

# A számítástudomány alapjai

Lineáris egyenletrendszerek

2022. október 18.

**Def:** Lineáris egyenlet: Ismeretlenek konstansszorosainak összege konstans.

**Def:** **Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Def:** **Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.  
**Megoldás:** Olyan értékadás, ami minden egyenletet igazgá tesz.

**Def: Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazgá tesz.

**Példa:**

$$3x - 4z = 666$$

$$33x - y + 77z = 42$$

$$\sqrt[3]{2}\sqrt{5}y - (\ln(\cos 42)) \cdot z = \pi e^\pi$$

**Def:** **Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.  
**Megoldás:** Olyan értékadás, ami minden egyenletet igazgá tesz.

**Def:** **Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.  
**Megoldás:** Olyan értékadás, ami minden egyenletet igazgá tesz.  
**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Def:** **Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazgá tesz.

**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Naiv módszer:** Minden fázisban egyik egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe helyettesítjük. A kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.



**Def:** **Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazá tesz.

**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Naiv módszer:** Minden fázisban egyik egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe helyettesítjük. A kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

**A megoldás formája:** Az eljárás végén kapott egyenletek nem tartalmazznak ismeretlent. Minden ilyen egyenlet vagy azonosság, vagy ellentmondás.

**Def: Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazá tesz.

**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Naiv módszer:** Minden fázisban egyik egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe helyettesítjük. A kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

**A megoldás formája:** Az eljárás végén kapott egyenletek nem tartalmazznak ismeretlent. Minden ilyen egyenlet vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás.

**Def: Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazzá tesz.

**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Naiv módszer:** Minden fázisban egyik egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe helyettesítjük. A kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

**A megoldás formája:** Az eljárás végén kapott egyenletek nem tartalmazznak ismeretlent. Minden ilyen egyenlet vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek tetszőleges értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

**Def:** **Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazgá tesz.

**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Naiv módszer:** Minden fázisban egyik egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe helyettesítjük. A kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

**A megoldás formája:** Az eljárás végén kapott egyenletek nem tartalmazznak ismeretlent. Minden ilyen egyenlet vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek tetszőleges értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

**Példa:**

$$\begin{array}{l} x - 2y = 7 \quad \boxed{x} = 7 + 2y \\ 3x + y = 11 \quad 21 + 6y + y = 11 \\ 2x + 3y = 5 \quad 14 + 4y + 3y = 5 \end{array} \quad \begin{array}{l} \boxed{x} = 7 + 2y \\ 7y = -10 \\ 7y = -9 \end{array} \quad \begin{array}{l} \boxed{x} = 7 + 2y \\ \boxed{y} = -\frac{10}{7} \\ 7 \cdot \frac{-10}{7} = -9 \quad \neq \end{array}$$

**Def: Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazgá tesz.

**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Naiv módszer:** Minden fázisban egyik egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe helyettesítjük. A kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

**A megoldás formája:** Az eljárás végén kapott egyenletek nem tartalmazznak ismeretlent. Minden ilyen egyenlet vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek tetszőleges értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

**Példa:**

**Def: Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazgá tesz.

**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Naiv módszer:** Minden fázisban egyik egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe helyettesítjük. A kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

**A megoldás formája:** Az eljárás végén kapott egyenletek nem tartalmazznak ismeretlent. Minden ilyen egyenlet vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek tetszőleges értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

**Példa:**

$$\begin{array}{l} x - 2y = 7 \quad \boxed{x} = 7 + 2y \quad \boxed{x} = 7 + 2y \quad \boxed{x} = 7 + 2y \\ 3x + y = 11 \quad 21 + 6y + y = 11 \quad 7y = -10 \quad \boxed{y} = -\frac{10}{7} \quad x = \frac{29}{7} \\ 2x + 3y = 4 \quad 14 + 4y + 3y = 4 \quad 7y = -10 \quad 7 \cdot \frac{-10}{7} = -10 \checkmark \quad y = -\frac{10}{7} \end{array}$$

**Def: Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazgá tesz.

**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Naiv módszer:** Minden fázisban egyik egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe helyettesítjük. A kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

**A megoldás formája:** Az eljárás végén kapott egyenletek nem tartalmazznak ismeretlent. Minden ilyen egyenlet vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek tetszőleges értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

