

EXCEL VBA ALKALMAZÁS FEJLESZTÉSE KOMPLEX SZÁM GYÖKEINEK KISZÁMÍTÁSÁRA

Fabulya Zoltán

Abstract: Komplex számokkal műveletek végzését már számológépeink is támogatják. Viszont a gyökvonás csak egyetlen eredményt ad, még számológépes programokkal is, annak ellenére, hogy annyi különböző komplex számot kellene kapnunk, ahányadik gyököt vonunk. Ezért szükséges kifejleszteni az Excel táblázatkezelő program Visual Basic for Applications szolgáltatásával egy gyökvonást elvégző alkalmazást, mely az összes eredményt megjeleníti. A felhasználóbarát kezeléshez egyetlen űrlap biztosítja az adatok megadását és az eredmények megjelenítését különböző vezérlők segítségével. Az alkalmazás lehetőséget ad a komplex számok kanonikus, trigonometrikus és exponenciális alakja esetén is a használatra, támogatja a megjelenített számok könnyebb értelmezéséhez a gyökös formát, valamint a szögek esetén a fok és radián mértékegységet a Pi karakter (π) felhasználásával.

Abstract: Calculating operations with complex numbers is already supported by calculators. However, root extraction gives only one result, even with computer programs, despite giving as many different complex numbers as many radicals. Therefore, it is necessary to develop a root extraction application with Visual Basic for Applications in Excel that displays all the results. For user-friendly operation, a single form allows you to enter data and display results using different controls. The application allows the use of complex numbers in canonical, trigonometric and exponential forms, supports the radical form for easier interpretation of the displayed numbers, and the degree and radian units for angles using the Pi character (π).

Kulcsszavak: komplex szám, Excel VBA, programozás, függvény, gyökvonás.

Keywords: complex number, Excel VBA, programming, function, root extraction.

1. Bevezetés

Napjainkban nem csak számológépek, de egyre több számítógépes program is segíti a számítási műveletek kiértékelését. Sok eszköz támogatja a komplex számok használatát minden művelet, illetve függvény esetén. Viszont csak egyetlen eredmény adódik még akkor is, amikor a valós számkörben végzett műveletekkel ellentétben több különböző komplex szám tekinthető eredménynek (Obádovics, 2012). Egy ilyen fontos, több eredményt adó művelet komplex számok esetén a gyökvonás (Zippel, 1985).

Jelen kutatás célja egy olyan számítógépes alkalmazás fejlesztése, melynek segítségével a komplex számok gyökvonása elvégezhető úgy, hogy

- felhasználóbarát módon legyen kezelhető grafikus felületen,
- a komplex szám kanonikus, trigonometrikus és exponenciális alakban is megadható és megjeleníthető legyen.

Több okból célszerű a Microsoft Excel programot választani a komplex szám gyökeit meghatározó alkalmazás számára:

- A legtöbb felhasználó ismeri az Excel használatát.
- Biztosítja számítások elvégzését.
- Támogatja saját függvények, alkalmazások fejlesztését a Visual Basic for Applications (VBA) szolgáltatással (Walkenbach, 2013).

Számos kutatás felhasználhatja a VBA lehetőségeit, mellyel automatizálhatjuk a számítások elvégzését bármely tudományterület igényeinek megfelelően, így pénzügyi tervek (Zsótér, 2017) és beruházások gazdaságossági számításainál is (Zsótér–Túri, 2017). Leggyakrabban az Excel táblázatok kezelése támogatható VBA szolgáltatásokkal (Hampel, 2021a, 2021b).

2. Anyag és módszer

A kutatás során az egyik érintett háttér a komplex számok, melynél áttekintendő a hozzá tartozó elméleti, matematikai háttér, különös tekintettel a különböző alakú megadásukra és a gyökvonás műveletére. A másik tárgyalandó terület a Visual Basic for Applications (VBA), mely az alkalmazás programjának fejlesztéséhez szükséges.

