Jelenkori társadalmi és gazdasági folyamatok, (2019) XIV. évfolyam, 1. szám, pp. 29–34.

EXCEL VBA FÜGGVÉNYEK KIALAKÍTÁSA HÁROMDIMENZIÓS VEKTOROK MATEMATIKAI ALKALMAZÁSÁRA

Fabulya Zoltán

Absztrakt: A háromdimenziós vektorokkal végzett műveletek számítógépes támogatottsága nem megfelelő. A sokak által használt Excel táblázatkezelő program sem rendelkezik testreszabott szolgáltatásokkal a vektorműveletekhez. Ezt a hiányosságot az Excel Visual Basic for Applications nyújtotta lehetőségekkel megszüntethetjük. Kialakíthatunk felhasználói függvényeket, melyekkel kiszámíthatjuk a vektorok matematikai alkalmazási területéhez kapcsolódó eredményeket:

- Vektor nagysága, abszolút értéke
- Két vektor által bezárt szög, illetve a szög koszinusza
- Két vektor skaláris szorzata
- Két vektor vektoriális szorzata
- Három vektor vegyes szorzata

Az így elkészített függvények a megszokott módon elérhetők lesznek a táblázatkezelő programban, így biztosítva a háromdimenziós vektorok témakörébe tartozó matematikai számítások egyszerű elvégzését.

Abstract: Computer support of three-dimensional vector operations is inadequate. Even Excel spreadsheet program, used by many, does not have customized services for vector operations. This defect can be solved with the options provided by Visual Basic for Applications in Excel. We can create user functions to calculate results related to the mathematical application of vectors:

- The size and the absolute value of vectors
- The angle between two vectors and the cosine of the angle
- The scalar product of two vectors
- The vector product of two vectors
- The mixed product of three vectors

The created user functions will be available in the spreadsheet program in the usual way, providing the simple execution of the mathematical calculations in the area of the three-dimensional vectors.

Kulcsszavak: Excel VBA, programozás, vektorok műveletei

Keywords: Excel VBA, programming, vector operations

1. Bevezetés

A matematika egyes területein sok számítást igénylő témakörrel találkozhatunk. Ilyen a lineáris algebra is, melyben a háromdimenziós vektorok és műveleteik különösen fontosak a gyakorlatban betöltött szerepük miatt. Ennek ellenére a számítógépes támogatottságban hiányolhatjuk, hogy a számításokhoz praktikusan alkalmazható, általánosan elterjedt Excel táblázatkezelő program nem rendelkezik ilyen függvényekkel (Matteson, 1995).

Célunk elkészíteni az Excel programozhatóságát nyújtó Visual Basic for Applicatons bővítménnyel a háromdimenziós vektorokon alkalmazható műveletek eredményét kiszámító függvényeket. Így kiegészítjük a táblázatkezelő programban elérhető függvények körét, s a megszokott használati módot biztosíthatjuk a vektorokkal végezhető műveletek számára (Kovalcsik, 2005).

2. Anyag és módszer

A Microsoft Excel 2010 verzióját használjuk a Visual Basic for Application (VBA) szolgáltatással, mely a programozási környezetet biztosítja az elkészítendő függvényekhez (Zimmerman, 1996). A függvények az argumentumukban szereplő, vektor értékű tömbváltozóktól függő eredményt számítanak ki (Wells–Harshberger, 1997).

Az Excel táblázatkezelő program széleskörűen használható. Így pénzügyi kalkulációknál (Zsótér–Túri, 2017), felmérések kiértékelésére (Zsótér–Tóth, 2014), statisztikai hipotézis vizsgálatok automatizálására (Hampel, 2018a, 2018b), vagy akár ipari folyamatok összehangolásának támogatására (Fabulya, 2017).

