

# GÁZ-MODELL

*Sztlávi Péter*

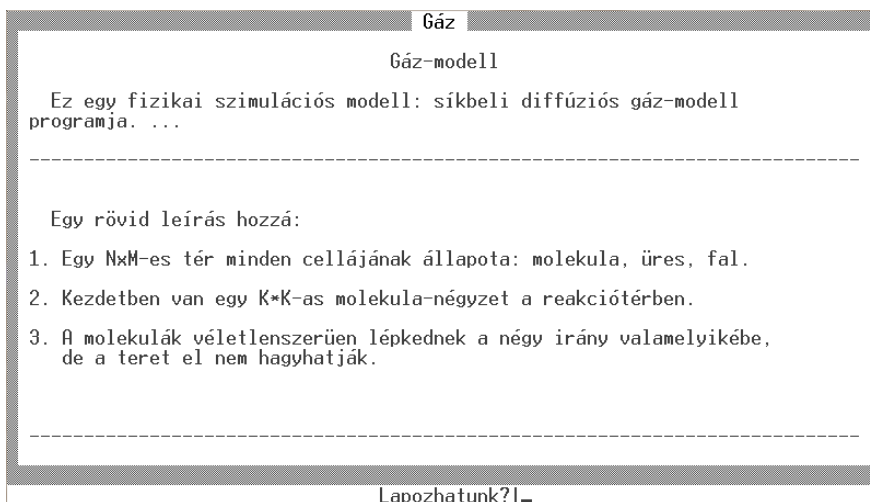
2000-2012

## TARTALOM

GÁZ-MODELL .....	1
1. Elvárások .....	2
2. A problémák .....	4
2.1. A reprezentációs lényeg – Ablakok unit .....	4
2.2. Az implementált műveletek fejsorai – Ablakok unit .....	4
2.3. Szimulációs keret .....	6
2.4. Szimulációs lépés – a lényeg.....	6
2.4.1. A szimulációhoz tartozó reprezentáció .....	6
2.4.2. A szimulációhoz tartozó implementáció .....	7
3. A keretprogram .....	7
4. Feladatok .....	7
4.1. Alapfeladat .....	7
4.2. Modellezési feladatok .....	7
4.2.0. Megfigyelések – megfontolások .....	7
4.2.1. Hő .....	10
4.2.2. Fal a reakciótérben .....	11
4.3. Szorgalmi feladat.....	11
5. A teljes anyag .....	12

## 1. Elvárások

Az elvárásokat legjobban az alábbi, futás során keletkezett ábrásor fejezi ki.