**Példa:**

**Def:** **Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazá tesz.

**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Naiv módszer:** Minden fázisban egyik egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe helyettesítjük. A kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

**A megoldás formája:** Az eljárás végén kapott egyenletek nem tartalmazznak ismeretlent. Minden ilyen egyenlet vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek tetszőleges értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

**Példa:**

$$\begin{array}{llll} x - 2y = 7 & \boxed{x} = 7 + 2y & \boxed{x} = 7 + 2y & \\ 3x - 6y = 21 & 21 + 6y - 6y = 21 & 21 = 21 \quad \checkmark & x = 7 + 2y \\ -2x + 4y = -14 & -14 - 4y + 4y = -14 & -14 = -14 \quad \checkmark & y \in \mathbb{R} \end{array}$$



**Def: Lineáris egyenlet:** Ismeretlenek konstansszorosainak összege konstans. **Lineáris egyenletrendszer:** Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazá tesz.

**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Naiv módszer:** Minden fázisban egyik egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe helyettesítjük. A kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

**A megoldás formája:** Az eljárás végén kapott egyenletek nem tartalmazznak ismeretlent. Minden ilyen egyenlet vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek tetszőleges értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

**Def:** Lineáris egyenlet: Ismeretlenek konstansszorosainak összege konstans. Lineáris egyenletrendszer: Véges sok lineáris egyenlet.

**Megoldás:** Olyan értékadás, ami minden egyenletet igazgá tesz.

**Cél:** Lineáris egyenletrendszer összes megoldásának megtalálása.

**Naiv módszer:** Minden fázisban egyik egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe helyettesítjük. A kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

**A megoldás formája:** Az eljárás végén kapott egyenletek nem tartalmazznak ismeretlent. Minden ilyen egyenlet vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek tetszőleges értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

A továbbiakban a megoldás egy áttekinthetőbb módszerét és az ahhoz kapcsolódó terminológiát mutatjuk be.

# Elemi sorekvivalens átalakítások

**Def:** Lineáris egyenletrendszer kibővített együtthatómátrixa: a sorok az egyenleteknek, az oszlopok az ismeretleneknek ill. az egyenletek jobb oldalainak felelnek meg, az egyes mezőkben pedig a megfelelő együttható ill. jobb oldali konstans áll.

**Példa:**

$$\begin{array}{r} x_1 - 3x_3 + 5x_4 = -6 \\ 7x_1 + 2x_2 + 3x_3 = 9 \\ x_2 + 7x_3 - 2x_4 = 11 \end{array} \quad \mapsto \quad \begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 0 & -3 & 5 & -6 \\ 7 & 2 & 3 & 0 & 9 \\ 0 & 1 & 7 & -2 & 11 \end{array}$$

# Elemi sorekvivalens átalakítások

**Def:** Lineáris egyenletrendszer kibővített együtthatómátrixa: a sorok az egyenleteknek, az oszlopok az ismeretleneknek ill. az egyenletek jobb oldalainak felelnek meg, az egyes mezőkben pedig a megfelelő együttható ill. jobb oldali konstans áll.

**Példa:**

$$\begin{array}{r} x_1 - 3x_3 + 5x_4 = -6 \\ 7x_1 + 2x_2 + 3x_3 = 9 \\ x_2 + 7x_3 - 2x_4 = 11 \end{array} \quad \mapsto \quad \begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 0 & -3 & 5 & -6 \\ 7 & 2 & 3 & 0 & 9 \\ 0 & 1 & 7 & -2 & 11 \end{array}$$

**Megj:** A kibővített együtthatómátrix a lineáris egyenletrendszer felírásának egy tömör módja: elkerüljük vele a műveleti- és egyenlőségjelekkel piszmogást, mégis teljesen áttekinthető módon tartalmaz minden lényeges információt.

## Elemi sorekvivalens átalakítások

**Def:** **Lineáris egyenletrendszer kibővített együtthatómátrixa:** a sorok az egyenleteknek, az oszlopok az ismeretleneknek ill. az egyenletek jobb oldalainak felelnek meg, az egyes mezőkben pedig a megfelelő együttható ill. jobb oldali konstans áll.