2.1. Háromdimenziós vektorok matematikai műveletei

A vektorok algebrai megadása három koordináta értékkel történik, mely a háromdimenziós tér egyes irányaiban a vektor komponenseit, adott irányú kiterjedését mutatja (*1. képlet*).

$$\bar{a} = (a_1; a_2; a_3) \tag{1}$$

ahol:
$\bar{a}$ – vektor
a_1, a_2, a_3 – a vektor komponensei, koordinátái

A vektor koordinátái egy-egy valós számmal adhatók meg. A vektorokon végzett műveletek eredményét a koordinátáik értéke alapján tudjuk kiszámítani.

Vektorok összegeként (*2*), különbségeként (*3*), skalárral szorzásakor (*4*), illetve két vektor vektoriális szorzatának (*5*) eredményeként is vektort kapunk a következő képletekkel:

$$\bar{a} + \bar{b} = (a_1; a_2; a_3) + (b_1; b_2; b_3) = (a_1 + b_1; a_2 + b_2; a_3 + b_3) \tag{2}$$

$$\bar{a} - \bar{b} = (a_1; a_2; a_3) - (b_1; b_2; b_3) = (a_1 - b_1; a_2 - b_2; a_3 - b_3) \tag{3}$$

$$\lambda \cdot \bar{a} = \lambda \cdot (a_1; a_2; a_3) = (\lambda \cdot a_1; \lambda \cdot a_2; \lambda \cdot a_3) \tag{4}$$

$$\bar{a} \times \bar{b} = (a_1; a_2; a_3) \times (b_1; b_2; b_3) = \\ = (a_2 \cdot b_3 - a_3 \cdot b_2; -a_1 \cdot b_3 + a_3 \cdot b_1; a_1 \cdot b_2 - a_2 \cdot b_1) \tag{5}$$

ahol:
$\lambda \in \mathbb{R}$ – skalár

Több olyan művelet is van vektorok esetén, melynek eredményeként nem vektort, csak egy számot, skalárt kapunk. Ilyen művelet egy vektor abszolút értéke (*6*), két vektor skaláris szorzata (*7*), három vektor vegyes szorzata (*8*), két vektor által bezárt szög koszinusza (*9*) és a bezárt (*10*). Ezeket az értékeket az alábbi képletekkel számíthatjuk ki:

$$|\bar{a}| = |(a_1; a_2; a_3)| = \sqrt{a_1^2 + a_2^2 + a_3^2} \tag{6}$$

$$\bar{a} \cdot \bar{b} = (a_1; a_2; a_3) \cdot (b_1; b_2; b_3) = a_1 \cdot b_1 + a_2 \cdot b_2 + a_3 \cdot b_3 \tag{7}$$

$$\begin{aligned} \bar{a} \cdot \left(\bar{b} \times \bar{c}\right) &= (a_1; a_2; a_3) \cdot \left((b_1; b_2; b_3) \times (c_1; c_2; c_3)\right) = \\ &= a_1 \cdot (b_2 \cdot c_3 - b_3 \cdot c_2) - a_2 \cdot (b_1 \cdot c_3 - b_3 \cdot c_1) + \\ &+ a_3 \cdot (b_1 \cdot c_2 - b_2 \cdot c_1) \end{aligned} \tag{8}$$

$$cos\varphi = \frac{\bar{a} \cdot \bar{b}}{|\bar{a}| \cdot |\bar{b}|} \tag{9}$$

$$\varphi = arccos \frac{\bar{a} \cdot \bar{b}}{|\bar{a}| \cdot |\bar{b}|} \tag{10}$$

2.2. Excel VBA bővítmény

A Visual Basic for Applications bővítmény lehetőséget nyújt saját függvények készítésére, melyek ugyanúgy használhatók, mint az Excel többi függvénye. A programozás során az Excel biztosítja az integrált programfejlesztő környezetet a Visual Basic programozási nyelvvel. Az elkészítendő függvények argumentumában szereplő változók háromdimenziós tömbök adatait tárolják, azaz a vektorok koordinátáit. A tömbváltozók egyes adatait indexük segítségével érhetjük el. Így például egy vektor ($\bar{a}$) első koordinátáját (a_1) a tömbváltozó (a) egyes értékű indexén ($a(1)$) találjuk. Tehát az index értékét a változó nevét követő zárójelek között kell megadni. A függvény eredményét az argumentumukban megadott változók segítségével kell képezni. Ehhez értékadó utasítást kell alkalmazni matematikai kifejezésekkel, melyeket a szokásos matematikai műveleti jelekkel (operátorokkal) tudunk megadni.