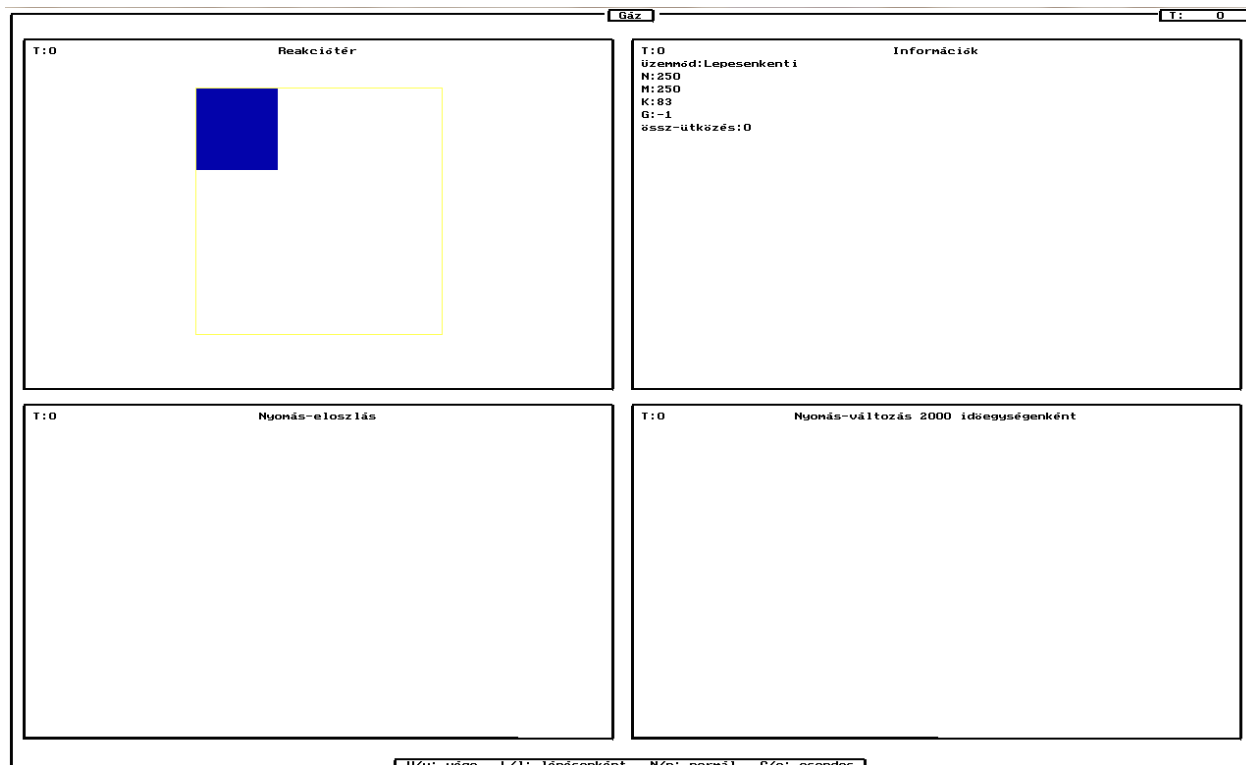


**1. ábra. Egy futási kép – Tájékoztató**

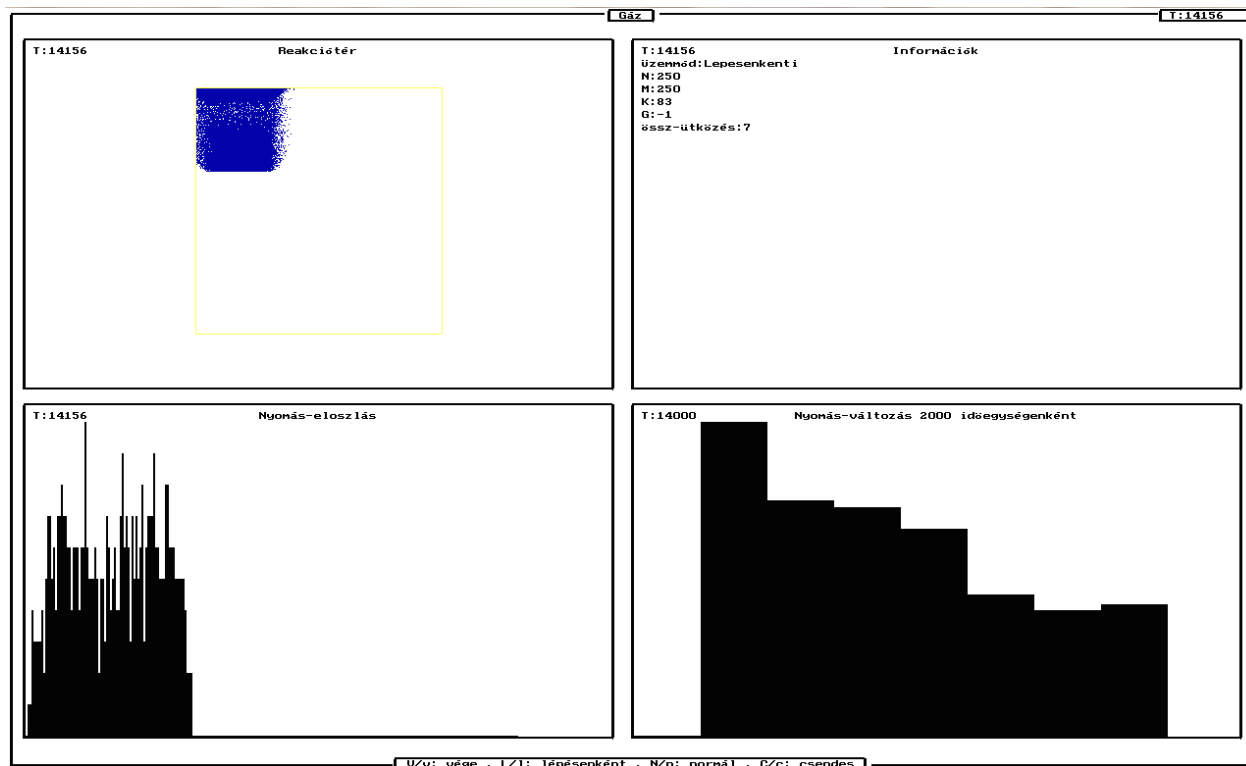


**2. ábra. Egy futási kép – Paraméterezés (maximális méretek, „antigravitációval”)**

## Gáz-modell



3. ábra. Egy futási kép – A szimuláció legelején (T=0)



4. ábra. Egy futási kép – Közbülső állapot (T=14156)

valamint egy próba: [GazMo.exe](#).

## 2. A problémák

A Darázs-modell alapján elkészült szimulációs keretprogram azon részeit, amelyek a továbbiakban –lényegileg vagy ténylegesen– változatlanul felhasználhatók kiemeltük az `Ablakok` unitba. Ebbe jobbra a megjelenítéssel kapcsolatos reprezentációt és implementációt rejtettük el.