**Példa:**

$$\begin{array}{l} x_1 - 3x_3 + 5x_4 = -6 \\ 7x_1 + 2x_2 + 3x_3 = 9 \\ x_2 + 7x_3 - 2x_4 = 11 \end{array} \quad \mapsto \quad \begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 0 & -3 & 5 & -6 \\ 7 & 2 & 3 & 0 & 9 \\ 0 & 1 & 7 & -2 & 11 \end{array}$$

**Megj:** A kibővített együtthatómátrix a lineáris egyenletrendszer felírásának egy tömör módja: elkerüljük vele a műveleti- és egyenlőségjelekkel piszmogást, mégis teljesen áttekinthető módon tartalmaz minden lényeges információt.

**Megoldás módszere:** Ekvivalens átalakításokat végzünk. Ezek során a megoldások halmaza nem változik. Konkrétan: egyenleteket felcserélünk, egyenletet nemnullával végigszorozunk ill. az  $i$ -dik egyenletet kicseréljük az  $i$ -dik és  $j$ -dik egyenletek összegére. Erről szól a következő definíció.

# Elemi sorokvivalens átalakítások

**Def:** **Lineáris egyenletrendszer kibővített együtthatómátrixa:** a sorok az egyenleteknek, az oszlopok az ismeretleneknek ill. az egyenletek jobb oldalainak felelnek meg, az egyes mezőkben pedig a megfelelő együttható ill. jobb oldali konstans áll.

## Elemi sorekvivalens átalakítások

**Def:** Lineáris egyenletrendszer kibővített együtthatómátrixa: a sorok az egyenleteknek, az oszlopok az ismeretleneknek ill. az egyenletek jobb oldalainak felelnek meg, az egyes mezőkben pedig a megfelelő együttható ill. jobb oldali konstans áll.

**Def:** A kibővített egyhómx elemi sorekvivalens átalakítása (ESÁ): (1) sorcsere, (2) sor nemnulla konstanssal végigszorzása, (3) az  $i$ -dik sor helyettesítése az  $i$ -dik és  $j$ -dik sorok (koordinátánkénti) összegével (csupa0 sor elhagyása, az  $i$ -dik sor helyettesítése az  $i$ -dik sor és a  $j$ -dik sor konstansszorosának összegével).

## Elemi sorekvivalens átalakítások

**Def:** Lineáris egyenletrendszer kibővített együtthatómátrixa: a sorok az egyenleteknek, az oszlopok az ismeretleneknek ill. az egyenletek jobb oldalainak felelnek meg, az egyes mezőkben pedig a megfelelő együttható ill. jobb oldali konstans áll.

**Def:** A kibővített egyhómx elemi sorekvivalens átalakítása (ESÁ): (1) sorcsere, (2) sor nemnulla konstanssal végigszorzása, (3) az  $i$ -dik sor helyettesítése az  $i$ -dik és  $j$ -dik sorok (koordinátánkénti) összegével (csupa0 sor elhagyása, az  $i$ -dik sor helyettesítése az  $i$ -dik sor és a  $j$ -dik sor konstansszorosának összegével).

**Állítás:** ESÁ hatására a megoldások halmaza nem változik.



## Elemi sorekvivalens átalakítások

**Def:** Lineáris egyenletrendszer kibővített együtthatómátrixa: a sorok az egyenleteknek, az oszlopok az ismeretleneknek ill. az egyenletek jobb oldalainak felelnek meg, az egyes mezőkben pedig a megfelelő együttható ill. jobb oldali konstans áll.

**Def:** A kibővített egyhómx elemi sorekvivalens átalakítása (ESÁ): (1) sorcsere, (2) sor nemnulla konstanssal végigszorzása, (3) az  $i$ -dik sor helyettesítése az  $i$ -dik és  $j$ -dik sorok (koordinátánkénti) összegével (csupa0 sor elhagyása, az  $i$ -dik sor helyettesítése az  $i$ -dik sor és a  $j$ -dik sor konstansszorosának összegével).