3. Eredmények és értékelésük

Feladatunk elkészíteni a vektorokon végzett műveletek eredményét kiszámító függvények programját. Ehhez viszont azt is célszerű figyelembe venni, hogy egyes műveletek eredménye felhasználható egy másik műveletnél. A legegyszerűbb műveletek, mint a vektorok összeadása, kivonása és skalárral szorzása, még csak függvény programozását sem igénylik, hiszen közvetlenül a cellában kialakított formulával képezhetők. Ezek a szokásos formuláktól ezek annyiban térnek el, hogy az eredmény nem egy érték egy cellában, hanem egy három cellából álló tömb az eredmény vektor három koordinátájának számára.

Két vektor skaláris szorzatát az alábbi függvénnyel kapjuk meg:

```
Public Function VektorSkalarSzorzat(a, b)
    s = 0
    For i = 1 To 3
        s = s + a(i) * b(i)
    Next i
    VektorSkalarSzorzat = s
End Function
```

A függvény neve `VektorSkalarSzorzat`, melynek eredménye két, argumentumában szereplő vektorból (a és b tömbváltozókból) képezhető a *7. képlet* alapján. A programban az összegzés elemi algoritmusára volt szükség, hogy egy változó kezdeti 0 értékét (`s=0`) növelve az azonos indexű koordináták szorzatával (`s=s+a(i)*b(i)`) megkapjuk az eredményt, a függvény értékét (`VektorSkalarSzorzat=s`).

Egy vektor abszolút értékét (*6. képlet*) megkaphatjuk a skalárszorzat eredményének felhasználásával:

$$|\bar{a}| = \sqrt{\bar{a} \cdot \bar{a}} = (\bar{a} \cdot \bar{a})^{\frac{1}{2}}$$

Ezt kihasználva a `VektorAbs` nevű függvény könnyen elkészíthető:

```
Public Function VektorAbs(a)
    VektorAbs = VektorSkalarSzorzat(a, a) ^ 0.5
End Function
```

A kalap jel (`^`) a hatványozás elvégzésének operátora, mellyel a skalárszorzat eredményéből gyököt vontunk.

Két vektor által bezárt szög koszinuszát a *9. képlettel* számíthatjuk ki, melyben szükség van vektorok skalár szorzatának és abszolút értékének eredményére. A függvény neve `VektorCosFi`:

```
Public Function VektorCosFi(a, b)
    szam = VektorSkalarSzorzat(a, b)
    nev = VektorAbs(a) * VektorAbs(b)
    If nev = 0 Then
        VektorCosFi = 0
    Else
        VektorCosFi = szam / nev
    End If
End Function
```

Elsőként a végeredményt adó tört számlálójának (`szam`) és nevezőjének (`nev`) értékét számítjuk ki. Viszont a tört nevezője nem lehet nulla, ezért ezt ellenőrizni kell. A nevező csak úgy lehet nulla, ha legalább az egyik vektor nulla vektor, tehát nulla az abszolút értéke. De ekkor a vektorok skalár szorzata is nulla, mely a vektorok merőlegességének szükséges és elegendő feltétele. Így a vektorok által bezárt szög koszinusza is nulla értékű (`VektorCosFi=0`) amikor merőlegesek. Ha a nevező nem nulla, akkor a tört értéke (`szam/nev`) adja a végeredményt.

