.4 Single expression

e.g.: $f(x) = x;cond;(y)$ in this case if the condition is verified, the *InfoMath function* is replaced by the output “y” and the answer will be $f(x) = y$. In other hand, if the condition is not verified, the *InfoMath function* is replaced by “1”, the answer will be $f(x) = 1$.

So as a result

$$f(x) = \begin{cases} y & \text{if } x \text{ verify the condition} \\ 1 & \text{if } x \text{ doesn't verify the condition} \end{cases}$$

Briefly the *Infomath function* can be used in any kind of expression

2.3 How the InfoMath function works

The form of the *InfoMath function* is:

inputs; cond1, cond2, cond3 ...; output1, output 2 ...

If the *input* verify the condition (*condition 1,2,3 ...*), the return value is *output*, and the output replace the expression of the *InfoMath function*.

If we have many conditions to verify with the input, we put a punctuation mark as a comma (,) between conditions, the same for the output, for example:

input; cond1, cond2, cond3 ...; (output1 for cond1), (output2 for cond2), (output3 for cond3) ...

Remark: only one output will replace the expression of the *InfoMath function* if its condition is verified, the priority is for the first condition and the first output then for the next.

It works in order from the most important condition to the less important; the same thing is for the output.

e.g.: $f(x) = x; > 0, > 3, = 5; (y1), (y2), (y3)$

If $x = 5$ the output is $y1$ and not $y3$ because the priority is for the first condition and the first output.

To obtain the output $y3$, we must put its condition in the first position and the same for the output, in this case the expression will be

$$f(x) = x; = 5, > 0, > 3; (y3), (y2), (y1)$$

So in this case if $x = 5$ the output is $y3$ and not $y1$ because the priority is for the first condition and the first output.

- One can see that conditions will be verified respectively, if the first condition is not verified the test will be on the second condition and so on...

- The Else (/) operator must be at the end of the output if it exists, because it can't be placed at the middle of the test.

Example 1:

$f(x) = x; = a, \in [b, c], < d, M(2); (c1), (c2), (c3), (c4)/(c5)$
if all conditions are not verified for the test then the output will be (c5).

Explanation of this example:

- if $x = a \Rightarrow$ the output will be (c1) even if the other conditions are verified, because the priority is for the first condition.

- If $x \neq a$ and if $x \in [b, c] \Rightarrow$ the output will be (c2) even if the other conditions are verified.

- If the two first conditions are not verified and if $x < d \Rightarrow$ the output will be (c3).

- If the three first conditions are not verified and if x is a multiple of 2 \Rightarrow the output will be (c4).

- If all conditions are not verified and if we put Else (/) at the end, in this case the output will be (c5).

- If all conditions are not verified and we haven't Else (/) at the end, in this case the output will be equal to “1” according to the default value that we have chosen.

Example 2: in this example we have 4 conditions and 3 values for the output, as the previous example.

$f(x) = x; = a, \in [b, c], < d, M(2); (c1), (c2), (c3)/(c5)$

if the number of conditions is greater than the number of outputs, in this case the first condition is for the first output, the second condition is for the second output, the third and the last conditions are for the last output, in this way the priority is always for the first condition and the first output and so on...

- If $x = a \Rightarrow$ the output will be (c1) even if the other conditions are verified, because the priority is for the first condition.

- If $x \neq a$ and if $x \in [b, c] \Rightarrow$ the output will be (c2) even if the other conditions are verified.
- If the two first conditions are not verified and if $x < d \Rightarrow$ the output will be (c3).
- If the three first conditions are not verified and if x is a multiple of 2 \Rightarrow the output will be (c3).
- If all conditions are not verified and if we put Else (/) at the end, in this case the output will be (c5).

Important remark is that the user must be very careful by using many conditions for one output.

The number of outputs must be less or equal to the number of conditions because the output must be a singular value and not a multiple choice. For example:

$f(x) = x; = a, \in [b, c]; (c1), (c2), (c3)/(c5)$ is not a correct expression because if $x \in [b, c]$ the output must be (c2) and not (c3), we conclude that (c3) can't be written, but in default case it is not a syntax error because it will not appear at the output even if conditions are verified or not.

3 Practical Examples in Mathematics

In this section, many practical examples are shown in order to help us to understand more quickly the *InfoMath function* and how it will be used in an expression to obtain the desired value.