2.1. A komplex számok és a gyökvonás művelet

Mint ismeretes, a négyzetgyök művelet negatív valós számon nem értelmezhető a valós számok körében. Ezért vált szükségessé a valós számok halmazának (\mathbb{R}) bővítése. A negatív számok gyökének képzéséhez elegendő egyetlen olyan szimbólumot (i vagy j) bevezetni, melynek négyzete -1 értékű, ahogy ez az (1) egyenletben is látható (McKeague, 2011). A továbbiakban az i szimbólum szerepel ebben a közleményben.

$$i^2 = -1 \Rightarrow i = \sqrt{-1} \quad (1)$$

Egy negatív szám négyzetgyöke így már megkapható egy valós szám és az i szorzataként, ahogy ez a (2) képletben látható.

$$\sqrt{-64} = \sqrt{64 \cdot (-1)} = \sqrt{64} \cdot \sqrt{-1} = 8i \quad (2)$$

Viszont ekkor már az eredményül kapott komplex szám gyökének is képezhetőnek kell lennie. Ráadásul minden olyan szám a gyökvonás eredményének tekintendő, melynek négyzeteként visszakapható az eredeti szám (Caviness–Fateman, 1976). Könnyen ellenőrizhető, hogy a $8i$ komplex számnak két gyöke is van (3).

$$\begin{aligned} (2 + 2i)^2 &= 2^2 + 2 \cdot 2 \cdot 2i + 2^2 i^2 = 4 + 8i - 4 = 8i \\ (-2 - 2i)^2 &= (-2)^2 + 2 \cdot (-2) \cdot (-2i) + (-2)^2 i^2 = 4 + 8i - 4 = 8i \end{aligned} \quad (3)$$

A fentiek alapján egy komplex szám két tag összegeként adható meg, mely tagokat a komplex szám valós részének és képzetes részének nevezünk. Így a komplex számok halmaza a (4) képletben látható, a komplex szám kanonikus alakjával leírt módon.

$$\mathbb{C} = \{x + y \cdot i\} \quad (4)$$

ahol:

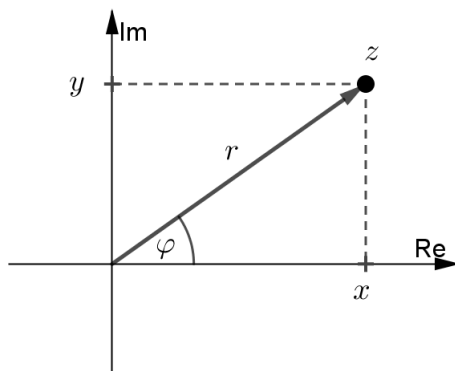
\mathbb{C} = a komplex számok halmaza

x = a komplex szám valós része ($x \in \mathbb{R}$)

y = a komplex szám képzetes része ($y \in \mathbb{R}$)

Egy komplex szám ábrázolása a komplex számsíkon történik, melyen egy pont reprezentál egy értéket (Silver, 1986). A pont megadható a vízszintes és függőleges tengely, azaz a valós (Re) és képzetes tengely (Im) egy-egy értékével, ahogy ez az 1. ábrán látható.

1. ábra: Komplex szám ábrázolása



Forrás: a szerző saját szerkesztése.

A komplex szám (z), mint pont az ábrán, megadható a pont origótól mért távolságával (r), és a helyvektorának a valós tengely pozitív felével bezárt szögével (φ) az óramutató járásával ellentétes értelemben pozitív szöggként (Bansal, 2006). Így adódik a komplex szám trigonometrikus alakja (5), míg az Euler-képlettel (6) a komplex szám exponenciális alakja (7).

$$z = x + y \cdot i = r \cdot (\cos \varphi + i \cdot \sin \varphi) \quad (5)$$

$$\cos \varphi + i \cdot \sin \varphi = e^{i \cdot \varphi} \quad (6)$$

$$z = x + y \cdot i = r \cdot e^{i \cdot \varphi} \quad (7)$$

ahol:

z = a komplex szám

$x = r \cos \varphi$ a komplex szám valós része ($x \in \mathbb{R}$)

$y = r \sin \varphi$ a komplex szám képzetes része ($y \in \mathbb{R}$)

$r = |z| = \sqrt{x^2 + y^2}$ a komplex szám nagysága

φ = a komplex szám argumentuma

A trigonometrikus vagy az exponenciális alak esetén a φ argumentumnak nem csak egyetlen értéke biztosítja ugyanazt a komplex számot. Ez a trigonometrikus függvények periodikusságából következik (8).

$$z = r \cdot (\cos(\varphi + 2k\pi) + i \cdot \sin(\varphi + 2k\pi)) = r \cdot e^{i \cdot (\varphi + 2k\pi)} \quad (8)$$

ahol:

z = a komplex szám

r = a komplex szám nagysága

$\varphi + 2k\pi$ = a komplex szám argumentuma ($k \in \mathbb{Z}$)