Két vektor által bezárt szög kiszámításához felhasználhatjuk a szög koszinuszát. Az eredményt az arkusz-koszinusz (`Acos`) munkalapfüggvénnyel kapjuk meg:

```
Public Function VektorFi(a, b)
     CosFi = VektorCosFi(a, b)
     VektorFi = WorksheetFunction.Acos(CosFi)
End Function
```

Két vektor vektoriális szorzatát olyan függvénnyel kell meghatározni, melynek eredménye egy vektor. Ehhez egy tömbváltozót (`t(1 To 3)`) kell létrehozni, melynek három elemére indexükkel hivatkozhatunk. A függvény két argumentuma az összeszorzandó vektorok (`a, b`). A függvény programja a következő:

```
Public Function VektorVektorSzorzat(a, b)
    Dim t(1 To 3)
    t(1) = a(2) * b(3) - a(3) * b(2)
    t(2) = -a(1) * b(3) + a(3) * b(1)
    t(3) = a(1) * b(2) - a(2) * b(1)
    VektorVektorSzorzat = Application.Transpose(t)
End Function
```

A `VektorVektorSzorzat` függvényben az eredmény vektor három koordinátájának értékét kell kiszámolni. Mivel egy vektort oszlopként szoktunk értelmezni, ezért az Excelben sorként értelmezett tömbnek oszloppá alakítása transzponálással (`Transpose`) történik. Ez az oszlopként értelmezett tömb lesz a vektoriális szorzat eredményének vektora. Ha ezt az eredményt egy másik függvényben fel szeretnénk használni, akkor transzponálással alakíthatjuk vissza adatsorként értelmezhető tömbváltozóba. Erre szükségünk lesz három vektor vegyes szorzatának függvényénél (`VektorVegyesSzorzat`):

```
Public Function VektorVegyesSzorzat(a, b, c)
    Dim t As Variant
    t=Application.Transpose(VektorVektorSzorzat(b,c))
    VektorVegyesSzorzat = VektorSkalarSzorzat(a, t)
End Function
```

A vegyes szorzat képzéséhez elsőként vektoriális szorzatot képzünk, majd ennek eredményét felhasználjuk a végeredményt adó skaláris szorzatnál. Így elegendő a már elkészített függvényeket használnunk.

4. Következtetések, összegzés, záró megjegyzések, záró gondolatok

Az elkészített függvényeket az Excelben megszokott módon tudjuk használni háromdimenziós vektorok matematikai műveleteinek elvégzésére. A függvények nem csak abban a munkafüzetben lesznek elérhetők, amelyekben létrehoztuk és tároljuk őket, ha ezt is megnyitjuk munkánk során. Lehetőségünk van arra is, hogy bővítményt hozzunk létre, hogy a fájl megnyitása nélkül is használhassuk felhasználóbarát módon a függvényeket.

Irodalomjegyzék

Fabulya Z. (2017): Hőkezelési folyamatok összehangolása Excel VBA szolgáltatásokkal. *Jelenkori társadalmi és gazdasági folyamatok*, 12 (4): 19–25.

Hampel Gy. (2018a): Excel alkalmazása normális eloszlás tesztelésére Shapiro Wilk próbával. *Jelenkori társadalmi és gazdasági folyamatok*, 13 (1–2): 77–82.

Hampel Gy. (2018b): Egymintás t-próba programozható kialakítása Excel VBA környezetben. *Jelenkori társadalmi és gazdasági folyamatok*, 13 (3–4): 169–175.

Kovalcsik G. (2005): *Az Excel programozása*. Computerbooks, Budapest.

Matteson B. L. (1995): *Microsoft Excel Visual Basic Programmer's Guide*. Microsoft Press, Washington.

Wells E., Harshberger S. (1997): *Microsoft Excel 97 Developer's Handbook*. Microsoft Press, Washington.

Zimmerman M. W. (1996): *Microsoft Office 97 Visual Basic Programmer's Guide*. Microsoft Press, Washington.

Zsótér B., Tóth A. (2014): Examination of statisfaction related to investments (2006-2011) accomplished by the local council in Abony. *Analecta Technica Szegedinensia* 8 (1): 33–37.

Zsótér B., Túri I. (2017): Economical calculations related to a smoking technology investment of a pork processing plant. *Annals of Faculty of Engineering Hunedoara – International Journal of Engineering* 15 (4): 57–61.