3.1 Using the Logical operators

Let's say $x \in \mathbb{N}, y \in \mathbb{N}$

$$a) y = x + x; = 1|3|5; (2) \Rightarrow$$

$$y = \begin{cases} x + 1 & \text{if } x \neq \{1, 3, 5\} \\ x + 2 & \text{if } x = \{1, 3, 5\} \end{cases}$$

if $x=1$ or $x=3$ or $x=5$ the output will be (2) that will replace the InfoMath function.

$$b) y = x + x; = 1|3|5; (2) \Rightarrow$$

$$y = \begin{cases} 2x & \text{if } x \neq \{1, 3, 5\} \\ 3x & \text{if } x = \{1, 3, 5\} \end{cases}$$

$$c) y = x + x; = 1|3|5; 0(2) \Rightarrow$$

$$y = \begin{cases} x & \text{if } x \neq \{1, 3, 5\} \\ 3x & \text{if } x = \{1, 3, 5\} \end{cases}$$

$$d) y = x + x; (> 1 \& < 10); 0(4x) \Rightarrow$$

$$y = \begin{cases} x & \text{if } x \neq]1, 10[\\ x + 4x^2 & \text{if } x =]1, 10[\end{cases}$$

3.2 Using the Relational Operators

Let's say $x \in \mathbb{R}, y \in \mathbb{R}$

$$a) y = x + x; > 5; (x) \Rightarrow y = \begin{cases} 2x & \text{if } x \leq 5 \\ x + x^2 & \text{if } x > 5 \end{cases}$$

$$b) y = x + x; \geq 5; 0(\cos(x))$$

$$\Rightarrow y = \begin{cases} x & \text{if } x < 5 \\ x + x \cdot \cos(x) & \text{if } x \geq 5 \end{cases}$$

$$c) y = x + x; \geq 15, \geq 4; 0(x + 1), (3x + x^2)$$

$$\Rightarrow y = \begin{cases} x & \text{if } x < 4 \\ 4x + x^2 & \text{if } 4 \leq x < 15 \\ 2x + 1 & \text{if } x \geq 15 \end{cases}$$

$$d) y = x + x; \cos(x); > 0.5; (x)$$

$$\Rightarrow y = \begin{cases} 2x & \text{if } \cos(x) \leq 0.5 \\ x + x^2 & \text{if } \cos(x) > 0.5 \end{cases}$$

$$e) y_n = n + n; \geq 5; (\sum_{i=n}^{2n} (n + n^2))$$

$$\Rightarrow y_n = \begin{cases} n + 1 & \text{if } n < 5 \\ n + \sum_{i=n}^{2n} (n + n^2) & \text{if } n \geq 5 \end{cases}$$

3.3 Using Intervals

Let's say $x \in \mathbb{R}, y \in \mathbb{R}$

$$a) y = x + x; \in [5, 10] \cup [19, 23]; (\ln(x))$$

$$\Rightarrow y = \begin{cases} 2x & \text{if } x \notin [5, 10] \cup [19, 23] \\ x + x \cdot \ln(x) & \text{if } x \in [5, 10] \cup [19, 23] \end{cases}$$

$$b) y = x + x; \in [2, 4[, = 10; (\ln(x))$$

$$\Rightarrow y = \begin{cases} 2x & \text{if } x \notin [2, 4[\text{ or } x \neq 10 \\ x + x \cdot \ln(x) & \text{if } x \in [2, 4[\text{ or } x = 10 \end{cases}$$

$$c) y = x + x; \in [2, 4[, = 10; 0(\ln(x)), (3x)$$

$$\Rightarrow y = \begin{cases} x & \text{if } x \notin [2, 4[\text{ or } x \neq 10 \\ x + x \cdot \ln(x) & \text{if } x \in [2, 4[\\ x + 3x^2 & \text{if } x = 10 \end{cases}$$

d)

$$y = x + x; \in [2, 4[| = 10, = 12; 0(\ln(x)), (3x)$$

$$\Rightarrow y = \begin{cases} x & \text{if } x \notin [2, 4[\text{ or } x \neq 10 \text{ or } x \neq 12 \\ x + x \cdot \ln(x) & \text{if } x \in [2, 4[\text{ or } x = 10 \\ x + 3x^2 & \text{if } x = 12 \end{cases}$$

3.4 Using the Relation with groups or elements

Let's say $x \in \mathbb{N}, y \in \mathbb{R}, A = \{1, 2, 4, 6\}, B = \{9, 10\}, C = \{1, 2, 5\}$.

