

Demográfiai modellek ([folytatás](#))

4. A teljesebb anyag

4.1. A megoldás egy változata

Alábbiakban az előző gyakorlaton szereplő keretprogramból kapható egy lehetséges megoldást részletezzük. (Ha már a sajátja kész, ugorja át ezt az alfejezetet!) Pirossal a keretből hiányzó kódbetűket emeltük ki.

```

Program Demografia;
(*
  A demográfiai feladat megoldása.
*)
...

Var
  N:TEgyedIndex;
  K:TKorcsoportIndex;
  egyedek:TEgyedek;
  szulRatak,
  halRatak:TValosKCs;
  kCsElo:TEgeszKCs;
  T:LongInt;
  leptek,           {a képernyőre normáláshoz}
  felSzorz:Real;   {a tároltból a tényleges létszám meghatározásához}
  Tulnepesedes:Boolean;

Procedure Param;
...

Function Folytatas:Boolean;
...

Procedure KorCsopKi;
...

Procedure SzimulaciosLepes;
  Const
    szelRata = 0.8; {a tömörítéskor alkalmazott szelekciós ráta}
  Var
    gyDb:TEgyedSzam;
    i,j:TEgyedIndex;

  Procedure Hullatakaritas(Var N:TEgyedIndex);
  {hullatakarítás: 0-k kihagyása az egyedek[1..N] tömbből
    N = tömör egyedek tömb hossza
    számlálás: kCsElo[1..K] = korcsoportok gyakorisága}
  Var
    i,j:TEgyedIndex;
  Begin
    For j:=1 to K do kCsElo[j]:=0;

```

```

j:=0;
For i:=1 to N do
Begin
  If egyedek[i]>0 then
  Begin
    Inc(kCsElo[egyedek[i]]);
    Inc(j); egyedek[j]:=egyedek[i];
  End
End;
N:=j
End;

Procedure AranyosTomorites;
{Átlagosan minden x. marad meg, a többi "meghal";
 felSzorz = felSzorz/x
 most x=1/(1-szelRata)=5}
Var
  i:TEgyedIndex;
Begin
  GotoXY(65,11); Writeln('Túlnépesedés!'); BillreVar;
  For i:=1 to N do
  Begin
    If (egyedek[i]>0) and (Random<szelRata) then egyedek[i]:=0
  End;
  Hullatakaritas(N);
  felSzorz:=felSzorz*(1-szelRata);
End;

Begin {SzimulaciosLepes}
gyDb:=0;
for i:=1 to N do
Begin
  If Random<halRatak[egyedek[i]] then {meghalt}
  Begin
    egyedek[i]:=0;
  End
  Else
  Begin
    If Random<szulRatak[egyedek[i]] then {szült => gyerekszámítás}
    Begin
      Inc(gyDb);
    End;
    Inc(egyedek[i]); {öregszik}
  End;
End;
{hullatakarítás és számlálás:}
Hullatakaritas(N);
While N+gyDb>=MaxN do {túlnépesedés}
Begin
  AranyosTomorites;
  {a gyerekek szelektálása:}
  gyDB:=Round((1-szelRata)*gyDb);
End;
For i:=1 to gyDb do egyedek[N+i]:=1;
kCsElo[1]:=gyDb;
N:=N+gyDb;
End;

```

Begin
 ...
 End.

4.2. A második megoldás egy változata

A második változatban a születés naiv megoldását helyettesítjük egy valósághűbel, a **Poisson-eloszlással** leírt véletlen **gyermekszámmal**.

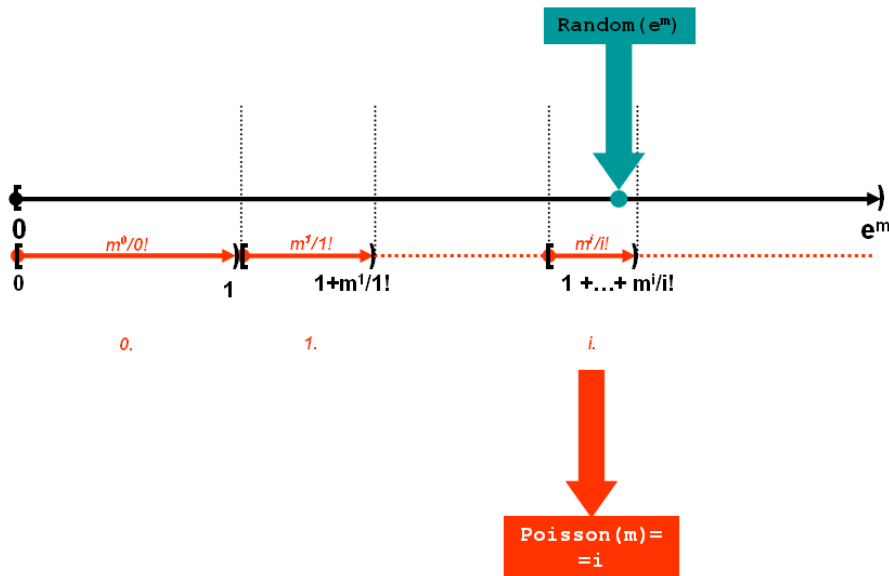
A lényeg: most nem egy valószínűség írja le az 1 gyerek születését, hanem a születendő gyerekek számát generáljuk (a demográfusok szerint jól működő) Poisson-eloszlással. Ehhez módosítani kell a programot a paraméterbeolvasásnál és a szimulációs lépésnél. Az utóbbi az érdekes természetesen: egyszerűen a Poisson-eloszlás „beépített függvényének” felhasználásával a gyerekek számát növelni kell (nem eggyel és nem egy valószínűségtől függően).

