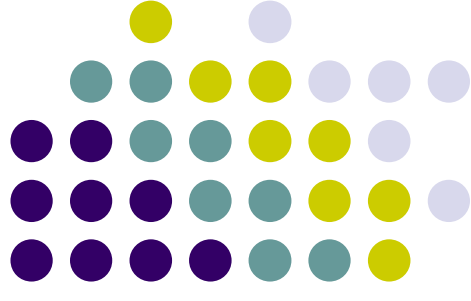
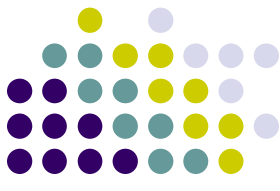


# Lineáris egyenletrendszerek

A fizika numerikus módszerei I.  
mf1n1a06- mf1n2a06  
Csabai István



# Gauss-elimináció



- Oldjuk meg a következő lineáris egyenletet:  $Ax = b$

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

...

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

- Nem változik a megoldás, ha:
  - Két sort felcserélünk
  - Két oszlopot felcserélünk (pontosabban az együttható mátrix oszlopait és az ismeretlen vektor megfelelő sorait)
  - Egy sor  $t$  helyettesítünk a sor és más sorok lineáris kombinációjával

# Gauss-elimináció



- Tegyük fel, hogy  $a_{11} \neq 0$
- Vonjuk ki minden  $i \geq 2$  sorból az első sor  $\frac{a_{i1}}{a_{11}}$  - szeresét:
- Azt kapjuk, hogy:

$$a'_{11}x_1 + a'_{12}x_2 + \dots + a'_{1n}x_n = b'_1$$

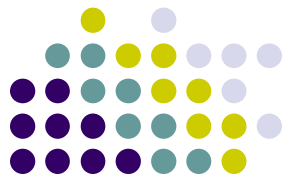
$$0 + a'_{22}x_2 + \dots + a'_{2n}x_n = b'_2$$

...

$$0 + a'_{n2}x_2 + \dots + a'_{nn}x_n = b'_n$$

- A fenti átalakítás nagyságrendileg  $N^2$  művelet végrehajtását igényelte

# Gauss-elimináció



- Vonjuk ki minden  $i \neq 2$  sorból a második sor  $\frac{a'_{i2}}{a'_{22}}$  - szeresét:
- Azt kapjuk, hogy:

$$a''_{11}x_1 + 0 + \dots + a''_{1n}x_n = b''_1$$

$$0 + a''_{22}x_2 + \dots + a''_{2n}x_n = b''_2$$

...

$$0 + 0 + \dots + a''_{nn}x_n = b''_n$$

# Gauss-elimináció



- Hasonlóan folytatva minden sorral, végül a következő mátrixhoz és jobb oldalhoz jutunk:

$$\begin{pmatrix} a_{11}^* & 0 & \dots & 0 & 0 & \dots & 0 & b_1^* \\ 0 & a_{22}^* & \dots & 0 & 0 & \dots & 0 & b_2^* \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_{rr}^* & 0 & \dots & 0 & b_r^* \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & a_{nn}^* & \dots & 0 & b_n^* \end{pmatrix}$$

- Végül minden sort leosztva a diagonális elemekkel, az átalakított jobboldal helyén megkapjuk az ismeretleneket.
- Tehát N-szer végeztünk el, lépésenként  $N^2$  műveletet, tehát nagyságrendileg  $N^3$  műveletet kellett összesen végrehajtani.

# Gauss-elimináció



- Összefoglalva az átalakításokat:

$$Ax = b \Rightarrow Ix = b^*$$

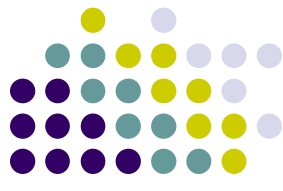
- Ha az ismeretlenek helyére egy  $X$  mátrixot, a jobb oldal helyett pedig eredetileg egy egységmátrixot írunk, akkor:

$$AX = I \Rightarrow IX = A^{-1}$$

- Vagyis a Gauss-elimináció segítségével a mátrix inverze is megkapható
- Mátrix-invertálás – lineáris egyenletrendszer megoldás műveletigénye (futási idő):

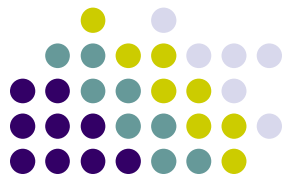
$$T \sim N^3$$

# Problémák



- Túlhátározott egyenletrendszer ( $M > N$ ), vagy szinguláris mátrix – nincs megoldás
  - Kereshetjük azt a megoldást amire az  $\|Ax - b\|$  minimális
- Alulhátározott mátrix ( $M < N$ ) – több megoldás
  - A megoldások alterében kereshetjük azt, amire  $\|x\|$  minimális
- Matematikailag egzaktul megoldható probléma a numerikus kerekítések miatt
  - Nincs megoldás
  - Pontatlan, hibás eredmény