$$a) y = x/x; \in E(A); (2)/(4)$$

$$\Rightarrow y = \begin{cases} x/2 & \text{if } x = 1,2,4 \text{ or } 6 \\ x/4 & \text{if } x \neq 1,2,4 \text{ or } 6 \end{cases}$$

$$\begin{aligned} \text{b) } y &= x/x; \in (E(A) \cup E(B)); (2)/(4) \\ \Rightarrow y &= \begin{cases} \frac{x}{2} & \text{if } x = 1,2,4,6,9 \text{ or } 10 \\ \frac{x}{4} & \text{if } x \neq 1,2,4,6,9 \text{ or } 10 \end{cases} \end{aligned}$$

$$\begin{aligned} \text{c) } y &= x/x; \in E(A), E(C); (2), (3)/(4) \\ \Rightarrow y &= \begin{cases} \frac{x}{2} & \text{if } x = 1,2,4 \text{ or } 6 \\ \frac{x}{3} & \text{if } x \notin E(A) \text{ and } x = 5 \\ \frac{x}{4} & \text{if } x \neq 1,2,4,5 \text{ or } 6 \end{cases} \end{aligned}$$

3.5 Using the Domain of the number

Let's say $x \in \mathbb{C}, y \in \mathbb{C}, x = a + ib$ with a is the real part and b is the imaginary part

$$\begin{aligned} \text{a) } y &= x + Re(x); = 2; (7)/(ang_{Re(x)}(a)) \\ \Rightarrow y &= \begin{cases} x + 7 & \text{if } a = 2 \\ x + ang_{Re(x)}(a) & \text{if } a \neq 2 \end{cases} \end{aligned}$$

$$\begin{aligned} \text{b) } y &= x + x; Re; (7)/(rec_{2\pi}(a)) \\ \Rightarrow y &= \begin{cases} x + 7 & \text{if } b = 0 \text{ (} x \text{ is real)} \\ x + rect_{2\pi}(a) & \text{if } b \neq 0 \text{ (} x \text{ is complexe)} \end{cases} \end{aligned}$$

3.6 Using the Characteristic and nature of the number

Let's say $x \in \mathbb{Z}, y \in \mathbb{Z}$

$$\begin{aligned} \text{a) } y &= x + x; Ev; 0(1) \\ \Rightarrow y &= \begin{cases} x + 1 & \text{if } x \text{ is an Even number} \\ x & \text{if } x \text{ is an Odd number} \end{cases} \end{aligned}$$

$$\begin{aligned} \text{b) } y &= x + x; M(3); 0(1) \\ \Rightarrow y &= \begin{cases} x + 1 & \text{if } x \text{ is a multiple of } 3 \\ x & \text{if } x \text{ isn't a multiple of } 3 \end{cases} \end{aligned}$$

$$\begin{aligned} \text{c) } y &= x^{x; Pr; (3)} \\ \Rightarrow y &= \begin{cases} x^3 & \text{if } x \text{ is a Prime number} \\ x & \text{if } x \text{ isn't a Prime number} \end{cases} \end{aligned}$$

4 Application of Infomath function in electrical and electronic domains

In this section, some examples are given in order to have an idea about how to use the *Infomath function* and how it is important to use it when it is necessary.

4.1 Derivative form of the Angular function

The Angular functions are new mathematical functions introduced also by the author, that produce a rectangular signal, in which period is function of angles and not of time as the previous functions. Similar to trigonometric functions, the angular functions have the same properties as the precedent, but the difference is that a rectangular signal is obtained instead of a sinusoidal signal, and moreover, one can change the width of each positive and negative alternate in the same period. This is not the case of any other trigonometric function. In other hand, one can change the frequency, the amplitude and the width of any period of the signal at any position by using the general form of the angular function.

In this section, the author will not develop the angular function, it was published by WSEAS, for more information about the Angular function, please refer to the paper [20]. But the target of this section is how the *Infomath function* is introduced to formulate the derivative form of the angular function.

4.1.1 Derivative form of the Angular function $ang_{\alpha}(\beta(x + \gamma))$

The derivative form of the $ang_{\alpha}(\beta(x + \gamma))$ is:

$$ang_{\alpha}(\beta(x + \gamma))' = \begin{cases} 2(-1)^{2K} \cdot \delta \left(x - \left(\frac{2K\pi - \alpha}{\beta} - \gamma \right) \right) & (1) \end{cases}$$

$$\begin{cases} 2(-1)^{2K+1} \cdot \delta \left(x - \left(\frac{2K\pi + \alpha}{2\beta} - \gamma \right) \right) & (2) \end{cases}$$

with $K \in \mathbb{Z}$. The expression (1) gives $+2\delta$, when the value of $ang_{\alpha}(\beta(x + \gamma))$ pass from -1 to +1.