A (3) képletnél már látható volt, hogy a négyzetgyök két eredményt ad. Általánosságban elmondható, hogy az n -edik gyök n darab különböző komplex számot eredményez, mely az exponenciális alak alapján egyszerűen értelmezhető (9).

$$z_k = \sqrt[n]{z} = \sqrt[n]{r \cdot e^{i(\varphi+2k\pi)}} = \sqrt[n]{r} \cdot e^{i \frac{\varphi+2k\pi}{n}} \quad (k = 0, 1, \dots, n-1) \quad (9)$$

ahol:

z_k = az egyik eredmény

z = a komplex szám

r = a komplex szám nagysága

$\varphi + 2k\pi$ = a komplex szám argumentuma

n = ahányadik gyököt vonunk ($n \in \mathbb{N}, n > 1$)

Tehát n darab különböző eredmény adódik úgy, hogy a komplex szám nagysága n -edik gyökvonással, míg az argumentuma n -nel osztással képezhető (McKeague, 2011). Ez könnyen adódik a hatvány hatványozására vonatkozó azonosságból, valamint a gyökvonás törtkitevős alakjából (10).

$$\sqrt[n]{a^m} = (a^m)^{\frac{1}{n}} = a^{m \cdot \frac{1}{n}} = a^{\frac{m}{n}} \quad (10)$$

2.2. A Visual Basic for Applications programozási lehetőségei

A Microsoft Office Excel táblázatkezelő program lehetővé teszi a felhasználók számára, hogy a Visual Basic for Applications (VBA) szolgáltatással elkészíthető programokkal tegyék egyszerűbben kezelhetővé számítási tevékenységeiket (Kovalcsik, 2005). A programok kezelőfelülete vezérlők segítségével alakítható ki. Az adatok bevitele és az eredmények megjelenítése nem csak a munkafüzet celláiban történhet, hanem úrlapon kialakított vezérlők segítségével is a kifejlesztett VBA program által (Matteson, 1995).

A legsűrűbben alkalmazott vezérlők:

- Úrlap.
- Beviteli mező.
- Lista.
- Legördülő lista.
- Jelölő négyzet.
- Felirat.
- Parancsgomb.

Az alkalmazásaink programozása eseményvezérelt módon történik. Ez azt jelenti, hogy a vezérlőkön bekövetkező, a program használója által kiváltott eseményekhez társított programrészletekből épül fel a program (Matteson, 1995). A leggyakrabban a vezérlőre klikkelés és az adat megváltozásának eseménye igényel programozást.

3. Eredmények

A komplex szám gyökeit meghatározó alkalmazás felhasználói felületét a 2. ábrán látható űrlap biztosítja.

2. ábra: Az alkalmazás felhasználói felülete

Forrás: a szerző saját szerkesztése.

Az űrlap felső részén megjelenik a gyökök kiszámítását biztosító képlet, míg az alján a kilépés parancsgomb. Az adatok megadása és az eredmények megjelenítése *multipage* vezérlővel történik. Így a felhasználó lapozhat az *Adatok* és az *Eredmények* lapokra.

3.1. Az adatok megadása

A gyökvonáshoz ($\sqrt[n]{z}$) két adat megadása szükséges:

- Hányadik gyök (n).
- Komplex szám (z).

Az adatok lap tetején **Spinbutton** vezérlő (léptető gomb) található az n értékének 2 és 100 közötti beállítására és tárolására. Az érték egyesével csökkenthető vagy növelhető klikkeléssel. Mivel az adatot nem jeleníti meg a vezérlő, ezért szükség volt egy **Label** (címke) vezérlőre is, melynek tartalma megváltozik a **Spinbutton Change** eseményekor lefutó program hatására:

```
Private Sub SpinButton1_Change()  
    Label6 = SpinButton1.Value & ". gyök "End Sub
```

A komplex szám megadása három módon történhet:

- Kanonikus alakban.
- Trigonometrikus alakban.
- Exponenciális alakban.

A három mód egyikének kiválasztása legördülő lista (**ComboBox**) segítségével történik (3. ábra).

3. ábra: A komplex szám megadási módjának kiválasztása

Forrás: a szerző saját szerkesztése.

Kanonikus alak esetén a komplex szám valós (x) és képzetes (y) része adható meg, míg trigonometrikus illetve exponenciális alaknál a komplex szám nagysága (r) és argumentuma (φ). Egy-egy **Frame** (keret) vezérlő tartalmazza a szükséges beviteli mezőket, melyek közül csak az egyik lesz látható. A trigonometrikus alakhoz tartozó megjelenés látható a 4. ábrán.