Érdekes röviden belepillantani a Poisson-eloszlású véletlenszám generálásába, elvileg és „programozásilag”. Elvi tudnivaló ennyi: a Poisson-eloszlás egy természetes szám értékű, egyparaméteres valószínűségi változó eloszlása. Ez a valószínűségi változó rendszerint egy esemény ismétlődés-számát írja le ¹:

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}, \quad k = 0, 1, 2, \dots; \lambda \text{ a paramétere.}$$

A paraméter az ismétlődés-szám várható értéke (és melleleg szórásnégyzete). Ez a paraméter köti az egyes korcsoportokhoz, ezt kell a korábbi valószínűségek helyett beolvasni.

Programozás logikájához értsük meg az alábbi ábrát!



8. ábra. A Poisson-eloszlás „szimulálásának” magyarázata. (A fenti formula λ -ját m -mel jelöltük az ábrán.

L. még: <http://people.inf.elte.hu/szlavi/InfoRendsz/Szimulacio/PoissonAbra.ppt>)

¹ <http://hu.wikipedia.org/wiki/Poisson-eloszl%C3%A1s>

És végül maga a Pascal függvényeljárás:

```

{
  Poisson-eloszlású véletlenszámot generáló függvény. Paramétere a
  várhatóértéke (és szórásnégyzete).
  Értéke: 0..MaxInt
}
Function Poisson(m:Real):Integer;
  Var
    i:Integer;
    r,t,s:Real;
Begin
  r:=random;
  i:=0; t:=exp(-m); s:=t;
  While r>=s do
  Begin
    i:=succ(i); t:=t*m/i; s:=s+t;
  End;
  Poisson:=i;
End;

```

A szimulációs programunk lényegi részei következnek. (A **piros** az újdonság. Jól látszik ez az általánosítás felé tett lépés algoritmus-egyszerűsödéssel járt!)

```

Program DemografiaP;
(*
  A demográfiai feladat megoldása.
  Poisson-eloszlással.
*)
...

Procedure SzimulaciosLepes;
  Const
    szelRata = 0.8; {a tömörítéskor alkalmazott szelekciós ráta}
  Var
    gyDb:TEgyedSzam;
    i,j:TEgyedIndex;

  Procedure Hullatakaritas(Var N:TEgyedIndex);
    ...

  Procedure AranyosTomorites;
    ...

Begin {SzimulaciosLepes}
  gyDb:=0;
  For i:=1 to N do
  Begin
    If Random<halRatak[egyedek[i]] then {meghalt}
    Begin
      egyedek[i]:=0;
    End
    Else
    Begin
      Inc(gyDb,Poisson(szulRatak[egyedek[i]]));
      Inc(egyedek[i]); {öregszik}
    End;
  End;
End;