### 2.1. A reprezentációs lényeg – Ablakok unit

Type

```

TAbalak=Record
    bfx, bfy, jax, jay : Word;
    cim                : String[32];
    szin               : Byte;
End;

THisztAbalak=Record
    bfx, bfy, jax, jay : Word;
    cim                : String[32];
    adatDb             : Word;
    leptek             : Real;
    adatok             : Array [0..MaxDb] of Word;
    szin               : Byte;
End;

TKarAbalak=Record
    bfx, bfy, jax, jay : Word;
    cim                : String[32];
    adatDb             : Word;
    adatok             : Array [0..MaxMdb] of
                        Record
                            nev,ert:String;
                        End;
    szin               : Byte;
End;

TCella =(Fal,Ures,Reszecske);
TReaTer=Array [0..MaxN+1,0..MaxM+1] of TCella;

```

Megjegyzések:

1. Az ablakok fajtája bővült: a hisztogramokat tartalmazó (`THisztAbalak`) mellé, mint típus került be a reakcióteret tartalmazó `TAbalak`, továbbá a jellemző modell-paraméterek numerikus értékeit folyamatosan mutató információs ablak, a `TKarAbalak` (mint karakteres ablak). Ezekhez sajátos műveletek tartoznak –természetesen–, amint ezt alább látni is fogjuk.
2. A reakcióterben a részecske és annak hiánya mellett lényeges szerepe lesz a későbbiek során a `fal`nak, amely nem feltétlenül csak a tér befoglalójául használható föl.
3. A reakcióteret körül vesszük fallal. Ezeket helyezzük el a `0.` és `MaxN+1`-dik sorba, és a `0.` `MaxM+1`-dik oszlopba.

### 2.2. Az implementált műveletek fejsorai – Ablakok unit

Elsőként a hagyományos I/O-műveletek (a konzol-ablak kezelését végzők) szerepelnek:

```

Procedure BillentyureVar (Var c: Char);
Procedure SzovegesLap (Const Fejlec:String);
Procedure Tajekoztato (Const Fejlec,TajekoztatoFile:String);

```

Az egyes ablakokban megjelenő modell-időt kiíró eljárások, amelyek azt mutatják, hogy mikori állapotú az ábra. Ez azért fontos, mert az időspórolás miatt nem minden időegységben jelenik meg minden.

```

Procedure FoIdoKiiras ({Const FoAblak:TFoAblak;} Const T:LongInt);
Procedure ReaIdoKiiras (Const TerAblak:TAblak; Const T:LongInt);
Procedure IdoKiiras (Const Graf:TThisztAblak; Const T:LongInt);

```

Az alábbiak létrehozzák az adott típusú ablak egy újabb változatát. Még nem jelentetik meg a képernyőn.

```

Procedure KarAblakLetrehozas (Var KarAblak:TKarAblak;
                               Const bx,by,jx,jy:Word;
                               Const c:String; Const sz:Byte);
Procedure ReaAblakLetrehozas (Var TerAblak:TAblak;
                               Const bx,by,jx,jy:Word;
                               Const c:String; Const sz:Byte);
Procedure AblakLetrehozas (Var Graf : TThisztAblak;
                              Const bx,by,jx,jy:Word;
                              Const c:String; Const aDb:Word;
                              Const l:Real; Const sz:Byte);

```

A következők az ablakok megjelenítését végzik, kezdve a „fő” képernyő változatlan részeinek kirajzolását végzővel:

```

Procedure GrafikusKepernyoAllandoReszei (Const Fejlec:String);
Procedure KarAblakRajzolas (Const KarAblak:TKarAblak);
Procedure ReaTerAblakRajzolas (Const TerAblak:TAblak);
Procedure AblakRajzolas (Const Graf:TThisztAblak);

```

Az egyes ablakok újrarajzolását végző rutinok (ezek csak a változó részeket rajzolják ki):

```

Procedure ReaTerUjraRajzol (Const TerAblak:TAblak;
                              Const ReaTer:TReaTer; Const N,M:Integer;
                              Const T:LongInt);
Procedure TeljesGrafikon (Var Graf:TThisztAblak);
Procedure EloszlasFv (Szam:Word; Var Graf:TThisztAblak; Const T:LongInt);
Procedure IdoDiagram (Szam:Word; Var Graf:TThisztAblak; Const T:LongInt);
Procedure ReszecskeRajzolas (Const TerAblak:TAblak; Const N,M:Integer;
                              Const i,j:Integer; Const c:TCella);

```

Végezetül a karakteres (info-) ablak mezőkkel ellátó eljárások következnek:

```

Procedure KarAblakUjMezo (Var KarAblak:TKarAblak; Const nev,ert:String);
Procedure KarAblakUjErtek (Var KarAblak:TKarAblak; Const nev,ert:String);

```

Segéd függvények:

```

Function TP2FP_Y (y:Integer) :Integer;
Function TP2FP_X (x:Integer) :Integer;

```

Néhány megjegyzés:

1. A TP2FP\_X és TP2FP\_Y függvény egy „történelmi emlék”. Arra emlékeztet, hogy a TP-világban a képernyőméret meglehetősen korlátozott volt, és az FP-re való áttérés konverzióját valószínűsíti meg.
2. Nem jelenik meg a Tájékoztató eljárás „párja”, a futás legvégén szerephez jutó összefoglalóMegjelenítés. Ok: ezt szükséges hangolni a konkrét szimulációs modellhez, így a fő program részeként jelenik meg. (El lehet gondolkodni az általánosításon, de ettől most eltekintek.)