4. ábra: A trigonometrikus/exponenciális alak beviteli felülete

Forrás: a szerző saját szerkesztése.

A választott módnak megfelelő megjelenést eredményező program:

```
Private Sub ComboBox1_Change()
    If ComboBox1.ListIndex = 0 Then
        Frame2.Visible = True
        Frame3.Visible = False
    Else
        Frame2.Visible = False
        Frame3.Visible = True
    End If
    Call kijelez
End Sub
```

A komplex szám két adatának megadásakor megjelenik egymás alatt a kanonikus, a trigonometrikus és az exponenciális alak is (5. ábra). Ehhez egy korábbi kutatás eredményeként elkészített VBA függvényre volt szükség (Fabulya, 2021), mely az értékeket gyökös alakjukban, illetve a π karakter segítségével képes mutatni.

5. ábra: A komplex szám három alakjának megjelenítése

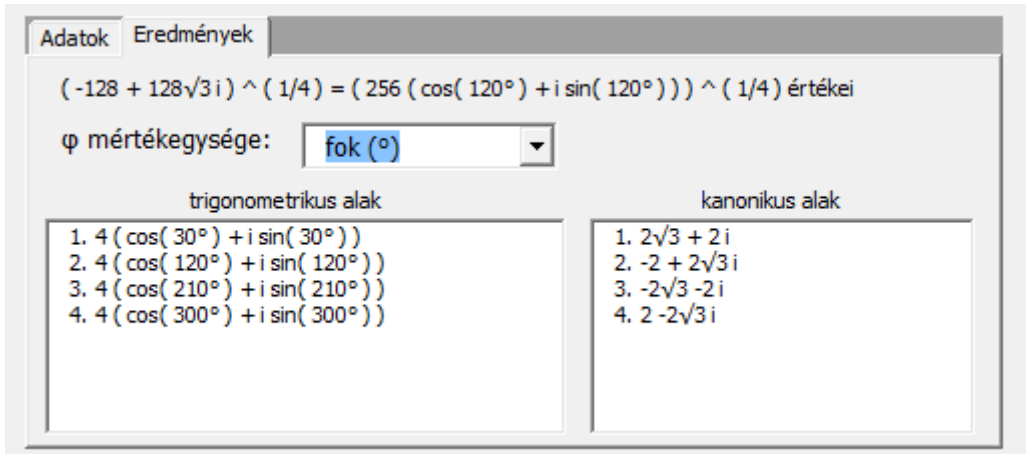
Trigonometrikus / Exponenciális		
r =	<input type="text" value="256"/>	-128 + 128√3i
φ [°] =	<input type="text" value="120"/>	256 (cos(120°) + i sin(120°))
		256 e ^ (i 2π / 3)

Forrás: saját kutatás adatai alapján a szerző szerkesztése.

3.2. Az eredmények megjelenítése

Átkapcsolva az eredmények lapra a gyökök megjelennek trigonometrikus és kanonikus alakban (6. ábra).

6. ábra: Az eredmények kijelzésének felülete



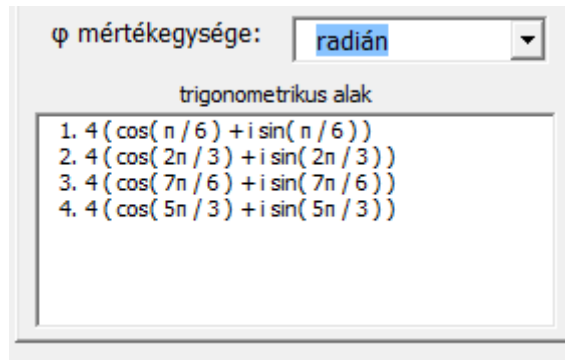
Forrás: saját kutatás adatai alapján a szerző szerkesztése.