The expression (2) gives -2δ , when the value of $ang_{\alpha}(\beta(x + \gamma))$ pass from +1 to -1.

By using the *InfoMath* function, the two expressions (1) and (2) can be written in one expression as the following:

$$ang_{\alpha}(\beta(x + \gamma))' = 2 \frac{K}{|K|} (-1)^K \cdot \delta \left(x - \left(\frac{\left(\frac{K+K; I; \left(-\frac{K}{|K|} \right) \right) \pi - \alpha \frac{K}{|K|} (-1)^K}{\beta} - \gamma \right) \right) \quad (3)$$

With $K \in \mathbb{Z}^*$ or $\mathbb{Z} - \{0\}$ (only for this case).

$$\text{And } K; I; \left(-\frac{K}{|K|} \right) = \begin{cases} -\frac{K}{|K|} & \text{if } k \text{ is an odd number} \\ +1 & \text{if } k \text{ is an even number} \end{cases}$$

4.1.2 Derivative form of the Angular function $ang_{|\alpha|}(\beta(x + \gamma))$

The derivative form of the $ang_{|\alpha|}(\beta(x + \gamma))$ is:

$$ang_{|\alpha|}(\beta(x + \gamma))' = \begin{cases} 2(-1)^{2K} \cdot \delta \left(x - \left(\frac{2K\pi}{\beta} - \gamma \right) \right) & (4) \end{cases}$$

$$\begin{cases} 2(-1)^{2K+1} \cdot \delta \left(x - \left(\frac{2K\pi + \alpha}{2\beta} - \gamma \right) \right) & (5) \end{cases}$$

With $K \in \mathbb{Z}$. The expression (4) gives $+2\delta$, when the value of $ang_{|\alpha|}(\beta(x + \gamma))$ pass from -1 to +1.

The expression (5) gives -2δ , when the value of $ang_{|\alpha|}(\beta(x + \gamma))$ pass from +1 to -1.

By using the *InfoMath* function, the two expressions (4) and (5) can be written in one expression as the following:

$$ang_{|\alpha|}(\beta(x + \gamma))' = 2(-1)^K \cdot \delta \left(x - \left(\frac{(K+K;I;(-1))\pi+K;p;(0).\alpha}{\beta} - \gamma \right) \right) \quad (6)$$

With:

- $K \in \mathbb{Z}$
- $\underline{K; I; (-1)} = \begin{cases} -1 & \text{if } k \text{ is an odd number} \\ +1 & \text{if } k \text{ is an even number} \end{cases}$
- $\underline{K; P; (0)} = \begin{cases} 0 & \text{if } k \text{ is an even number} \\ +1 & \text{if } k \text{ is an odd number} \end{cases}$

4.2 Dirac Function

$$\delta(t - t_0) = \begin{cases} +\infty & \text{if } t = t_0 \\ 0 & \text{if } t \neq t_0 \end{cases}$$

is equivalent to: $\delta(t - t_0) = \underline{t; t_0; 0(\infty)}$

4.3 Sign function

$$Sign(x) = \begin{cases} +1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

is equivalent to: $Sign(x) = \underline{x; 0, < 0; (0), (-1)}$

4.4 Saturation function

$$Sat(x) = \begin{cases} +1 & \text{if } x > 1 \\ x & \text{if } x \in [-1, 1] \\ -1 & \text{if } x < -1 \end{cases}$$

is equivalent to:

$$Sat(x) = \underline{x; < -1, > 1; (-1), (1)/(x)}$$

4.5 Step Function

$$U(t - t_0) = \begin{cases} +1 & \text{if } t \geq t_0 \\ 0 & \text{if } t < t_0 \end{cases}$$

is equivalent to: $U(t - t_0) = \underline{t; < t_0; (0)}$

4.6 Rectangular Function

$$Rect_T(t - t_0) = \begin{cases} +1 & \text{if } t \in [t_0 - T, t_0 + T] \\ 0 & \text{if } t \notin [t_0 - T, t_0 + T] \end{cases}$$

is equivalent to:

$$Rect_T(t - t_0) = \underline{t; \notin [t_0 - T, t_0 + T]; (0)}$$

4.7 Parabolic Pulse Function

This is an original function introduced also by the author.