## 2.3. Szimulációs keret

A modellezés bemeneti adatai:

- $T$  – idő
- $N, M$  – reakciótér-méret
- $K$  – a kezdetben elhelyezendő részecskék száma (ami most a modell működése során nem fog változni)
- $G$  – gravitációs együttható (-1..+1)

A nagyvonalú algoritmus (nem változik a korábbihoz képest):

```

üzemMod:=Lépesenkénti
Tájékoztató
Inicializálás; T:=0
KezdőKépernyő
EredményMegjelenítés
Ciklus amíg nem VégeE
    T:+1
    SzimulációsLépés
    EredményMegjelenítés
Ciklus vége
ÖsszefoglalóMegjelenítés

```

## 2.4. Szimulációs lépés – a lényeg

A szimuláció során figyeljük a molekulák „szabad” mozgását. A mozgást persze befolyásolja a közvetlen szomszédság, és a gravitáció. Oda nem léphet molekula, ahol már van molekula. A gravitációt sztochasztikusan vesszük számításba: nagyobb gravitáció esetén nagyobb eséllyel lép lefelé. A gravitációt számszerűsítjük: -1..+1. -1 esetén maximális „sebességgel” (1 valószínűséggel) lépjen felfelé. +1 esetén maximális „sebességgel” (1 valószínűséggel) lépjen lefelé, 0 esetén azonos a fel és lefelé mozgás esélye.

### 2.4.1. A szimulációhoz tartozó reprezentáció

#### Konstans

IdoDiagValt:Egész(2000) [ennyi időegységenként rajzolja az időDiagramot  
összük-öt ennyi időre számolja]

#### Változó

```

[Ablakok:]
TérAblak: TAblak      [Reakciótér]
KarAblak: TKarAblak  [Karakteres információk]
Graf1,
Graf2   : THisztAblak [Hisztogramok]
[Szimulációs adatok: paraméterek:]
T       : Egész [Kellően nagy, mert hosszú futásra kell számítani]
N,M,K   : Egész
ReaTér  : TReaTér
voltÜtk: Logikai [volt-e ütközés ebben az időegységben a bal falon?]
ütkY    : Egész [ha volt ütközés, akkor ebben a magasságban]
összÜtk: Egész [az ütközések össz száma]

```

...

## 2.4.2. A szimulációhoz tartozó implementáció

**Eljárás** SzimulációsLépés:

**Változó**

i, j,  
ii, jj: Egész

**Ciklus**

[mozgó részecske: (i, j)]  
i:=Random(N)+1; j:=Random(M)+1  
[potenciális véletlen szomszéd: (ii, jj)]  
jj:=j+Random(3)-1

**Elágazás**

R<(1-G)/3 **esetén** ii:=i-1  
R<(2-G)/3 **esetén** ii:=i  
**egyéb esetben** ii:=i+1

**Elágazás vége**

**amíg** ReaTér(i, j)≠Részecske **vagy** ReaTér(ii, jj)≠Üres (\*)

**Ciklus vége**

ReaTér(i, j):=Üres; ReaTér(ii, jj):=Részecske (\*\*)

RészecskeRajzolás(TérAblak, N, M, i, j, ReaTér(i, j))

RészecskeRajzolás(TérAblak, N, M, ii, jj, ReaTér(ii, jj))

[bal fali ütközés-kezelés:]

voltÜtk:=(jj=1) [**és** (j=2)]

**Ha** voltÜtk **akkor**

ütkY:=ii; összÜtk:=+1

**Elágazás vége**

**Eljárás vége**

Megjegyzés: a **pirossal** jelölt rész egy szigorúbb ütközés többlet feltétele. A (\*)-gal és (\*\*)-gal hivatkozunk később erre a feltételre.