Mivel az n -edik gyökvonás során n darab különböző gyök adódik, ezért egy-egy lista megjelenítésére alkalmas *ListBox* (lista) vezérlőt kellett alkalmazni a trigonometrikus és a kanonikus alak számára. A vezérlők adatokkal feltöltésének programja:

```
Public Sub kiir(r, fi, n, fok)
    ListBox1.Clear
    ListBox2.Clear
    p = WorksheetFunction.Pi
    r1 = r ^ (1 / n)
    For k = 0 To n - 1
        fil = (fi + k * 2 * p) / n
        txt = (k + 1) & ". " &
            kanonikus(r1 * Cos(fil), r1 * Sin(fil))
        ListBox1.AddItem txt
        txt = (k + 1) & ". " &
            trigonometrikus(r1, fil, fok)
        ListBox2.AddItem txt
    Next k
End Sub
```

A programban *For* ciklus eredményezi, hogy minden gyök kiszámítódjon és bekerüljön a listába az *AddItem* metódussal. Az eljárás a *fok* argumentum segítségével figyelembe veszi azt is, hogy *fok* vagy radián mértékegységgel szükséges az eredmény. A mértékegység beállításakor a kijelzés módosul (7. ábra).

7. ábra: Az eredmény radián mértékegységben



Forrás: saját kutatás adatai alapján a szerző szerkesztése.

4. Következtetések

Az elkészített alkalmazás a Microsoft Excel táblázatkezelő program Visual Basic for Applications szolgáltatását használva jól áttekinthető formában, egy párbeszédablakban (űrlap) teszi lehetővé komplex számok n -edik gyökeinek meghatározását. A komplex szám megadásakor támogatja a kanonikus, trigonometrikus és exponenciális alakot, melyek közötti átváltást is elvégzi. Az eredmények megjelenítése fok és radián mértékegységben kérhető. Az értékek kijelzésekor felismeri a gyökös forma, valamint a π karakter alkalmazhatóságát, mellyel könnyebben értelmezhetővé teszi az eredményeket.

Irodalomjegyzék

- Bansal, R.K. (2006): *New Approach to CBSE Mathematics IX*. Laxmi Publications. p. 25.
- Caviness, B.F., Fateman, R.J. (1976): Simplification of Radical Expressions. *Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation*, p. 329.
- Fabulya Z. (2021): Excel VBA függvény kialakítása Fontosabb irracionális számok felismerésére. *Jelenkori Társadalmi és Gazdasági Folyamatok*, 16 (3-4): 77–86. <https://doi.org/10.14232/jtgf.2021.3-4.77-86>.
- Hampel Gy. (2021a): A Microsoft Excel táblázatkezelő döntéstámogatás során hasznos szolgáltatásai. *Jelenkori Társadalmi és Gazdasági Folyamatok*, 16 (1-2): 189–202. <https://doi.org/10.14232/jtgf.2021.1-2.189-202>
- Hampel Gy. (2021b): Információfeldolgozás és döntéstámogatás gazdák számára Microsoft Excellel. In: Hampel Gy.; Kis K.; Monostori T. (szerk.): *Mezőgazdasági és vidékfejlesztési kutatások a jövő szolgálatában 2.: Tudomány: iránytű az elérhető jövőhöz*. Magyar Tudományos Akadémia Szegedi Akadémiai Bizottság Mezőgazdasági Szakbizottság, Szeged. 51–60.
- Kovalcsik G. (2005): *Az Excel programozása*. Computerbooks, Budapest.
- Matteson B. L. (1995): *Microsoft Excel Visual Basic Programmer's Guide*. Microsoft Press, Washington.
- McKeague, C. P. (2011). *Elementary algebra*. Cengage Learning. Boston, MA.
- Obádovics J. Gyula (2012): *Matematika. Középiskolai tanulók, főiskolai és egyetemi hallgatók, valamint műszaki és gazdasági szakemberek számára, gyakorlati alkalmazásokkal*. Tizenkilencedik, bővített kiadás, Scolar Kiadó, Budapest.
- Silver, H. A. (1986): *Algebra and trigonometry*. Englewood Cliffs, NJ: Prentice-Hall, Englewood Cliffs, NJ.

- Walkenbach, J. (2013): *Excel VBA Programming for Dummies*. 3rd edition. John Wiley & Sons Inc., New Jersey.
- Zippel, R. (1985): Simplification of Expressions Involving Radicals. *Journal of Symbolic Computation*, 1 (2): 189–210. [https://doi.org/10.1016/S0747-7171\(85\)80014-6](https://doi.org/10.1016/S0747-7171(85)80014-6)
- Zsótér B. (2017): Financial planning in connection with accomodation development in a sport centre. *Quaestus Multidisciplinary Research Journal*, 4 (11): 172–177.
- Zsótér B., Túri I. (2017): Economical calculations related to a smoking technology investment of a pork processing plant. *Annals of Faculty of Engineering Hunedoara*, 15 (4): 57–61.