```

```

{hullatakarítás és számlálás;}
Hullatakaritas(N);
While N+gyDb>=MaxN do {túlnépesedés}
Begin
  AranyosTomorites;
  {a gyerekek szelektálása;}
  gyDB:=Round((1-szelRata)*gyDb);
End;
For i:=1 to gyDb do egyedek[N+i]:=1;
  kCsElo[1]:=gyDb;
  N:=N+gyDb;
End;

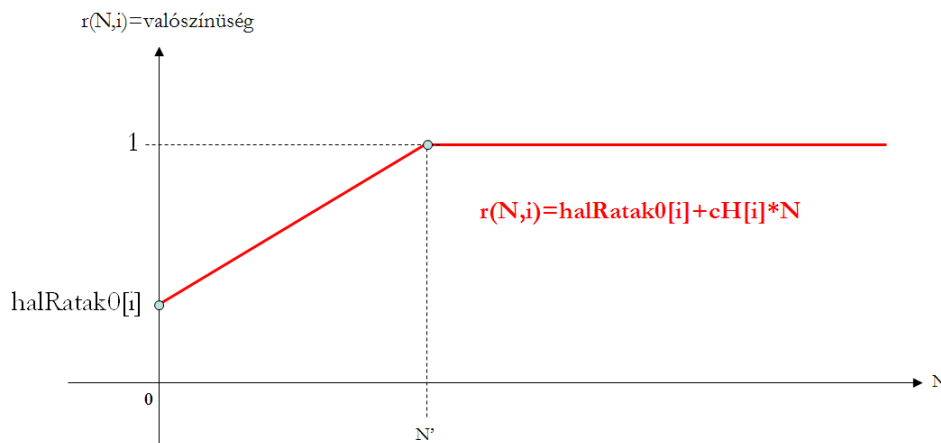
Begin
  ...
End.

```

4.3. A harmadik megoldás egy változata

Ebben **egyedszámtól² függővé** tesszük a paramétereket. Mind a szül(et)ési ráták (Poisson-várhatóértékek), mind a halálozás ráták (valószínűségek) legyenek –egyszerűség kedvéért– az összlétszám **lineáris függvényei**.

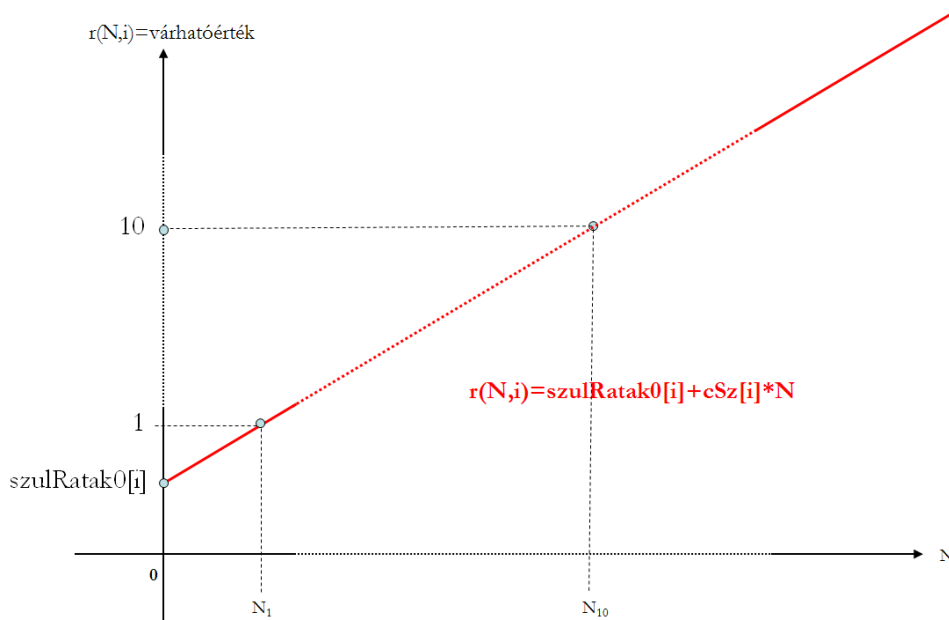
Ehhez az egyes korcsoportokhoz tartozó rátákat, mint függvényeket, már két paraméter fogja meghatározni: egy a növekedés ütemét és egy kezdeti értéket maghatározó paraméter. L. az alábbi ábrákat.



9. ábra. A halálozási ráta függése az összlétszámtól.
(Egy ideig növekszik a valószínűség, majd N' -től stagnál, 1-re beáll.)

² Bár tehetnénk ugyanígy pl. az időtől is.

Ehhez hasonló a Poisson-eloszlás paraméterének a létszámfüggő változata, csak –értelemszerűen– nincs felső korlát, hiszen ez a paraméter várhatóértéket jelent, amely akármekkora pozitív szám lehet.



10. ábra. A születési ráta függése az összlétszámtól.