## 3. A keretprogram

L. [GazKe.pas](#), [Ablakok.pas](#).

## 4. Feladatok

### 4.1. Alapfeladat

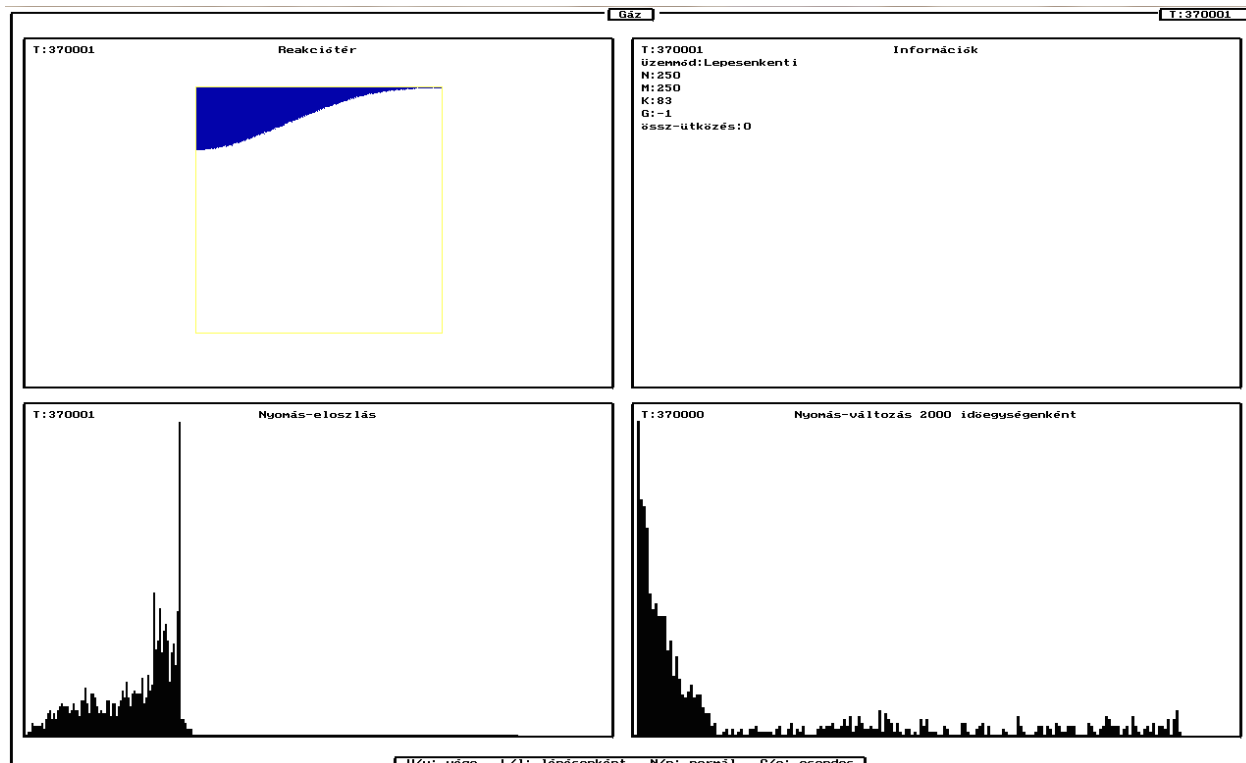
A keretprogram hiányzó részeit pótolni, azaz '**(\*ide kell a kód\*)**' megjegyzéssel megjelölt részek kitöltése:

- SzimulaciosLepes-ben
- OsszefoglaloMegjelenites bővítése a KonkluzioFajl-lal összefüggésben

### 4.2. Modellezési feladatok

#### 4.2.0. Megfigyelések – megfontolások

Figyelje meg, az ábrákon látható paraméterezés mellett hogyan alakul a nyomás időbeli változása! Ugye nem tetszik Önnek? Nekem sem. (L. az 5. ábrát.)



5. ábra. Egy futási kép – Közel a végállapothoz (T=370001)

Józanész alapján megmagyarázhatatlan, hogy miért csökkent le az össz-nyomás (kb. 80000 időegység tájékán)! Ugye világos? Az a „logikus” algoritmikus ötletünk, hogy csak azt tekintjük eseménynek, hogy egy részecske valóban odébb megy (részecske $\leftrightarrow$ üres csere), megakadályozza, hogy a falra tapadt részecske-kupac fal melletti molekulái is „izgehessenek-mozoghassanak”, pedig ezt érzékelhetnénk „nyomásként” a modell szerint. Javítsa ki ezt a hibát!

A megoldás:

`amíg` ReaTér(i,j) $\neq$ Részecske **vagy** ReaTér(ii,jj)=Fal (\*)

`Csere`(ReaTér(i,j),ReaTér(ii,jj)) (\*\*)