**Állítás:** ESÁ hatására a megoldások halmaza nem változik.

**Biz:** Minden ESÁ előtti megoldás megoldás marad az ESÁ után is. Minden ESÁ fordítottja megkapható ESÁ-ok egymásutánjaként is. Ezért minden ESÁ utáni megoldás megoldja az ESÁ előtti előtti rendszert is. □

## Elemi sorekvivalens átalakítások

**Def:** Lineáris egyenletrendszer kibővített együtthatómátrixa: a sorok az egyenleteknek, az oszlopok az ismeretleneknek ill. az egyenletek jobb oldalainak felelnek meg, az egyes mezőkben pedig a megfelelő együttható ill. jobb oldali konstans áll.

**Def:** A kibővített egyhómx elemi sorekvivalens átalakítása (ESÁ): (1) sorcsere, (2) sor nemnulla konstanssal végigszorzása, (3) az  $i$ -dik sor helyettesítése az  $i$ -dik és  $j$ -dik sorok (koordinátánkénti) összegével (csupa0 sor elhagyása, az  $i$ -dik sor helyettesítése az  $i$ -dik sor és a  $j$ -dik sor konstansszorosának összegével).

**Állítás:** ESÁ hatására a megoldások halmaza nem változik.

## Elemi sorekvivalens átalakítások

**Def:** Lineáris egyenletrendszer kibővített együtthatómátrixa: a sorok az egyenleteknek, az oszlopok az ismeretleneknek ill. az egyenletek jobb oldalainak felelnek meg, az egyes mezőkben pedig a megfelelő együttható ill. jobb oldali konstans áll.

**Def:** A kibővített egyhómx elemi sorekvivalens átalakítása (ESÁ): (1) sorcsere, (2) sor nemnulla konstanssal végigszorozása, (3) az  $i$ -dik sor helyettesítése az  $i$ -dik és  $j$ -dik sorok (koordinátánkénti) összegével (csupa0 sor elhagyása, az  $i$ -dik sor helyettesítése az  $i$ -dik sor és a  $j$ -dik sor konstansszorosának összegével).

**Állítás:** ESÁ hatására a megoldások halmaza nem változik.

**Cél:** A lineáris egyenletrendszerből kiindulva, ESÁ-ok alkalmazásával elérni, hogy a kapott rendszer a megoldások leírását adja.

Először definiáljuk a megoldást leíró kibővített egyhómx-ot.

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

- (1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)
- (2) minden  $v_1$  feletti sorban van ettől a  $v_1$ -től balra eső másik  $v_1$ .

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

(1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy  **$v_1$** )

(2) minden  $v_1$  feletti sorban van ettől a  $v_1$ -től balra eső másik  $v_1$ .

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

(3)  $M$  LA és (2)  $M$ -ben minden  $v_1$  felett csak nullák állnak.

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

- (1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)
- (2) minden v1 feletti sorban van ettől a v1-től balra eső másik v1.

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

- (3)  $M$  LA és (2)  $M$ -ben minden v1 felett csak nullák állnak.

**Példa:** LA mátrix

$$\begin{pmatrix} 1 & 3 & -2 & 0 & 1 & 1 & 7 \\ 0 & 0 & 1 & 2 & 0 & 5 & -2 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

ill.

RLA mátrix

$$\begin{pmatrix} 0 & 1 & 0 & -2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

(1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy  **$v_1$** )

(2) minden  $v_1$  feletti sorban van ettől a  $v_1$ -től balra eső másik  $v_1$ .

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

(3)  $M$  LA és (2)  $M$ -ben minden  $v_1$  felett csak nullák állnak.

**Példa:**

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

(1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)

(2) minden  $v_1$  feletti sorban van ettől a  $v_1$ -től balra eső másik  $v_1$ .

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

(3)  $M$  LA és (2)  $M$ -ben minden  $v_1$  felett csak nullák állnak.

**Példa:** RLA kibővített egyhómx:

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array}$$



## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

(1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)

(2) minden  $v1$  feletti sorban van ettől a  $v1$ -től balra eső másik  $v1$ .

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

(3)  $M$  LA és (2)  $M$ -ben minden  $v1$  felett csak nullák állnak.

**Példa:** RLA kibővített egyhómx:

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array}$$

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

(1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)

(2) minden  $v_1$  feletti sorban van ettől a  $v_1$ -től balra eső másik  $v_1$ .

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

(3)  $M$  LA és (2)  $M$ -ben minden  $v_1$  felett csak nullák állnak.