$$Parb_{T/b}(t - t_0) = \begin{cases} 0 & \text{if } t \notin \left[t_0 - \frac{T}{2}; t_0 + \frac{T}{2} \right] \\ b \left[1 - \frac{4(t-t_0)^2}{T^2} \right] & \text{if } t \in \left[t_0 - \frac{T}{2}; t_0 + \frac{T}{2} \right] \end{cases}$$

is equivalent to:

$$Parb_{T/b}(t - t_0) = b \left(1 - \frac{4(t-t_0)^2}{T^2} \right) \cdot \underline{t; \notin \left[t_0 - \frac{T}{2}, t_0 + \frac{T}{2} \right]; (0)}$$

4.8 Circular Pulse Function

This is an original function introduced also by the author.

$$Circ_T(t - t_0) = \begin{cases} 0 & \text{if } t \notin \left[t_0 - \frac{T}{2}; t_0 + \frac{T}{2} \right] \\ \sqrt{\frac{T^2}{4} - (t - t_0)^2} & \text{if } t \in \left[t_0 - \frac{T}{2}; t_0 + \frac{T}{2} \right] \end{cases}$$

is equivalent to:

$$Circ_T(t - t_0) = \sqrt{\frac{T^2}{4} - (t - t_0)^2} \cdot \underline{t; \notin \left[t_0 - \frac{T}{2}, t_0 + \frac{T}{2} \right]; (0)}$$

4.9 Dead zone function

$$y = \begin{cases} a_1x + b_1 & \text{if } x < -x_1 \\ 0 & \text{if } -x_1 \leq x \leq x_2 \\ a_2x + b_2 & \text{if } x > x_2 \end{cases}$$

is equivalent to:

$$y = \underline{a_{x; < -x_1, > x_2; 0(1),(2)} \cdot (x + x; < -x_1, > x_2; (x_1), (-x_2))}$$

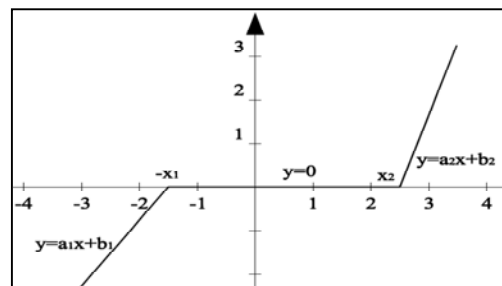


Fig.2: represents the Dead zone function.

4.10 Hysteresis curve

We have an infinite number of Hysteresis curves but we want to take a simple case only to demonstrate that it can be written into one formula using the *InfoMath function*. As an example we will take a quadrilateral parallelogram as shown in the figure below.

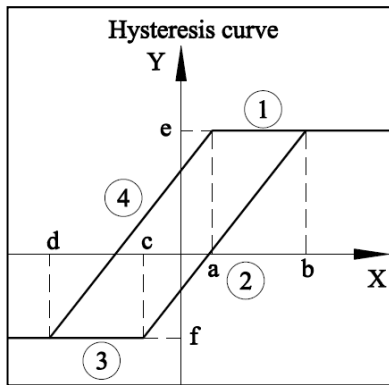


Fig. 3: represents the Hysteresis curve as a quadrilateral parallelogram.

- The curve (1) has the equation as: $y = e$ For $x > a$
- The curve (2) has the equation as:
 $y = \frac{e-f}{b-c}x + e - b\frac{e-f}{b-c}$ For $x \in [c, b]$
- The curve (3) has the equation as: $y = f$ For $x < c$
- The curve (4) has the equation as: $y = \frac{e-f}{a-d}x + e - a\frac{e-f}{a-d}$ For $x \in [d, a]$

To simplify the expression we can choose $b - c = a - d$ and $a > c$.
 For each point on the axe x we have 2 points on the axe y (y_1 and y_2).

For $x < d$ we have $y_2 = f$

$$\text{For } x \in [d, c[\text{ we have } \begin{cases} y_1 = \frac{e-f}{a-d}x + e - a\frac{e-f}{a-d} \\ y_2 = f \end{cases}$$

$$\text{For } x \in [c, a] \text{ we have } \begin{cases} y_1 = \frac{e-f}{a-d}x + e - a\frac{e-f}{a-d} \\ y_2 = \frac{e-f}{b-c}x + e - b\frac{e-f}{b-c} \end{cases}$$

$$\text{For } x \in]a, b] \text{ we have } \begin{cases} y_1 = e \\ y_2 = \frac{e-f}{b-c}x + e - b\frac{e-f}{b-c} \end{cases}$$

For $x > b$ we have $y_1 = e$

We can write these formulae into one formula by using the *Infomath function* as the following:

$$y_{i=1,2} = \left(\frac{e-f}{a-d}x - \frac{e-f}{a-d} \cdot i; 1; (a)/(b) \right) \cdot x; \in ([d, c[\&(i = 2)] | ([a, b] \&(i = 1))); (0) + e \cdot x; \in [d, c[\&(i = 2); (f/e)] \cdot x; \notin [d, b]; (0) + x; > b, < d; 0(e), (f)$$

4.11 Resistor-Capacitor-Coil element (RCC)

This is an original approach created by the author in order to unify many elements with different characteristics into one element that maintain all original characteristics of the elements.