L. [GazMo2.exe](#).

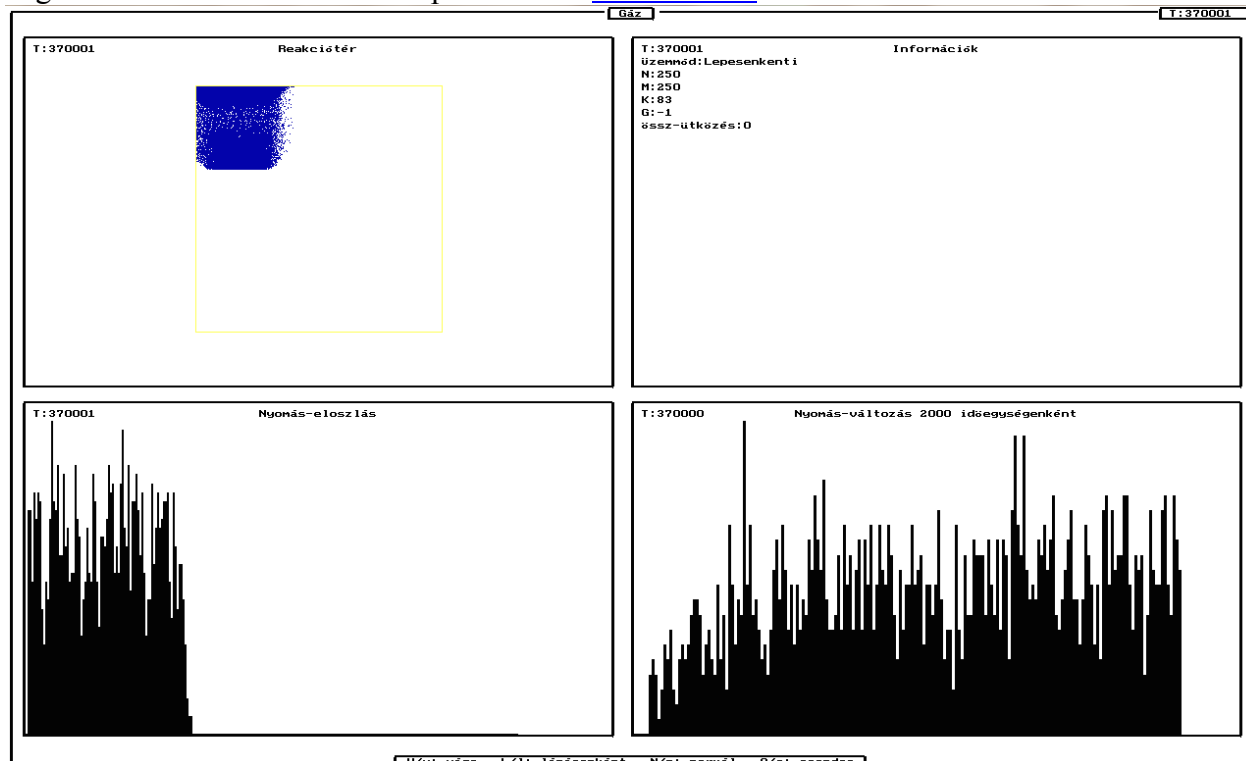
A javított megoldás előbbihez közeli állapotát a 6. ábra mutatja. Sajnos ezzel is –láthatólag– baj van. A nyomás-változás rendben van, de a végállapothoz való közeledés jócskán **lelassult**: hiszen a részecske-kupac belsejében levő részecskepárok cseréje is „eseménnyé” vált, azaz sokkal több időegységet kell várni a részecske várható végállapotának eléréséhez. Mennyi a lassulás? (L. a 7. ábrát.) Az alapmegoldásban csak a „nyitott” felszín változásai jelentettek eseményt. Most majdnem minden részecske esemény részese lehet. Azaz kb. **négyzetes a lassulás**. Ezek után már nem csodálkozhatunk a részecske-kupac felszínének csekély változásán, a reakciótér-konvergenciájának lecsökkenésén. A sebességen növelni lehet az `IdoDiagValt` konstans pl. megtízszerezzük.

Elgondolkozhatunk, hogy vajon a kezdő részecske-négyzet tekinthető-e egyáltalán gáz-szerűnek. Valószínűleg ilyen „sűrű” gázból kiindulni természetellenes. Újabb modell-változathoz jutunk, ha szabályos részecske-alakzat gyanánt pl. olyant választunk, amely azért kellően „liha”, de sza-