**Példa:** RLA kibővített egyhómx:

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2, x_4 \in \mathbb{R} \\ x_1 = 1 - 3x_2 + 2x_4 \\ x_3 = 7 - 5x_4 \end{array}$$

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

- (1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)
- (2) minden v1 feletti sorban van ettől a v1-től balra eső másik v1.

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

- (3)  $M$  LA és (2)  $M$ -ben minden v1 felett csak nullák állnak.

**Példa:** RLA kibővített egyhómx:

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2, x_4 \in \mathbb{R} \\ x_1 = 1 - 3x_2 + 2x_4 \\ x_3 = 7 - 5x_4 \end{array}$$

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array}$$

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

- (1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)
- (2) minden v1 feletti sorban van ettől a v1-től balra eső másik v1.

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

- (3)  $M$  LA és (2)  $M$ -ben minden v1 felett csak nullák állnak.

**Példa:** RLA kibővített egyhómx:

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2, x_4 \in \mathbb{R} \\ x_1 = 1 - 3x_2 + 2x_4 \\ x_3 = 7 - 5x_4 \end{array}$$

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \\ 0 = 1 \\ 0 = 0 \end{array}$$

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

- (1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)
- (2) minden v1 feletti sorban van ettől a v1-től balra eső másik v1.

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

- (3)  $M$  LA és (2)  $M$ -ben minden v1 felett csak nullák állnak.

**Példa:** RLA kibővített egyhómx:

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2, x_4 \in \mathbb{R} \\ x_1 = 1 - 3x_2 + 2x_4 \\ x_3 = 7 - 5x_4 \end{array}$$
$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \\ 0 = 1 \\ 0 = 0 \end{array} \mapsto \text{⚡}$$

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

- (1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)
- (2) minden v1 feletti sorban van ettől a v1-től balra eső másik v1.

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

- (3)  $M$  LA és (2)  $M$ -ben minden v1 felett csak nullák állnak.

**Példa:** RLA kibővített egyhómx:

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2, x_4 \in \mathbb{R} \\ x_1 = 1 - 3x_2 + 2x_4 \\ x_3 = 7 - 5x_4 \end{array}$$

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \\ 0 = 1 \\ 0 = 0 \end{array} \mapsto \text{⚡}$$

**Def:** Kib. egyhómx **tilos sora**:  $0 \dots 0 \mid x$  alakú sor, ha  $x \neq 0$ .

**Def:** A RLA kib. egyhómx v1-hez tartozó változója **kötött**, a többi változó (amihez nem tartozik v1) **szabad** (vagy **szabad paraméter**).

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

(1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)

(2) minden v1 feletti sorban van ettől a v1-től balra eső másik v1.

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

(3)  $M$  LA és (2)  $M$ -ben minden v1 felett csak nullák állnak.

**Példa:** RLA kibővített egyhómx:

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2, x_4 \in \mathbb{R} \\ x_1 = 1 - 3x_2 + 2x_4 \\ x_3 = 7 - 5x_4 \end{array}$$

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \\ 0 = 1 \\ 0 = 0 \end{array} \mapsto \text{⚡}$$

**Def:** Kib. egyhómx **tilos sora**:  $0 \dots 0 \mid x$  alakú sor, ha  $x \neq 0$ .

**Def:** A RLA kib. egyhómx v1-hez tartozó változója **kötött**, a többi változó (amihez nem tartozik v1) **szabad** (vagy **szabad paraméter**).

**Megf:** Ha a kib. egyhómx RLA, akkor (1) minden sor vagy a v1-hez tartozó változó értékadása, vagy tilos sor, vagy csupa0 sor.

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

(1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)

(2) minden v1 feletti sorban van ettől a v1-től balra eső másik v1.

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

(3)  $M$  LA és (2)  $M$ -ben minden v1 felett csak nullák állnak.

**Példa:** RLA kibővített egyhómx:

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2, x_4 \in \mathbb{R} \\ x_1 = 1 - 3x_2 + 2x_4 \\ x_3 = 7 - 5x_4 \end{array}$$

**Def:** Kib. egyhómx **tilos sora**:  $0 \dots 0 \mid x$  alakú sor, ha  $x \neq 0$ .

**Def:** A RLA kib. egyhómx v1-hez tartozó változója **kötött**, a többi változó (amihez nem tartozik v1) **szabad** (vagy **szabad paraméter**).

