TÁBLÁZATKEZELÕ PROGRAMRENDSZEREK FELHASZNÁLÁSA A NUMERIKUS MATEMATIKÁBAN

Perge Imre
Eszterházy Károly Tanárképzõ Fõiskola

Abstract

In this work we would like to show the advantages of using spreadsheet for solving numerical computations. We can give the numerical solution of non-linear equations, linear equations systems and ordinary differential equations with the help of spreadsheets.

1. Bevezetés

A programnyelvek megjelenésével szinte egyidõben elsõ alkalmazásként a numerikus matematikai témakörök (nemlineáris egyenletek, lineáris egyenletrendszerek megoldása, interpoláció, numerikus integrálás stb.) megoldására szolgáló programok készültek el a szóban forgó programnyelveken.

A táblázatkezelõ programrendszerek, mint "óriási" táblázatok kezelõje pedig elsõsorban a gazdasági jellegû, nyilvántartási és tervezési feladatok alkalmazását sugallta és valósította meg. Nem találkoztunk a táblázatkezelõ rendszerekrõl szóló irodalomban és a feladatok között sem, annak numerikus matematikai táblázatokra való felhasználásával. Pedig ezek nagy része néhány képlet begépelése és átmásolása után – rendkívül szemléletesen, automatikusan szolgáltatja a kívánt megoldást.

A táblázatkezelõ rendszer rendkívül egyszerûen és logikájából eredõen jól alkalmazható a numerikus matematikai problémák zömének a megoldására, mivel azok nagy része bizonyos iterációs lépések sorozatával oldható meg. Ez a feladat pedig a táblázatkezelõ rendszereknél az egy sorban (vagy oszlopban) szereplõ feltétlen, vagy feltételes értékadó utasítások sorozatának az átmásolásával oldható meg. A másolás minden táblázatkezelõ rendszerben rendelkezésre áll. Ha ugyancsak másolással gondoskodunk az iteráció új értékkel történõ végrehajtásáról, akkor a cellák újraszámításával a program automatikusan végrehajtódik és a szóban forgó feladat megoldást nyer.

A dolgozatban bemutatjuk a fenti módszert és a teljesség igénye nélkül megadjuk azokat a legfontosabb eljárásokat, amelyek jól szemléltetik a táblázatkezelõ rendszerek felhasználásának elõnyeit a numerikus matematikai feladatok megoldásánál.

A dolgozatban alkalmazott jelölések:

- Az **i.** sorbeli cellák: **Ai**, **Bi**, **Ci**, **Di**, . . .
- A tartományok: **Ai..Cj**, vagy **ai..cj**.
- Értékadás: **Ai: <kifejezés> (E)**
  Az **Ai** cella felveszi a **<kifejezés>** értékét. Ennek megvalósítása: Általában az **Ai** cellára állva begépelendõ a **<kifejezés>** ENTER-rel zárva.
- A parancsmenü hívása: **/**
- A **b1..c1** forrástartomány másolása **b2..c10** céltartományra:

/ B C b1..c1 (E) b2..c10 (E)

B(lock) **C**(opy) ENTER: **(E)**

2. Nemlineáris egyenletek közelítõ megoldása

2.1. *Newton-Raphson módszer*

Ismeretes, hogy ha **c** egyik gyöke az **f(x)=0** egyenletnek, **f'(c)** nem zérus és **f"(x)** folytonos az **x=c** pontban, akkor az

x–f(x)/f'(x)

iterációs sorozat konvergál a **c** gyökhöz.

Az iterációs lépéseket a táblázat **i.** sorában helyezzük el:

- **Ai: x (E)** constans kezdõérték. Ha az **(a,b)** intervallumban **f'*f"<0**, akkor **x=a**, egyébként **x=b** választandó.
- **Bi: f(Ai) (E)**
- **Ci: f'(Ai) (E)**
- **Di: +Ai–Bi/Ci (E)**
- A **bi..di** tartomány képleteit átmásoljuk a következõ néhány (5-6) sorba:

/ B C bi..di (E) bi+1..di+6 (E)

Mivel az iterációt a **Di**-ben lévõ értékkel kell folytatni, ezért azt az **Ai+1**-nek "átadjuk".