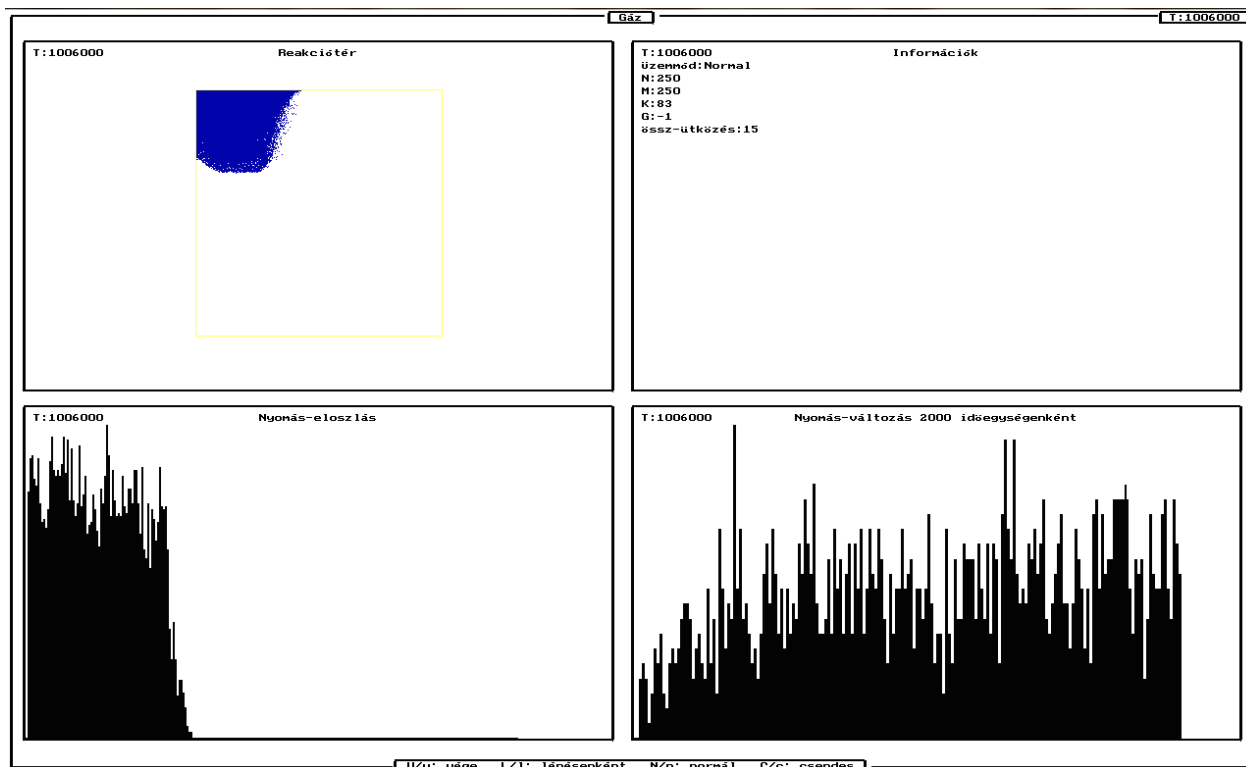


## Gáz-modell

bályos. Így a diffúzió „működését” azon is jól lemérhetjük, miközben elkerüljük a fentebb taglalt negatívumokat. Ezt is érdemes kipróbálni! L. [GazMo3.exe](#).



6. ábra. A módosított program egy futási képe – Közel (?) a végállapothoz (T=370001)



7. ábra. A módosított program egy futási képe – Közel (?) a végállapothoz (T=1006255)

Baj van azonban **elvileg** is ezzel a részecske-részecske cserével. Ugyanis **elrontja a gravitáció** okozta esetleges **anizotrópiát**. Egy szélsőséges példával könnyű belátni ezt. Képzeljük el a  $G=-1$  esetet! Ekkor bár a kiválasztott részecske engedelmeskedve a szélsőségesen nagy „antigravitációnak” feljebb lép, addig a „társa”, fittyet hányva a rá éppen úgy érvényes erőre, lefele lép. Tehát a (\*)-os és (\*\*)-os módosítást mégsem fogadhatjuk el. Keressünk más megoldást!

Az eredeti probléma az volt, hogy a falra tapadt tömbben se belül, se a fal mellett mozgásra kísérletet sem tehetett. Értelmezzük át némileg a „nyomásmérést”. Legyen a nyomást okozó jelenség az, amikor egy részecske egy falelemmel akarna helyet cserélni. A helycsere ugyan tiltott, de a nyomásváltozást érzékelhetjük. A szimulációs lépés alábbi módon alakul:

**Ciklus**

```
[mozgó részecske: (i,j)]
i:=Random(N)+1; j:=Random(M)+1
[potenciális véletlen szomszéd: (ii,jj)]
jj:=j+Random(3)-1
```

**Elágazás**

```
R<(1-G)/3 esetén ii:=i-1
R<(2-G)/3 esetén ii:=i
egyéb esetben ii:=i+1
```

**Elágazás vége**

**amíg** ReaTér(i,j)≠Részecske **vagy** ReaTér(ii,jj)=Részecske

**Ciklus vége**

**Ha** ReaTér(ii,jj)=Fal **akkor** [mozgás nincs,  
csak esetleges nyomás-adminisztrálás]

```
[bal fali ütközés-kezelés:]
```

```
voltÜtk:=(jj=0) [és (j=2)]
```

