DESIGNING AN EXCEL VBA FUNCTION TO RECOGNIZE MORE IMPORTANT IRRATIONAL NUMBERS

Zoltán Fabulya

Faculty of Engineering, University of Szeged, Mars tér 7, 6724, Szeged, Hungary
e-mail: fabulya@mk.u-szeged.hu

ABSTRACT

Calculations typically performed on a calculator or computer show the result as a decimal fraction if it is not an integer. It would be easier to interpret the result if a value could be expressed with integers and operations, such as the root subtraction operation. This article shows how this can be done with a developed algorithm in Microsoft Excel, which recognizes the most famous irrational numbers and displays them in text form together with the character of the operation sign. For example, "5√3/2" is given for 4.330127019. It is also useful to display irrational numbers with integers because only an infinite number of decimal places in a decimal fraction could show the exact value, and that is not possible. So, the developed algorithm can display a more interpretable and accurate form of the irrational number. In addition to the results that can be written as square roots, the algorithm is capable of displaying irrational numbers that can be expressed as the number Pi, using the π character. The Excel algorithm which was implemented in Visual Basic for Applications shows all rational numbers as the quotient of two integers that are relative primes.

Keywords: irrational number, Excel VBA, programming, function, root recognition

1. INTRODUCTION

Irrational numbers appear in decimal form on the display of our electronic devices. The exact value will not be readable, because an infinite number of decimal digits make up the irrational numbers, in which there is no infinitely repeating section [1]. Such is the number Pi (π), but as a result of extracting the root, an irrational number is obtained even for integers if it is not a square number. An irrational number is shorter, more intelligible, and more accurate if appropriate symbols such as the root (√) or Pi (π) are used in their description. However, this type of display is not supported by our programs.
It is often necessary to enter the angles in radians. The decimal fraction form of a famous angle in radians is almost incomprehensible, as shown in (1).

$$30° = \frac{\pi}{6} = 0.523598775598299 \tag{1}$$

The values of the trigonometric functions in the case of famous angles are easier to interpret in radical form (2).

$$\cos 30° = \frac{\sqrt{3}}{2} = 0.866025403784439 \tag{2}$$

In the above formulas, the equation editor provided the appropriate display, but character mode can also be used (π/6, √3/2). This mode now allows us to read a decimal number obtained as a result of a calculation in a character-like, text-type manner in Excel. This article describes the development of an Excel worksheet function to display the most famous types of irrational numbers in character mode, where the essential element was the development of an algorithm for recognizing numbers in the Visual Basic for Applications (VBA) development environment. There are several research articles using the capabilities of Excel VBA [2] [3].

DOI: https://doi.org/10.14232/analecta.2022.1.62-70

2. MATERIALS AND METHODS

The theoretical background of the research is in the field of irrational numbers, while programming tasks require an overview of Excel VBA.
Since there are an infinite number of irrational numbers, including those that cannot be represented by special characters, the article deals only with the recognition of some of the more important types of irrational numbers, which have the following (3) and (4) forms.

$$\frac{m}{n} \times \pi \tag{3}$$

$$\frac{m}{n} \times \sqrt{a} \tag{4}$$

where:
$m =$ integer
$n =$ positive integer $(1 \leq n)$
$a =$ positive integer $(1 \leq a)$

So, the irrational number to be recognized is based on either the number Pi or the square root of an integer. An additional constraint is needed for numbers *n* and *a* since the recognition process cannot be extended to infinity [4]. During the tests, an upper limit of 100 proved to be sufficient. The numbers *m* and *n* should be integers so that the irrational value that causes the display problem can only be the result of Pi or square root subtraction.

2.1. Irrational numbers

Irrational numbers, by definition, are real numbers cannot be written as the quotient of two integers. It is easy to prove that the square root of a positive integer is also an irrational number, except for square numbers, where the square root is an integer [5].
Irrational numbers consist of an infinite number of decimal places without repeating sections. For this reason, their exact value cannot be described in decimal form, although any precision can be achieved by increasing the number of decimal places. That, in turn, will make the described number longer.
A famous group of irrational numbers are the ones that result from the square root of a rational number. Another group are the ones that denote an angle in radians which can be writen with rational numbers in degrees unit of measure [6]. These irrational numbers can be written in the form given by formula (3) or (4). Our goal was to develop an Excel VBA function that can display an irrational number, if possible, in this form.