Ai+1: +Di (E)

Az iteráció további folytatását e képlet átmásolásával biztosítjuk:

/ B C ai+1..ai+1 (E) ai+2..ai+7 (E)

A fentiek bemutatására vizsgáljuk meg az alábbi feladatokat:

2.1.1. *Feladat*

Számítsuk ki a

–x+2–log(x)=0

egyenlet gyökét.

Mivel **f(1)=1** és **f(2)<=0**, ezért az **(1,2)** intervallumban van valós gyöke. Az **f'(x)= –1–1/(x*ln(10))** és **f''(x)=1/(ln(10)*x^2)** miatt **f'(x)<0** és **f"(x)>0** minden **x>0**-ra, ezért a gyök kezdõértéke **x=1**. Az **1.** és **2.** sorban készítsünk fejlécet, így **i=3** lesz az algoritmusban.

A3: 1 (E)

	A	B	C	D
1	**x**	**f(x)**	**f'(x)**	**x-f/f'**
2	===========	===========	===========	===========
3	1	1	-1.43429	1.697207
4	1.697207	0.73058	-1.25589	1.75538
5	1.75538	0.000249	-1.24741	-1.755579
6	1.75579	2.82E-09	-1.24738	1.755579
7	**1.755579**			

Az algoritmus alapján a táblázat **3.** sorába a **B C D** oszlopokba a fenti képleteket írva és azokat átmásolva **b4..d6** tartományba:

/ B C b3..d3 (E) b4..d6 (E)

Az eljárás folytatásához az **A4: +d3**–at írva és ezt átmásolva **a5..a7**-be

/ B C a4..a4 (E) a5..a7 (E)

a fenti táblázatot kapjuk. A közelítõ gyök az A oszlopban olvasható le a **B**-oszlopban látható pontossággal.

2.2. *A húrmódszer*

Grafikusan is könnyen bizonyítható, hogy ha egy adott intervallumon az **f''(x)** és **f'(x)** megtartja elõjelét és különbözõ elõjelûek, akkor az intervallum kisebb végpontját választva " **a**"-nak, a nagyobbat pedig " **b**"-nek, ha pedig **f''*f'>0**, akkor fordítva az intervallum nagyobb végpontját választva " **a**"-nak és a másikat " **b**"-nek, a húr mindig **a b** és a **gyök** között metsz, vagyis a " **b**" értékek sorozatával közelíti a **gyököt** nevezetesen a következõ **x** az

x–f(x)(a–x)/(f(a)–f(x))

formulával kapható (**a** és **f(a)** konstans).

Az iterációs lépéseket a táblázat **i.** sorában helyezzük el (szabadon választható):

– **Ai: x** (E) constans.
– **Bi: f(Ai)** (E)
– **Ci: +Ai–Bi*(a–Ai)/(f(a)–Bi)**(E)

A **bi..ci** tartomány képleteit átmásoljuk a következõ néhány (8-9) sorba

/ B C bi..ci (E) bi+1..ci+8 (E)

Mivel az iterációt a **Ci-**ben lévõ értékkel kell folytatni, ezért azt az **Ai+1-**nek "átadjuk".

Ai+1: +Ci (E)

Az iteráció további folytatását e képlet átmásolásával biztosítjuk:

/ B C ai+1..ai+1 (E) ai+2..ai+9 (E)

A fentiek bemutatására vizsgáljuk meg az alábbi feladatot:

2.2.1. *Feladat*

Határozzuk meg az elõzõek során már megoldott

f(x)=–x+2–log(x)

egyenlet gyökét húrmódszerrel !

Példánkban **f(1)=1**; **f(2)=–0.30103**; **f'(x)=1–1/(x*ln(10))** és így **f'(x)<0**; **f''(x)=1/(ln(10)*x^2)** és **f''>0** minden **x>0-**ra.