In this section, an imaginary element is created that changes its performance and characteristics from Resistor to Capacitor and finally to Coil by varying a parameter like the temperature or the pressure on the element or any other phenomena.

Expressions of elements:

Resistor: $V = RI$

Capacitor: $V = I/(Cjw)$

Coil: $V = jwLI$

V is the voltage applied on the element.

I is the current that pass through the element.

R is the resistance of the element if it is a Resistor.

L is the inductance of the element if it is a Coil.

C is the capacitance of the element if it is a Capacitor.

j is the imaginary unit.

n is the parameter that changes the characteristic of the element.

w is the frequency of the circuit with $w = 2\pi f$.

The expression of the new element RCC is:

$$V = nRI \cdot n; = 2, = 3; (-j/CwRn), (Ljw/Rn)$$

With $n = [1,2,3]$

$$\text{With } \begin{cases} n = 1 \Rightarrow V = RI \\ n = 2 \Rightarrow V = I/(Cjw) \\ n = 3 \Rightarrow V = jwLI \end{cases}$$

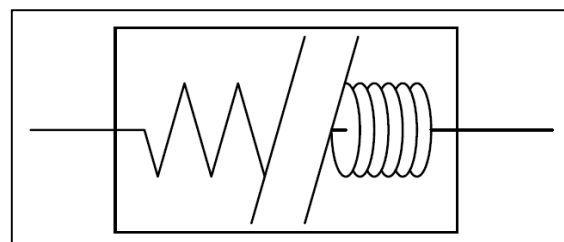


Fig. 4: represents the new element named RCC

This kind of elements can be introduced into a circuit and it will affect the desired output of the circuit.

This is a sample that demonstrates how the *Infomath function* can be used easily in order to develop a certain circuit of elements.

4.12 Circular Pulse Function combined with the angular function

We can combine multi-functions into only one function using the *Infomath function*. This is a modern concept of combining different equations, functions and expression in order to obtain something new that can be used in the electrical and electronic domains.

For example:

The expression of the Circular Pulse function is:

$$y_i = \begin{cases} \sqrt{\frac{T^2}{4} - (x - x_i)^2} & \text{for } x \in \left[x_i - \frac{T}{2}; x_i + \frac{T}{2} \right] \\ -a & \text{for } x \in \left] x_i + \frac{T}{2}; x_{i+1} - \frac{T}{2} \right[\end{cases}$$

With $i \in \mathbb{Z}, x \in \mathbb{R}$, let $x_i > iT, x_i = kiT$

$$y = \sum_{i=-\infty}^{+\infty} y_i$$

Suppose that $A = \left[ki - \frac{1}{2}, ki + \frac{1}{2} \right]$

And $B = \left] ki - \frac{1}{2}, ki + \frac{1}{2} \right[$

Suppose $x = XT$ then the expression is equivalent to:

$$y = \sum_{i=-\infty}^{+\infty} T \sqrt{\frac{1}{4} - (x - ki)^2} \cdot x \in A, B; (1), \left(-\frac{a}{T \sqrt{\frac{1}{4} - (x - ki)^2}} \right)$$

Or

$$y = \sum_{i=-\infty}^{+\infty} T \sqrt{\frac{1}{4} - (x - ki)^2} \cdot x \in B; \left(-a / (T \sqrt{1/4 - (x - ki)^2}) \right)$$

Other method

$$y = \sum_{i=-\infty}^{+\infty} \left(-a \left(\frac{T}{-a} \sqrt{\frac{1}{4} - (x - ki)^2} \right) \right)_{x \in \left] ki - \frac{1}{2}, ki + \frac{1}{2} \right[; (0)}$$

The graph of the function will be as shown in Fig. 5.

As we have seen, we can have many methods to write the expression, but the important thing is to choose the best method in a way to minimize the length and the complexity of the expression without losing any information or characteristic comparing with the initial one.