2.2. Excel VBA

The typical purpose of using Excel spreadsheet program is to get calculated end results. This is achievable with built-in formulas and worksheet functions. In addition, user defined, custom functions can also be created [7]. To do that, some programming is necessary in Visual Basic for Applications (VBA). VBA is based on the Basic programming language. The functions have a name and give a result that depends on the variables in their argument [8]. The general format of the function is:

```
Public Function_name(arguments)
  statements
  function_name = value
```

DOI: https://doi.org/10.14232/analecta.2022.1.62-70

Analecta Technica Szegedinensia
Vol. 16, No. 01 ISSN 2064-7964 2022

```
End Function
```

When using functions, the value of their argument needs to be specified. Executing the statement creates a value that will be the result of the function `function_name = value` statement.
By default, the instructions in the function are executed once in a row. This is called the sequence instruction structure and has the following form [9]:

```
…
statement1
statement2
statement3
…
```

A selection structure is needed when the execution of the instruction depends on a condition [9]:

```
If condition Then
  statement1
Else
  statement2
End If
```

When the condition is met, the statements in `statement1` block are executed. Otherwise, the `statement2` block is performed.
An iteration structure is necessary in the program when instructions must be repeated several times [9]. In the following conditional loop structure, the statements are performed again and again until the condition is met:

```
While condition
  statements
Wend
```

Any program can be written with the sequence, selection, and iteration structures described above.

2.3. Search algorithm

An essential pillar of this work is to recognize from a number whether it can be produced in the needed form. This problem can be solved with a search algorithm. The point is to find the item that meets the criteria from a list of available items in case there is one. This is achievable by examining the items one by one until items run out or a suitable item is found [10]. This can be accomplished with a conditional loop structure. At the end of the loop, another selection structure is used to check whether the loop ended with a successful search or the items in the list ran out. The pseudocode of the algorithm is as follows:

```
Let the examined item to be the first item on the list
While the examined item is not appropriate and_
                                there is another item not yet tested
  Let the examined item to be the next item on the list
Wend
If the examined item is appropriate Then
  The hit is the examined item
Else
  No hit
End If
```

3. RESULTS

Within the research, the conditions were checked whether a number can be written in the form (3) or (4). This was necessary in order not to limit the number of digits that can be displayed in the given form. The

DOI: https://doi.org/10.14232/analecta.2022.1.62-70

next task was to create a program containing the necessary algorithms, and the last phase was to create the function that can be used in the Excel worksheet, which displays the number in the given (3) or (4) format.

3.1. Irrational numbers that can be accurately displayed

An irrational number can only be short and accurate if it can be expressed as an irrational number of a known value that can be described by a finite sequence of symbols or by a finite sequence of functions. For form (3) or (4), the Pi or the root sign is needed.

We can write any angular value in form (3) that can be described with a rational number $\left(\frac{p}{q}\right)$ in degrees, as shown in (5).

$$\frac{p}{q}[°] = \frac{p}{q} \times \frac{\pi}{180}[rad] = \frac{p}{180 \times q} \times \pi[rad] = \frac{m}{n} \times \pi[rad] \tag{5}$$

where:
$p =$ integer
$q =$ positive integer $(1 \leq q)$
$m =$ integer $(m = p)$
$n =$ positive integer $(n = 180 \times q)$

From derivation (6), it is visible that the form (4) can be utilized to write down the square root of a non-negative integer and the square root of any non-negative rational number.

$$\sqrt{\frac{p}{q}} = \sqrt{\frac{p \times q}{q^2}} = \frac{\sqrt{p \times q}}{q} = \frac{1}{q} \times \sqrt{p \times q} = \frac{m}{n} \times \sqrt{a} \tag{6}$$

where:
$p =$ not negative integer
$q =$ positive integer $(1 \leq q)$
$m =$ integer $(m = 1)$
$n =$ positive integer $(n = q)$
$a =$ positive integer $(1 \leq n = p \cdot q)$

The following derivation (7) shows an example where the square root of an integer can be given in several forms (4) using the square root of a smaller integer.

$$\begin{gathered}\sqrt{32} = \sqrt{4 \times 8} = 2 \times \sqrt{8} \\ \sqrt{32} = \sqrt{16 \times 2} = 4 \times \sqrt{2}\end{gathered} \tag{7}$$

Within the research, a search algorithm was developed which provides the form (4) as the square root of the smallest positive integer.