Mivel **f'*f''<0**, ezért a húrmódszernél az intervallum nagyobb végpontját **x=2-**õt választjuk (**b**-nek) kiinduló gyöknek. Készítsünk fejlécet is, így **i=3.**

	A	B	C
1	x	f(x)	x–f(x)*(1–x)/(1–f(x))
2	=================	=================	=================
3	2	–0. 30103	1. 768622
4	1.768622	–0.01626	1.756326

5	1.756326	–0.00093	1.755622
6	1.755622	–5.4E–05	1.755582
7	1.755582	–3.1E–06	1.75558
8	1.75558	–1.8E–07	1.75558
9	1.75558	–1E–08	1.755579
10	1.755579	–5. 9E–10	1.755579
11	1.755579		

A 3. sorba beírjuk az **A3: 2 (E)**; **B3: –A3+2–@log(A3)**; **C3: +A3–B3*(1–A3)/(1–B3)** kifejezéseket.

A **b3..c3-**at átmásoljuk **b4..c10-**be :

/ B C b3..c3 (E) b4..c10 (E)

Az iteráció az **a4-**ben folytatódik **+c3** értékével. **A4: +C3 (E)**.

Az **a4-**et átmásoljuk **a5..a11-**be:

/ B C a4..a4 (E) a5..a11 (E)

A táblázat ezek után automatikusan kitöltõdik. Látható, hogy már az 5. lépés után eléri a 5 tizedesjegy pontosságot.

2.3. *A fokozatos közelítések módszere*

Az **f(x)=0** egyenlet helyett tekintsük a vele ekvivalens

x=g(x), ahol **g(x)=x+f(x)**

egyenletet. Ha az így generált iterációs sorozat konvergens, akkor az eljárás segítségével tetszõleges pontossággal meghatározható az egyenlet gyöke.

Az iterációs lépéseket a táblázat i. sorában helyezzük el (szabadon választható):

- **Ai: x (E)** constans.
- **Bi: g(Ai) (E)**

A **Bi-**ben lévõ képletet átmásoljuk a következõ néhány (8-9) sorba

/ B C bi..bi (E) bi+1..ci+9 (E)

Mivel az iterációt a **Bi-**ben lévõ értékkel kell folytatni, ezért azt az **Ai+1-**nek "átadjuk".

Ai+1: +Bi (E)

Az iteráció további folytatását e képlet átmásolásával biztosítjuk:

/ B C ai+1..ai+1 (E) ai+2..ai+10 (E)

A fentiek bemutatására vizsgáljuk meg az alábbi feladatot:

2.3.1. *Feladat*

Határozzuk meg az

f(x)=–x+2–log(x)

E zérushelyét iterációs módszerrel.

Tekintsük az **x=g(x)**, azaz **x=2–log(x)** egyenletet. Mivel **ABS(g'(x))=1/ABS(x*LN(10))<1/2 <1** a gyök környezetében (**x>1** miatt) ezért az eljárás konvergál. Kezdõértéknek válasszuk az **x=2** értéket. (**i=3**)

– **A3: 2 (E)**
– **B3: 2–@log(A3) (E)**

Ha az eljárás konvergál ez a módszer a legegyszerûbben megoldható eljárást adja.

	A	B	C
1	**x**	**g(x)**	
2			
3	2	1.698970004336	
4	1.69897	1.769814288621	
5	1.769814	1.752072302937	
6	1.752072	1.756447975726	
7	1.756448	1.75536470906	
8	1.755365	1.755632637217	
9	1.755633	1.755566354216	
10	1.755566	1.755582751089	
11	1.755583	1.755578694826	
12	1.755579		

B3-at átmásoljuk **b4..b11**-be:

/ B C b3..b3 (E) b4..b11 (E)

A4-be beírjuk a **+B3** képletet: **A4: +B3 (E)**. Az iteráció tovább folytatva az **a5..a12** tartományban:

/ B C a4..a4 (E) a5..a12 (E),

amelyre a táblázat kitöltésre kerül és a gyök az **A** oszlopban leolvasható.

2.4. *"Felezési" eljárás*

Tegyük fel, hogy az **(a,b)** zárt intervallumban folytonos **f(x)** függvény helyettesítési értékei az intervallum végpontjaiban ellentétes elõjelûek. Ha az intervallumot felezzük **x=(a+b)/2** és **f(x)=0**, akkor **x** az egyenlet gyöke. Ha viszont **f(x)** nem zérus, akkor **(a,x)**, illetve **(x,b)** intervallumok közül azt választjuk, amelynek végpontjaiban a függvényértékek ellentétes elõjelûek.