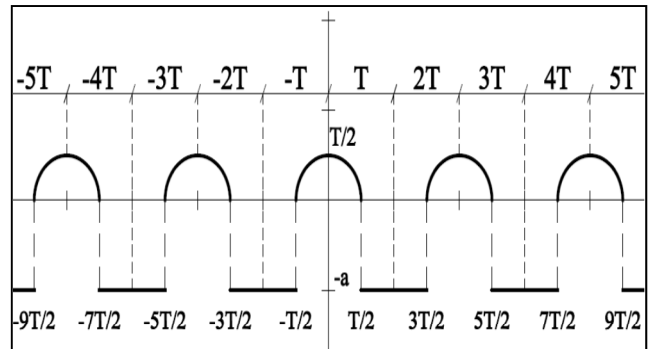


Fig. 5: represents the Circular Pulse Function combined with the Angular function

5 What is new with the *Infomath function*?

- This function is a live and dynamic function that we can manipulate as we want without limits, this is not the case of any other function, because any other function is specified for one job only, for example $y = f(x, z)$ describe curves only and not matrices or logical expressions.

- This function introduces an infinite number of methods to solve an equation or expression, this is not the case of any other function, for example: suppose $a < b$ and the expression of the condition part of the InfoMath function is: $x > a, x < b$ it is the same as $x \in]a, b[$, or $x \notin]-\infty, a]$ and $x \notin [b, +\infty[$, or $x \in [a, b] - (\{a\}, \{b\})$, or $x \not\leq a, x \not\geq b \dots$ but the result should be the same.

- It is a function that contains three parts, exactly as the computer language functions or routine or subroutine, and the parts are: input, conditions and output, and it can be transformed directly to the computer language, for example: $y = x + x; > a, < b; 0(1)$ is an expression using the InfoMath function, it is equivalent to a statement in the computer language as following:

```

if x > a or x < b then
    y = x + 1;
else y = x;
endif.
    
```

- The output can contain many different values but on can be consider as an output and replace the *InfoMath function* if its condition is verified, for example: $y = x + x; = a, b, > c; (2), (3), (4)$ it is an expression using the InfoMath function, it is equivalent to the following statements in a computer language:

Switch x :

(Select case in VB.NET)

Case $x = a$

$y = x + 2$; (is the output)

Case $x = b$

$y = x + 3$; (is the output)

Case $x > c$

$y = x + 4$; (is the output)

default $y = x + 1$; (is the output)

(Else case in VB.NET)

End switch

- As we have seen, briefly the Infomath function is the application of the computer language into the mathematics by using a defined shape and form. It is treated in the mathematics as any other function or expression.

6 Advantage and disadvantage of the *InfoMath function*

6.1 Advantages

- Easy to use and introduce in any expression and in any position in the expression.
- It can replace existing complicated functions or expressions by a simple expression.
- It can describe the form of any function even if it is very complicated.
- It makes many expressions easiest and simplest.
- We can use an infinite number of operators especially in the condition part.
- Try to minimize the size of the *InfoMath function* as possible by using your logical mentality in a goal to give your result a more important value and a more readable expression.
- It can be a constant number, scalar, parameter, function, vector, variable, matrix...
- It can have many inputs and many conditions.
- It has a singular result and output.
- It helps us to increase our logical mentality by searching and finding the best way to minimize the needed expression.

6.2 Disadvantages

- We can't have a general derivative or integral form for this function because it has no limit of conditions and inputs.
- It is not recommended to use this function just to have fun amongst students, because it is introduced only for one target, which is to reduce the number of many expressions if the classical methods can't do this job.

- It is not recommended to write a big expression in the condition part of the *InfoMath function* because it loses the aim of the mathematical logic.
- It is not recommended to use this function if its expression will be bigger than other methods that reduce the form of expressions.

7 Who must use this function?

- This function is based on the computer language like C++, VB.NET, MatLab or any other computer language; therefore it is logically that the user must know at least the basic of any computer language to understand how the function works.
- It can be used by students in their final project if it is necessary.
- It can be used when we need to reduce many formulae into only one formula.
- Anyone who is graduated and wants to continue his PhD degree and has problems to reduce the number of many equations and don't know how to do it...

8 Conclusion

The *Infomath function* is an original study introduced by the author into the mathematical domain. It is a powerful tool that can be used to simplify many complicated expressions into a single expression or function. It can be widely used in mathematics and as other functions it can be introduced in any kind of expression. The important side of this function is that it is not as the other functions which are limited in functionality. For example the function cosine has one function that produces a sinusoidal signal in which it can be considered as a sinusoidal function, it can't produce other type of forms, the same thing is for the logarithmic function and exponential function in which they have one characteristic by producing limited forms which are the characteristics of these functions. The case is different for the *Infomath function* in which it can take the form and characteristic of any other function. This will make it one of the greatest functions ever discovered in the modern history of mathematics. In previous sections, the *Infomath function* is developed and defined. Many examples are given in order to understand the functionality of this new function. Next studies will be done to present many applications in mathematical domain, chemistry and power electronics. These real applications reflect the importance of introducing the *Infomath function* into the mathematical domain.