3.2. The created functions

First, a universal search function was created we could use for both forms (3) and (4). This was possible because these forms can be generalized, as shown in formula (8).

$$\frac{m}{n} \times \pi = \frac{m}{n} \times y, \frac{m}{n} \times \sqrt{a} = \frac{m}{n} \times y \tag{8}$$

where:
$m =$ integer
$n =$ positive integer $(1 \leq n)$
$a =$ positive integer $(1 \leq a)$
$y =$ generalized value $(y = \pi$ or $y = \sqrt{a})$

The search algorithm should indicate a successful search if the recognizable number (x) can be generated in the form (8), i.e., the relation (9) is met.

$$\mathrm{x} = \frac{m}{n} \times y \tag{9}$$

where:
$x =$ the number to recognize
$m =$ integer
$n =$ positive integer $(1 \leq n)$
$y =$ generalized value

A further interesting feature of this generalization is that for $y = 1$, the recognizable number we can be obtained as the quotient of two integers.
The created search function is a bivariate function of x and y that searches for an n positive integer for which equation (9) is fulfilled in the case of some integer m. The result of rearranging (9) to (10) shows that the value of m can be calculated for any value of n, but it cannot be guaranteed that m is an integer. This is exactly what the algorithm checks: to see if the counter (s) providing exact equality is an integer because only then can it be considered suitable $(m = s)$.

$$\mathrm{x} = \frac{s}{n} \times y \implies s = \frac{n \times x}{y} \tag{10}$$

where:
$x =$ the number to recognize
$s =$ not necessarily integer
$n =$ positive integer $(1 \leq n)$
$y =$ generalized value

The name of the search function below is denominator because it searches for and returns the `denominator` value that provides the desired representation. If no suitable denominator can be found, the result will be 0.

```
Public Function denominator(x, y)
  Dim n As Integer
  n = 1
  s = Abs(n * x / y)
  d = Abs(WorksheetFunction.Round(s, 0) - s)
  While n < 100 And d > 0.000001
    n = n + 1
    s = Abs(n * x / y)
    d = Abs(WorksheetFunction.Round(s, 0) - s)
```

DOI: https://doi.org/10.14232/analecta.2022.1.62-70

```
  Wend
  If n < 100 Then
    denominator = n
  Else
    denominator = 0
  End If
End Function
```

The function checks whether the value *s* ensuring equality (9) is an integer by examining the absolute value of its deviation from its rounded value as a difference (*d*). Of course, this should be 0 if *s* is an integer, but due to the inaccuracy of number representation, a non-negative value we can be considered as a 0 if it is not greater than 0.000001. Rounding is done with the function `Round`. The function starts from 1 with increments of 1 up to 100, searching for a suitable denominator. So, if it finds one, it will be the smallest one.

The following function is named numerator because it determines the numerator (*m*) of the equation (9) if it exists, i. e. a suitable denominator is found. The numerator will be 0 if there is no such denominator.

```
Public Function numerator(x, y)
  Dim n, m As Integer
  n = denominator(x, y)
  If n = 0 Then
    numerator = 0
  Else
    m = n * x / y
    numerator = m
  End If
End Function
```

Functions are also necessary to find out whether the recognizable number (*x*) can be produced with one of the Pi or root characters, i.e., in the form (3) or (4). These functions are called `pi_base` and `root_base` which return 0 if the desired format is not possible. Generatability can be ascertained from the value of the function denominator from the suitability of $y = \pi$ or $y = \sqrt{a}$. In the case of generatability, the value of the functions will be the value that provides generation, that is, the value of *Pi* of the function `pi_base` and the value *a* of function `root_base`. However, the latter is also a search algorithm for the value *a*, to be examined starting from 1 with increments of 1. Thus, in the case of a hit, the lowest suitable value is acquired, even if more than one value would be suitable.

```
Public Function pi_base(x)
  Dim n As Integer
  y = WorksheetFunction.Pi
  n = denominator(x, y)
  If n = 0 Then
    pi_base = 0
  Else
    pi_base = y
  End If
End Function

Public Function root_base(x)
  Dim a, n As Integer
  a = 1
  n = denominator(x, a)
  While n = 0 And a < 100
```

DOI: https://doi.org/10.14232/analecta.2022.1.62-70

```
    a = a + 1
    n = denominator(x, a ^ (1 / 2))
  Wend
  If a < 100 Then
    root_base = a
  Else
    root_base = 0
  End If
End Function
```