Az iterációs lépéseket a táblázat i. sorában helyezzük el:

– **Ai: a (E)** constans

– **Bi: b (E)** constans
– **Ci: (Ai+Bi)/2 (E)**
– **Di: f(Ai) (E)**
– **Ei: f(Ci) (E)**

A **ci..ei** tartomány képleteit átmásoljuk a következõ néhány 15-16 sorba

/ B C ci..ei (E) ci+1..ei+14 (E)

Az iteráció intervallum-választással folytatható. Ha **f(a)*f(x)>0**, akkor az eljárást **a:=x**, egyébként **b:=x** választás mellett folytathatjuk. Ezt az **A** és **B** oszlopok **(i+1)-**ik sorába, az alábbi feltételes kifejezések beírásával érhetjük el:

– **Ai+1: @IF(Di*Ei>0, +Ci, +Ai)**
– **Bi+1: @IF(Di*Ei>0, +Bi, +Ci)**

Az iteráció további folytatását e képletek átmásolásával biztosítjuk:

/ B C ai+1..bi+1 (E) ai+2..bi+14 (E)

és a **C** oszlopban a gyök közelítõ értékeit az **E** oszlopban pedig a pontosságot olvashatjuk le.

A fentiek bemutatására vizsgáljuk meg az alábbi feladatot:

2.4.1. *Feladat*

Határozzuk meg az

f(x)=–x+2–log(x)

zérushelyét felezõ eljárással.

Készítsünk itt is fejlécet és így **i=3** lehet.

Mivel **f(1)=2** és **f(2)<0**, ezért **a=1** és **b=2** intervallumban a függvény folytonossága miatt valós gyöke van. Az **a3-**ba **1-**et **b3-**ba **2-**õt írva **c3-**ba ezen intervallum számtani közepét és **d3**, **e3-**ba az **f(A3)**, illetve **f(C3)** értékét írva:

– **A3:1**
– **B3:2**
– **C3:(A3+B3)/2**
– **D3:-A3+2–@log(A3)**
– **E3:–C3+2–@log(C3)**

Itt is a C, D, E oszloptartományt a harmadik sorból a többibe átmásoljuk:

/ B C c3..e3 (E) c4..e13 (E)

Az iterációt az alábbi értékadással folytatjuk:

- **A4: @IF(D3*E3>0, +C3, +A3)**
- **B4: @IF(D3*E3>0, +B3, +C3)**

Ezt a két cellát szintén átmásoljuk az **A**, **B** oszlop további soraiba

/ B C a4..b4 (E) a5..b13 (E)

és a **C** oszlopban kapjuk a gyök közelítõ értékeit.

	A	B	C	D	E
1	a	b	x=(a+b)/2	f(a)	f(x)
2					
3	1	2	1.5	1	0.323909
4	1.5	2	1.75	0.323909	0.006962
5	1.75	2	1.875	0.006962	–0.148
6	1.75	1.875	1.8125	0.006962	–0.07078
7	1.75	1.8125	1.78125	0.006962	–0.03197
8	1.75	1.78125	1.765625	0.006962	–0.01252
9	1.75	1.765625	1.7578125	0.006962	–0.00279
10	1.75	1.757813	1.75390625	0.006962	0. 002087
11	1.753906	1.757813	1.755859375	0.002087	–0.00035
12	1.753906	1.755859	1.7548828125	0.002087	0.000869
13	1.754883	1.755859	1.75537109375	0.000869	0.00026
14	1.755371	1.755859	1.755615234375	0.00026	–4.5E–05
15	1.755371	1.755615	1.755493164063	0. 00026	0. 000108

Vegyük észre, hogy a konvergáció itt jóval kisebb mint a többi módszernél.

3. Lineáris egyenletrendszerek közelítõ megoldása iterációval

Tegyük fel, hogy az egyenletrendszer eleget tesz az iterációval történõ megoldhatóság feltételének. Ismeretes, hogy az iteráció valamennyi változó **0** értékével indítható.

Írjuk az **A** oszlopba rendre a gyökök azonosítóit, a **B** oszlopba pedig az induló 0, 0, 0, ... gyököket.