A módosult szimulációs program (érdekes részletei):

```

Program DemografiaPF;
(*
  A demográfiai feladat megoldása.
  Poisson-eloszlással.
  Függvényes paraméterrel.
*)
...

Var
...
  szulRatak0,cSz,
  halRatak0,ch:TValosKCs;
...

Procedure Param;
  Var
    i:TEgyedIndex;
Begin
  ...
  Writeln(CrLf,'Halálozási ráták (h0 és szorzó):');
  For i:=1 to K-1 do
  Begin
    Readln(fBe,halRatak0[i],ch[i]);
    If fNBe<>'CON' then
    Begin
      If (i Mod 5)=0 then Writeln;
    
```

Demografia2 / 2012.12.07.

```
    Write(halRatak0[i]:8:5,cH[i]:8:5);
End;
End;
If fNBe<>'CON' then Writeln;
halRatak0[K]:=1;
Writeln(CrLf,'Szülési ráták (r0 és szorzó):');
For i:=1 to K do
Begin
    Readln(fBe,szulRatak0[i],cSz[i]);
    If fNBe<>'CON' then
    Begin
        If (i Mod 5)=0 then Writeln;
        Write(szulRatak0[i]:8:5,cSz[i]:8:5);
    End;
End;
If fNBe<>'CON' then Writeln;
If fNBe<>'CON' then BillreVar;
...
End;

Function Folytatas:Boolean;
...

Procedure KorCsopKi;
...

Procedure SzimulaciosLepes;
Const
    szelRata = 0.8; {a tömörítéskor alkalmazott szelekciós ráta}
Var
    gyDb:TEgyedSzam;
    i,j:TEgyedIndex;

Function szulRatak(const i:Integer):TVarhatoertek;{Poisson!}
Var
    r:Real;
Begin
    r:=szulRatak0[i]+cSz[i]*N;
    If r<0 then szulRatak:=0 else szulRatak:=r
End;

Function halRatak(const i:Integer):TValoszinuseg;
Var
    r:Real;
Begin
    r:=halRatak0[i]+cH[i]*N;
    If
        r>1 then halRatak:=1 else if
        r<0 then halRatak:=0
        else halRatak:=r
    {EndIf};
End;

Procedure Hullatakaritas(Var N:TEgyedIndex);
...

Procedure AranyosTomorites;
...
```

```

Begin {SzimulaciosLepes}
  gyDb:=0;
  For i:=1 to N do
  Begin
    If Random<halRatak(egyedek[i]) then {meghalt}
    Begin
      egyedek[i]:=0;
    End
    Else
    Begin
      Inc(gyDb, Poisson(szulRatak(egyedek[i])));
      Inc(egyedek[i]); {öregszik}
    End;
  End;
  {hullatakarítás és számlálás:}
  Hullatakaritas(N);
  While N+gyDb>=MaxN do {túlnépesedés}
  Begin
    AranyosTomorites;
    {a gyerekek szelektálása:}
    gyDB:=Round((1-szelRata)*gyDb);
  End;
  For i:=1 to gyDb do egyedek[N+i]:=1;
  kCsElo[1]:=gyDb;
  N:=N+gyDb;
End;

```

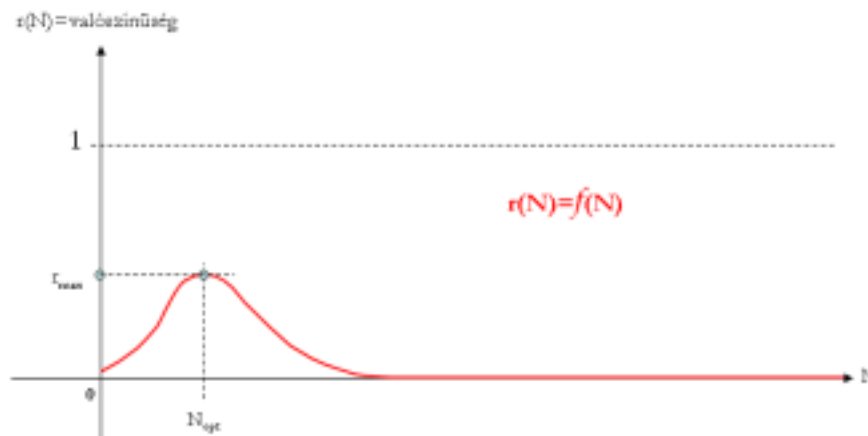
Begin

...

End.

A rátákra vonatkozó összefüggéssel érdemes eljátszozni. Megfigyelték, hogy a *születési ráta* mind kis, mind nagy egyedszám mellett viszonylag kicsi, s van egy *optimális népesség*, amely mellett *maximális*. A *halálozási rátára* is hasonlóan igaz, hogy az *optimális népesség* mellett a *legkisebb*, s minél nagyobb a tőle való távolság, annál jobban nő.

A modell is arról számol be, hogy ilyen változású ráták stabilizálják a populációt, azaz egy egyensúlyi létszám felé terelik őt.



11. ábra. A születési ráta függése az összlétszámtól.

5. Feladatok

Tegye alkalmassá a programot arra, hogy nemhez fűződő jelenségeket (pl. betegségeket, háborúk stb.) lehessen vizsgálni.