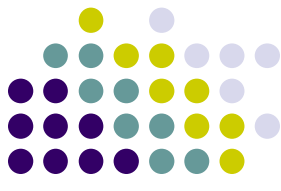
# Lineáris egyenletrendszerek



- Sok probléma megoldásánál fellép
- Nagyon sok numerikus feladat (paraméter-becslés, függvényillesztés, interpoláció, bizonyos differenciál egyenletek ...)  
közvetve erre vezet
- Általánosan  $N^3$  műveletet igényel a megoldás
- Speciális mátrixok (sáv-, Toeplitz-, ritka-), speciális módszerek ( $\sim N^3$ ,  $\sim N^2$ ,  $\sim N$  műveletigénnyel)
- A numerikus módszerek egyik fő kihívása: gyorsabb, pontosabb algoritmusok tervezése, a kerekítési hibák halmozódásának elkerülése
- Most nem vesszük részletesen: csak használjuk az Octave beépített módszerét



# Megoldás invertálással és a | operátorral



```
octave:##>A=[1, 1; 2, -3];  
octave:##> b=[3 5]';  
octave:##> inv(A)*b  
ans =  
 2.8000  
 0.2000
```

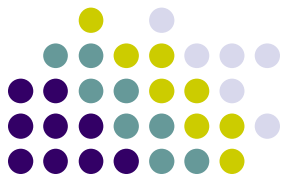
- **Ellenőrzés:**

```
octave:##> A*ans  
ans =  
 3.0000  
 5.0000
```

```
octave:##>A\b  
ans =  
 2.8000 0.2000
```

**Gyorsabb !!**

# Szinguláris mátrix



$$\begin{aligned}u + v + w &= 2 \\ 2u + 3w &= 5 \\ 3u + v + 4w &= 6\end{aligned}\quad (\text{a bal oldal az első két sor összege, a jobb oldal nem})$$

```
octave: #>A=[1 1 1
            2 0 3
            3 1 4];
octave: #>b=[ 2 5 6]';
octave: #>x=A\b;
warning: matrix singular to machine precision, rcond = 1.15648e-17
warning: attempting to find minimum norm solution
x =
  0.69048
 -0.11905
  1.09524
octave: #>A*x
ans =
  1.6667
  4.6667
  6.3333
```

# Alulhatározott mátrix

$$\begin{aligned}u + v + w &= 2 \\ 2u + 3w &= 5\end{aligned}$$

```
octave:##> A=[1,1,1;2,0,3];
```

```
octave:##> b=[2,5]';
```

```
octave:##> x=A\b
```

```
ans =
```

```
0.785714
```

```
0.071429
```

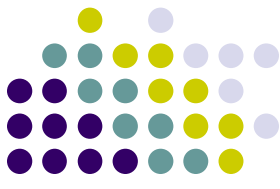
```
1.142857
```

```
octave:##> A*x
```

```
ans =
```

```
2.0000
```

```
5.0000
```



# Rosszul kondicionált mátrix



$$\begin{bmatrix} 1 & 1 \\ 1 & 1.01 \end{bmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2.01 \end{pmatrix}$$

```
octave:##>M=[1 1; 1 1.01]; b=[2; 2.01];
```

```
octave:##>x=M\b
```

```
x =
```

```
1.000000
```

```
1.000000
```

- **Kis perturbáció**

```
octave:##>M(1,2)=1.005;
```

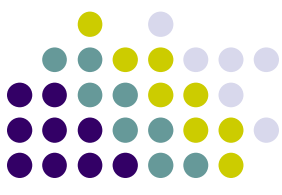
```
octave:##>x=M\b
```

```
x =
```

```
-0.01000000
```

```
2.00000000
```

- **Nagyobb mátrixoknál könnyebben előfordul!**



# Rosszul kondicionált mátrix

- Hilbert-féle mátrix
- Pl.:

$$H_{ij} = \frac{1}{i+j-1}.$$

$$H = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 & 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \\ 1/2 & 1/3 & 1/4 & 1/5 & 1/6 & 1/7 & 1/8 & 1/9 & 1/10 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 & 1/8 & 1/9 & 1/10 & 1/11 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 & 1/9 & 1/10 & 1/11 & 1/12 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 & 1/10 & 1/11 & 1/12 & 1/13 \\ 1/6 & 1/7 & 1/8 & 1/9 & 1/10 & 1/11 & 1/12 & 1/13 & 1/14 \\ 1/7 & 1/8 & 1/9 & 1/10 & 1/11 & 1/12 & 1/13 & 1/14 & 1/15 \\ 1/8 & 1/9 & 1/10 & 1/11 & 1/12 & 1/13 & 1/14 & 1/15 & 1/16 \\ 1/9 & 1/10 & 1/11 & 1/12 & 1/13 & 1/14 & 1/15 & 1/16 & 1/17 \\ 1/10 & 1/11 & 1/12 & 1/13 & 1/14 & 1/15 & 1/16 & 1/17 & 1/18 \\ 1/11 & 1/12 & 1/13 & 1/14 & 1/15 & 1/16 & 1/17 & 1/18 & 1/19 \\ 1/12 & 1/13 & 1/14 & 1/15 & 1/16 & 1/17 & 1/18 & 1/19 & 1/20 \\ 1/13 & 1/14 & 1/15 & 1/16 & 1/17 & 1/18 & 1/19 & 1/20 & 1/21 \\ 1/14 & 1/15 & 1/16 & 1/17 & 1/18 & 1/19 & 1/20 & 1/21 & 1/22 \\ 1/15 & 1/16 & 1/17 & 1/18 & 1/19 & 1/20 & 1/21 & 1/22 & 1/23 \\ 1/16 & 1/17 & 1/18 & 1/19 & 1/20 & 1/21 & 1/22 & 1/23 & 1/24 \\ 1/17 & 1/18 & 1/19 & 1/20 & 1/21 & 1/22 & 1/23 & 1/24 & 1/25 \\ 1/18 & 1/19 & 1/20 & 1/21 & 1/22 & 1/23 & 1/24 & 1/25 & 1/26 \\ 1/19 & 1/20 & 1/21 & 1/22 & 1/23 & 1/24 & 1/25 & 1/26 & 1/27 \\ 1/20 & 1/21 & 1/22 & 1/23 & 1/24 & 1/25 & 1/26 & 1/27 & 1/28 \\ 1/21 & 1/22 & 1/23 & 1/24 & 1/25 & 1/26 & 1/27 & 1/28 & 1/29 \\ 1/22 & 1/23 & 1/24 & 1/25 & 1/26 & 1/27 & 1/28 & 1/29 & 1/30 \end{bmatrix}.$$

```
octave: #>A=hilb(20);
```

```
octave-3.0.0.exe:36> x=linspace(1,20,20)';
```

```
octave-3.0.0.exe:37> b=A*x;
```

```
octave-3.0.0.exe:38> A\b
```

```
warning: matrix singular to machine precision, rcond =  
1.264e-020
```

```
warning: attempting to find minimum norm solution
```

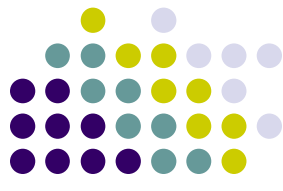
```
warning: dgeqsd: rank deficient 20x20 matrix, rank = 14
```

```
ans =
```

```
1.00000,2.00001,2.99976,4.00369,4.97135,6.12263,6.71047,  
8.32126,8.99804,9.73197,10.97731,12.24056,13.13674,13.83668,  
14.77220,16.00726,17.25326,18.12140,18.67825,20.11717
```

# Futási idő mérése Octave-ban

---



- `tíc` : stopper indítása
- `toc` : stopper leállítása

```
octave:##> tíc; x=A\b; toc  
ans = 0.00046100
```

- Feladat: lineáris egyenlet megoldás sebességének skálázása  $N$  függvényében

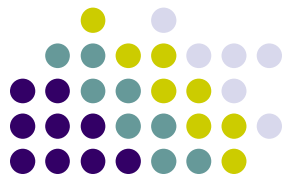
# `inv(A)` és `A\b` összehasonlítása

---



```
octave##> n=3000;
octave##> A=toeplitz(n:-1:1);
octave##> b=linspace(1,2,n)';
octave##> tic; inv(A)*b; toc
Elapsed time is 65 seconds.
octave##> tic; A\b; toc
Elapsed time is 23 seconds.
```

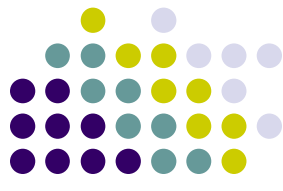
# Adatok beolvasása/kiírása



```
octave:##> save myMatrix A
bash-3.00$ cat myMatrix
# Created by Octave 2.1.57, Tue Mar 06
11:10:01 2007 CET
<csabai@evghumgmt.colbud>
# name: A
# type: matrix
# rows: 2
# columns: 2
1 2
5 4
```

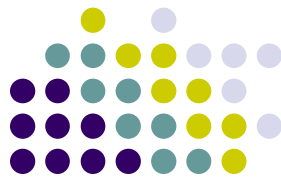


# Adatok beolvasása/kiírása



```
octave:14> clear all
octave:15> A
error: `A' undefined near line 15 column
1
octave:15> load myMatrix
octave:16> A
A =
    1    2
    5    4
```

# Adatok beolvasása/kiírása



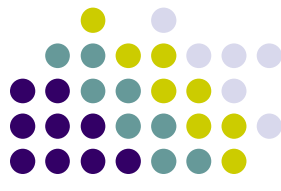
```
bash-3.00$ cat matrix2.dat
3 5 7
1 2 3
2 7 1
```

```
octave:17> B = load matrix2.dat
B =
   3   5   7
   1   2   3
   2   7   1
```

- További formátumok: Octave Manual 16. fejezet

# Olvasnivaló

---



- Stoyan G. MATLAB könyv, 54-67.o., 72-86.o.
- Stoyan G. Numerikus matematika, 41-46. o.
- P.J.G. Long: Octave Tutorial, 38-44. o.