**Ha** voltÜtk **akkor**

```
ütkY:=ii; összÜtk:+1
```

**Elágazás vége**

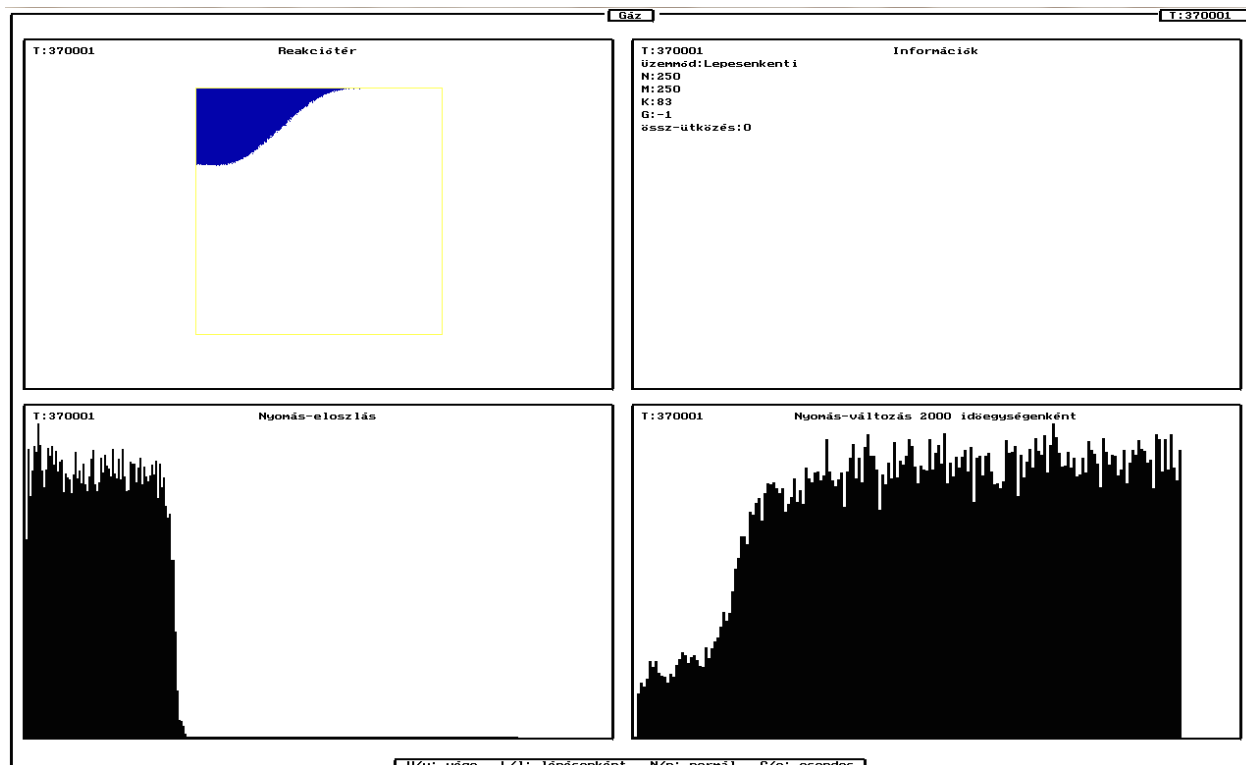
**különben** [mozgás, nyomás-okozás nélkül]

```
ReaTér(i,j):=Üres; ReaTér(ii,jj):=Részecske
```

```
RészecskeRajzolás(TérAblak,N,M,i,j,ReaTér(i,j))
```

```
RészecskeRajzolás(TérAblak,N,M,ii,jj,ReaTér(ii,jj))
```

**Elágazás vége**



8. ábra. Egy futási kép a nyomás-mérés szempontjából is helyes szimulációjú modellben – Jó úton a végállapot felé (T=370001)

L. [GazMo4.exe](#).

#### 4.2.1. Hő

Hogyan vinné a modellbe a **hőt**? Legyen a reakciótér hőmérséklet szempontjából nem „egyenletes” (eddig ilyen volt), azaz pl. a) az alsó harmadában nagyobb hőmérséklet uralkodna, mint a felette levő térrészben, vagy b) a magasságtól függően egyenletesen nőne a hőmérséklet.

**A megoldás ötlete:** ahol magasabb a hőmérséklet, ott intenzívebben mozognak a molekulák, vagyis az ottaniakat nagyobb eséllyel kell tudni kiválasztani, azaz a kiválasztandó molekula Y-koordinátája az Y-tól függően más és más kell legyen.

#### 4.2.2. Fal a reakciótérben

Helyezzen el „válaszfalat” a reakció térben. Ügyelnie kell, hogy a véletlen molekulaválasztáskor a falat nehogy „lebontsa”! Érdekes problémák kapcsolódhatnak a közbülső falhoz:

- mindkét oldalról érzékelje a molekulák ütközését, ez modellezi a nyomást; bizonyos mennyiségű különbség a balról, ill. a jobbról érő nyomás hatására legyen képes odébb csúszni!
- a fal legyen félig áteresztő; most is a kétoldali nyomás különbsége az érdekes (ozmózis).

#### 4.3. Szorgalmi feladat

Az ütközések érzékelése és megjelenítése helyett a szintenkénti **sűrűséggel** foglalkozzon! Vagyis valahány időegységként számlálja le az egyes magassági szinteken lévő molekulák számát, és jelenítse meg az alsó baloldali (most Nyomás-eloszlás című) hisztogram-ablakban! A jobboldali párjában pedig egy Ön által kiválasztott szint (esetleg szintintervallum, pl. az alsó harmad) molekulaszám-változását.

## *5. A teljes anyag*

Ez a kézirat, források, lefordított kódok, súgófájl...: [GazKe.zip](#).