References:

- [1] Claude Bayeh, M. Bernard, N. Moubayed, Introduction to the elliptical trigonometry, *WSEAS Transactions on Mathematics*, Issue 9, Vol. 8, 2009, pp. 551-560.
- [2] N. Moubayed, Claude Bayeh, M. Bernard, A survey on modeling and simulation of a signal source with controlled waveforms for industrial electronic applications, *WSEAS Transactions on Circuits and Systems*, Issue 11, Vol. 8, 2009, pp. 843-852.
- [3] M. Christopher, From Eudoxus to Einstein: A History of Mathematical Astronomy, *Cambridge University Press*, 2004.
- [4] Eric W. Weisstein, Trigonometric Addition Formulas, *Wolfram MathWorld*, 1999-2009.
- [5] Paul A. Foerster, Algebra and Trigonometry: Functions and Applications, *Addison-Wesley publishing company*, 1998.
- [6] Frank Ayres, Trigonométrie cours et problèmes, *McGraw-Hill*, 1991.
- [7] Robert C. Fisher and Allen D. Ziebur, Integrated Algebra and Trigonometry with Analytic Geometry, *Pearson Education Canada*, 2006.
- [8] E. Demiralp, Applications of High Dimensional Model Representations to Computer Vision, *WSEAS Transactions on Mathematics*, Issue 4, Vol. 8, 2009.
- [9] A. I. Grebennikov, Fast algorithm for solution of Dirichlet problem for Laplace equation, *WSEAS Transactions on Computers Journal*, 2(4), pp. 1039 – 1043, 2003.
- [10] I. Mitran, F. D. Popescu, M. S. Nan, S. S. Soba, Possibilities for Increasing the Use of Machineries Using Computer Assisted Statistical Methods, *WSEAS Transactions on Mathematics*, Issue 2, Vol. 8, 2009.
- [11] Q. Liu, Some Preconditioning Techniques for Linear Systems, *WSEAS Transactions on Mathematics*, Issue 9, Vol. 7, 2008.
- [12] A. I. Grebennikov, The study of the approximation quality of GR-method for solution of the Dirichlet problem for Laplace equation. *WSEAS Transactions on Mathematics*, 2(4), pp. 312-317, 2003.
- [13] R. Bracewell, Heaviside's Unit Step Function. The Fourier Transform and its Applications, 3rd edition, *New York: McGraw-Hill*, pp. 61-65, 2000.
- [14] Milton Abramowitz and Irene A. Stegun, eds, Handbook of mathematical functions with formulas, graphs and mathematical tables, 9 th printing, *New York: Dover*, 1972.
- [15] Vinit Kantabutra, On hardware for computing exponential and trigonometric functions, *IEEE Transactions on Computers*, Vol. 45, issue 3, pp. 328–339, 1996.
- [16] H. P. Thielman, A generalization of trigonometry, *National mathematics magazine*, Vol. 11, No. 8, 1937, pp. 349-351.
- [17] N. J. Wildberger, Divine proportions: Rational Trigonometry to Universal Geometry, *Wild Egg, Sydney*, 2005.
- [18] Cyril W. Lander, Power electronics, third edition, *McGraw-Hill Education*, 1993.
- [19] Claude Bayeh, Introduction to the Rectangular Trigonometry in Euclidian 2D-Space, *WSEAS Transactions on Mathematics*, Issue 3, Vol. 10, 2011, pp. 105-114.
- [20] Claude Ziad Bayeh, Introduction to the Angular Functions in Euclidian 2D-space, *WSEAS Transactions on Mathematics*, Issue 2, Vol. 11, 2012, pp.146-157
- [21] Claude Ziad Bayeh, Introduction to the General Trigonometry in Euclidian 2D-Space, *WSEAS Transactions on Mathematics*, Issue 2, Vol. 11, 2012, pp.158-172.
- [22] Georgi P. Tolstov, Fourier Series, *Courier Dover*, ISBN 0-486-63317-9, (1976).
- [23] William Mc C. Siebert, Circuits, signals, and systems, *MIT Press*, ISBN 978-0-262-19229-3, (1985), p. 402.