- **A1: 'x1=' (E)**
- **A2: 'x2=' (E)**
- **A3: 'x3=' (E)**
- **...**
- **B1: 0 (E)**
- **B2: 0 (E)**
- **B3: 0 (E)**
- **...**
-

A **C** oszlopba írjuk a kifejezett változók jobb oldali részét úgy, hogy az **x1**, **x2**, **x3**, ... változók helyett **A1**, **A2**, **A3**, ... cella neve írandó.

A **C** oszlopba irt kifejezéseket mint iterációs lépéseket a további oszlopokban az elõzõ oszlopok értékeivel megismételve, vagyis **c1..ci** tartomány **d1..ji** tartományra történõ átmásolásával az egyes oszlopokban megkapjuk az egyes gyökközelítéseket. (**j>d** oszlopjelölõ)

/ B C c1..ci (E) d1..ji (E) i>1; j={E, F, G, H, I, J, K, ...}

3.1. *Feladat*

Határozzuk meg az

- **x=0.1 y–0.1 z+4**
- **y=0.1 x–0.1 z+3**

– **z=0.1 y–0.1 u+2**
– **u=0.1 y–0.1 z+1**

egyenletrendszer gyökeit iterációs eljárással.

Írjuk az **A** oszlopba rendre a gyökök azonosítóit, a **B** oszlopba pedig a induló 0, 0, 0, 0 gyököket. A **C** oszlopba beírjuk a lenti táblázatban levõ képleteket:

	A	B	C	D
1	x=	0	0.1*B2–0.1*B3+4	
2	y=	0	0.1*B1–0.1*B3+3	
3	z=	0	0.1*B2–0.1*B4+2	
4	u=	0	0.1*B2–0.1*B3+1	

A **c1..c4** képleteket átmásolva **d1..h4** be az oszlopokban megkapjuk az egymás utáni gyökközelítéseket:

	A	B	C	D	E	F	G	H
1	x=	0	4	4.1	4.1	4.098	4.098	4.09798
2	y=	0	3	3.2	3.19	3.189	3.1889	3.18889
3	z=	0	2	2.2	2.21	2.209	2.2091	2.20909
4	u=	0	1	1.1	1.1	1.098	1.098	1.09798

4. Közönséges differenciálegyenletek közelítõ megoldása

4.1. *Heun-féle módszer*

A Heun-módszer algoritmusa az **y'=f(x,y)** közönséges elsõrendû differenciálegyenlet **y(x0)=y0** kezdõfeltételt kielégítõ numerikus megoldására:

```
x:=x0; y:=y0; h1:=h/2;
DO WHILE x<b
    f1:=f(x, y);
    x:=x+h;
    y:=y+h1*(f1+f(x, y+h*f1);
    PRINT: x, y
ENDDO
```

(**h** a lépésköz)

Megoldás a táblázatkezelõ rendszerrel:

Írjuk be az alábbi értékeket (képleteket) az alant megjelölt rekeszekbe

– **A1: h (E)** constans
– **A2: x0 (E)** constans
– **B2: y0 (E)** constans
– **C2: f(A2, B2) (E)** **(f1=f(x, y))**
– **D2: +B2+$A$1*C2 (E)** **(y1=y+h*f1)**
– **E2: +$A$1*(C2+f(A2+$A$1, D2))/2 (E)** **(h*(f1+f(x+h,y1))/2)**

Másoljuk át a **c2..e2** öt a **C**, **D**, **E** oszlopban a **h** és **b** értéknek megfelelõen:

/ B C c2..e2 (E) c3..e18 (E)

majd írjuk be a következõ **x**, **y** értékeket az **A3** és **B3** rekeszekbe

- **A3: +A2+$A$1 (E)**
- **B3: +B2+E2 (E)**

és a rekurzió miatt átmásolva **a4..b18**-ba

/ B C a3..b3 (E) a4..b18 (E)

az **A** és **B** oszlopokban kapjuk a kívánt eredményt. Az itt közölt program alkalmas bármely más **h** vagy a kezdõértékek megváltoztatása esetén a megoldást szemléletesen azonnal megadni.

4.2. *Feladat*

Határozzuk meg az

y'=x*y^2–y

differenciálegyenlet numerikus megoldásait az **y(0)=1** kezdeti feltétel mellett a **(0, 0.8)** intervallumban, **h=0.05** lépésközzel Heun módszerével!