The functions created so far were used for the final one called `txt_num` that gives the number to be recognized in the form (3) or (4). The result is the appropriate form in text data type if it exists, or otherwise the number with four decimal places.

```
Public Function txt_num(x)
  Dim s, n As Integer
  If x = 0 Then
    txt_num = "0"
    Exit Function
  End If
  st = ""
  p = WorksheetFunction.Pi
  If pi_base(x) > 0 Then
    s = numerator(x, p)
    n = deniminator(x, p)
    symb = ChrW(960)              ' character Pi
  ElseIf root_base(x) > 0 Then
    a = root_base(x)
    If a = 1 Then
      y = 1
      symb = ""
    Else
      y = a ^ (1 / 2)
      symb = ChrW(8730) & a     'character root
    End If
    s = numerator(x, y)
    n = deniminator(x, y)
  Else
    txt_num = "" & WorksheetFunction.Round(x, 4)
    Exit Function
  End If
  If s = -1 Then
    If symb = "" Then
      st = "-1"
    Else
      st = "-" & symb
    End If
  ElseIf s = 1 Then
    If symb = "" Then
      st = "1"
    Else
      st = symb
```

DOI: https://doi.org/10.14232/analecta.2022.1.62-70

```
    End If
  Else
    st = s & symb
  End If
  If n > 1 Then
    st = st & " / " & n
  End If
  txt_num = st
End Function
```

The format of the representation depends on several conditions:

- Can the number be generated, and if so, in which form?
- If the value of the numerator is 1, it is only necessary to display it if the recognized value is rational (for example, 1/4). Otherwise, if it is irrational, it is sufficient to display Pi or the square root value in the numerator without a value of 1 (for example, π / 4 or √3/2).
- The dominator does not need to be displayed if its value is 1, but in this case, the numerator is required if its value is 1 and has a rational value.
- When displaying a negative number, the negative sign should appear in front of the number. Otherwise, it is unnecessary.

Table 1 shows the representation of some numbers obtained with the function.

Table 1. Some recognized numbers and their representation

Value	Representation
1.13137085	4√2/5
12.12435565	7√3
4.188790205	4π/3
4.294117647	73/17

4. CONCLUSION

A worksheet function was developed for the Microsoft Excel spreadsheet program, that can display the value of the most important types of irrational numbers in a clear, short, and more accurate form than in the inaccurate and difficult-to-understand decimal format. The range of recognizable numbers can be expanded according to the needs, for example, with irrational numbers which can be displayed with the help of the third and fourth roots. The developed function can be used in Excel spreadsheets for easier interpretation of results whenever it is necessary.

REFERENCES

[1] J. Gy. Obádovics, Matematika. Középiskolai tanulók, főiskolai és egyetemi hallgatók, valamint műszaki és gazdasági szakemberek számára, gyakorlati alkalmazásokkal, Tizenkilencedik, bővített kiadás, Scolar Kiadó, Budapest, 2012
[2] Gy. Hampel, Excel VBA alkalmazása egy biometriai esettanulmány példáján bemutatva, Jelenkori társadalmi és gazdasági folyamatok, 12 (4) (2017), pp. 35-40.
[3] Gy. Hampel, Egymintás t-próba programozható kialakítása Excel VBA környezetben, Jelenkori társadalmi és gazdasági folyamatok, 13 (3-4) (2018), pp. 169-175.
[4] I. Niven, Irrational Numbers - The Carus Mathematical Monnographs, Number 11, The Mathematical Association of America, New Jersey, 1956

DOI: https://doi.org/10.14232/analecta.2022.1.62-70

[5] R. P., Agrawal, H. Agrawal, Origin of Irrational Numbers and Their Approximations. Computation, 9 (3) (2021), pp. 1-49.
[6] I. Georgiev, L. Kristiansen, F. Stephan, Computable irrational numbers with representations of surprising complexity, Annals of Pure and Applied Logic, 172 (2) (2021), pp. 1-30.
[7] G. Kovalcsik, Az Excel programozása, Computerbooks, Budapest, 2005
[8] B. L. Matteson, Microsoft Excel Visual Basic Programmer's Guide, MicrosoftPress, Washington, 1995
[9] M., Alexander, D. Kusleika, Excel 2019 Power Programming with VBA, Wiley & Sons, Indianapolis, Indiana, 2019
[10] J. Walkenbach, Excel VBA Programming for Dummies, 3rd edition, John Wiley & Sons Inc., New Jersey, 2013

DOI: https://doi.org/10.14232/analecta.2022.1.62-70