Írjuk be az alábbi értékeket (képleteket) az alant megjelölt rekeszekbe:

- **A1: 0.05** (E) constans
- **A2: 0** (E) constans
- **B2: 1** (E) constans
- **C2: +A2*B2^2-B2 (E)** **(f1=f(x, y))**
- **D2: +B2+$A$1*C2 (E)** **(y+h*f1)**
- **E2: +$A$1*(C2+(A2+$A$1)*D2^2-D2)/2 (E)** **(h*(f1+f(x+h, y1))/2)**

Másoljuk át a **c2..e2**-õt a **c3..e18** tartományba:

/ B C c2..e2 (E) c3..e18 (E)

majd írjuk be a következõ **x**, **y** értékeket az **A3** és **B3** rekeszekbe

– **A3: +A2+$A$1 (E)**
– **B3: +B2+E2 (E)**

A rekurzió miatt ezeket átmásolva **a4..b18**-ba

/ B C a3..b3 (E) a4..b18 (E)

az **A** és **B** oszlopokban kapjuk a kívánt eredményt.

	A	B	C	D	E
1	**0.05**	**HEUN módszer**	**A1: 0.05 (=h)**	**y'=x*y^2-y**	**y(0)=1**
2	0	1	–1	0.95	–0.04762
3	0.05	0.952378	–0.90703	0.907027	–0.04329
4	0.1	0.909084	–0.82644	0.867762	–0.03953
5	0.15	0.869552	–0.75613	0.831746	–0.03624
6	0.2	0.833314	–0.69443	0.798593	–0.03334
7	0.25	0.799975	–0.63998	0.767975	–0.03078
8	0.3	0.769199	–0.5917	0.739614	–0.0285
9	0.35	0.740703	–0.54868	0.713269	–0.02646
10	0.4	0.714242	–0.51019	0.688732	–0.02464
11	0.45	0.689605	–0.47561	0.665825	–0.02299
12	0.5	0.666611	–0.44443	0.64439	–0.02151
13	0.55	0.6451	–0.41622	0.624289	–0.02017
14	0.6	0.624933	–0.39061	0.605403	–0.01894
15	0.65	0.605989	–0.36729	0.587624	–0.01783
16	0.7	0.588159	–0.34601	0.570858	–0.01681
17	0.75	0.571347	–0.32652	0.555021	–0.01588
18	0.8	0.55547	–0.30863	0.540038	–0.01502

4.3. *Runge-Kutta féle módszer*

A Runge-Kutta féle módszer algoritmusa az

y'=f(x, y)

közönséges elsõrendû differenciálegyenlet **y(x0)=y0** kezdõfeltételt kielégítõ numerikus megoldására:

```
x:=x0;y:=y0;
DO WHILE x<b
    r1:=h*f(x, y);
    r2:=h*f(x+h/2, y+r1/2);
    r3:=h*f(x+h/2, y+r2/2);
```

r4:=h*f(x+h, y+r3);
y:=y+(r1+2*r2+2*r3+r4)/6;
PRINT:x+h, y;
x:=x+h;
ENDDO;

Megoldás a táblázatkezelõ rendszerrel:

Írjuk be az alábbi értékeket (képleteket) az alant megjelölt rekeszekbe

- **A1: h (E)** constans (lépésköz)
- **A2: x (E)** constans **x** kezdõérték **x=x0**
- **A3: y (E)** constans **y** kezdõérték **y=y0**
- **A4: f(A2, A3)*$A1 (E)** **r1=h*f(x, y)**
- **A5: +A2+$A1/2 (E)** **x1=x+h/2**
- **A6: +A3+A4/2 (E)** **y1=y+r1/2**
- **A7: f(A5, A6)*$A1 (E)** **r2=h*f(x1, y1)**
- **A8: +A3+A7/2 (E)** **y2=y+r2/2**
- **A9: f(A5, A8)*$A1 (E)** **r3=h*f(x1, y2)**
- **A10: +A2+$A1 (E)** **x2=x+h**
- **A11: +A3+A9 (E)** **y3=y+r3**
- **A12: f(A10, A11)*$A1 (E)** **r4=h*f(x2, y3)**
- **A13: (A4+2*A7+2*A9+A12)/6 (E)** **y**

Az **A** oszlop képleteit **INT(b/h)** számú oszlopba átmásoljuk:

/ B C a4..a13 (E) b4..h13 (E)

majd áttérünk a következõ iterációra a

- **B2: +A2+$A1 (E)**
- **B3: +A3+A13 (E)**

képletek beírásával, illetve átmásolásával:

/ B C b2..b3 (E) c2..h3 (E)

Az eredmény, vagyis az **(x, y)** értékpárok a **2.** és **3.** sorban olvashatók. A bemenõ adatok **(h, x0, y0)**, az **A1**, **A2** és **A3** cellákban egyszerû felülírással megváltoztatható és a program azonnal meghatározza a közelítõ megoldást. Ha a **h** miatt az oszlopszámokat növelni kell, akkor a másolásokat további oszlopokra is eszközölni kell.

4.3.1. *Feladat*

Határozzuk meg az

y'=x*y^2–y

differenciálegyenlet numerikus megoldásait az **y(0)=1** kezdeti feltétel mellett a **(0, 1.0)** intervallumban, **h=0.2** lépésközzel, Runge-Kutte módszerével!

Írjuk be az alábbi értékeket (képleteket) az alant megjelölt rekeszekbe

- **A1: 0.2 (E)** **(=h)**
- **A2: 0 (E)** **(x)**
- **A3: 1 (E)** **(y)**
- **A4: (A2*A3^2-A3)*$A1 (E)**
- **A5: +A2+$A1/2 (E)**
- **A6: +A3+A4/2 (E)**
- **A7: (A5*A6^2-A6)*$A1 (E)**
- **A8: +A3+A7/2 (E)**
- **A9: (A5*A8^2-A8)*$A1 (E)**
- **A10: +A2+$A1 (E)**
- **A11: +A3+A9 (E)**
- **A12: (A10*A11^2-A11)*$A1**
- **A13: (A4+2*A7+2*A9+A12)/6**

Az **A** oszlop képleteit az **F** oszlopig átmásoljuk:

/ B C a4..a13 (E) b4..f13 (E)

majd áttérünk a következõ iterációra a

- **B2: +A2+$A1 (E)**
- **B3: +A3+A13 (E)**

képletek beírásával, illetve átmásolásával:

/ B C b2..b3 (E) c2..f3 (E)

Az eredmény, vagyis az **(x, y)** értékpárok a **2.** és **3.** sorban olvashatók. A bemenõ adatok **(h, x0, y0)**, az **A1**, **A2** és **A3** cellákban egyszerû felülírással megváltoztathatók és a program azonnal meghatározza a közelítõ megoldást. Ha a **h** miatt az oszlopszámokat növelni kell, akkor a másolásokat további oszlopokra is eszközölni kell.

	A	B	C	D	E	F
1	**0.2**	**h**	**RUNGE–KUTTA**		**y'=x*y^2–y**	**y(0)=1**
2	0	0.2	0.4	0.6	0.8	1
3	1	0.833333	0.714286	0.625001	0.555557	0.500002
4	–0.2	–0.13889	–0.10204	–0.07813	–0.06173	–0.05
5	0.1	0.3	0.5	0.7	0.9	1.1
6	0.9	0.763889	0.663266	0.585939	0.524693	0.475002
7	–0.1638	–0.11777	–0.08866	–0.06912	–0.05538	–0.04536
8	0.9181	0.77445	0.669956	0.59044	0.527865	0.477321
9	–0.16676	–0.1189	–0.08911	–0.06928	–0.05542	–0.04534
10	0.2	0.4	0.6	0.8	1	1.2
11	0.833238	0.71443	0.625179	0.55572	0.50014	0.454661
12	–0.13888	–0.10205	–0.07813	–0.06173	–0.05	–0.04132
13	–0.16667	–0.11905	–0.08929	–0.06944	–0.05556	–0.04545
14						

15	A pontos értékek: **y=1/(1+x)**					
16	0	0.2	0.4	0.6	0.8	1
17	1	0.833333	0.714286	0.625	0.555556	0.5

Mivel a feladat pontos megoldása is ismeretes, ezért a táblázat **16.** és **17.** sorában közölt értékekkel jól összhasonlíthatjuk a közelítõ megoldás **2.** és **3.** sorát. (5 tizedesjegyre pontos a megoldás).