**Megf:** Ha a kib. egyhómx RLA, akkor (1) minden sor vagy a v1-hez tartozó változó értékadása, vagy tilos sor, vagy csupa0 sor.



## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

(1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)

(2) minden v1 feletti sorban van ettől a v1-től balra eső másik v1.

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

(3)  $M$  LA és (2)  $M$ -ben minden v1 felett csak nullák állnak.

**Példa:** RLA kibővített egyhómx:

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2, x_4 \in \mathbb{R} \\ x_1 = 1 - 3x_2 + 2x_4 \\ x_3 = 7 - 5x_4 \end{array}$$

**Def:** Kib. egyhómx **tilos sora**:  $0 \dots 0 \mid x$  alakú sor, ha  $x \neq 0$ .

**Def:** A RLA kib. egyhómx v1-hez tartozó változója **kötött**, a többi változó (amihez nem tartozik v1) **szabad** (vagy **szabad paraméter**).

**Megf:** Ha a kib. egyhómx RLA, akkor (1) minden sor vagy a v1-hez tartozó változó értékadása, vagy tilos sor, vagy csupa0 sor.

(2) Ha van tilos sor, akkor nincs megoldás.

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

- (1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)
- (2) minden v1 feletti sorban van ettől a v1-től balra eső másik v1.

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

- (3)  $M$  LA és (2)  $M$ -ben minden v1 felett csak nullák állnak.

**Példa:** RLA kibővített egyhómx:

$$\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2, x_4 \in \mathbb{R} \\ x_1 = 1 - 3x_2 + 2x_4 \\ x_3 = 7 - 5x_4 \end{array}$$

**Def:** Kib. egyhómx **tilos sora**:  $0 \dots 0 \mid x$  alakú sor, ha  $x \neq 0$ .

**Def:** A RLA kib. egyhómx v1-hez tartozó változója **kötött**, a többi változó (amihez nem tartozik v1) **szabad** (vagy **szabad paraméter**).

**Megf:** Ha a kib. egyhómx RLA, akkor (1) minden sor vagy a v1-hez tartozó változó értékadása, vagy tilos sor, vagy csupa0 sor.

(2) Ha van tilos sor, akkor nincs megoldás.

(3) Ha nincs tilos sor, a szabad paraméterek tetsz. értékadásához egyértelmű megoldás tartozik.

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

- (1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)
- (2) minden  $v_1$  feletti sorban van ettől a  $v_1$ -től balra eső másik  $v_1$ .

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

- (3)  $M$  LA és (2)  $M$ -ben minden  $v_1$  felett csak nullák állnak.

**Def:** Kib. egyhómx **tilos sora**:  $0 \dots 0 \mid x$  alakú sor, ha  $x \neq 0$ .

**Def:** A RLA kib. egyhómx  $v_1$ -hez tartozó változója **kötött**, a többi változó (amihez nem tartozik  $v_1$ ) **szabad** (vagy **szabad paraméter**).

**Megf:** Ha a kib. egyhómx RLA, akkor (1) minden sor vagy a  $v_1$ -hez tartozó változó értékadása, vagy tilos sor, vagy csupa0 sor.

(2) Ha van tilos sor, akkor nincs megoldás.

(3) Ha nincs tilos sor, a szabad paraméterek tetsz. értékadásához egyértelmű megoldás tartozik.

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

- (1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)
- (2) minden  $v_1$  feletti sorban van ettől a  $v_1$ -től balra eső másik  $v_1$ .

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

- (3)  $M$  LA és (2)  $M$ -ben minden  $v_1$  felett csak nullák állnak.

**Def:** Kib. egyhómx **tilos sora**:  $0 \dots 0 \mid x$  alakú sor, ha  $x \neq 0$ .

**Def:** A RLA kib. egyhómx  $v_1$ -hez tartozó változója **kötött**, a többi változó (amihez nem tartozik  $v_1$ ) **szabad** (vagy **szabad paraméter**).

**Megf:** Ha a kib. egyhómx RLA, akkor (1) minden sor vagy a  $v_1$ -hez tartozó változó értékadása, vagy tilos sor, vagy csupa0 sor.

(2) Ha van tilos sor, akkor nincs megoldás.

(3) Ha nincs tilos sor, a szabad paraméterek tetsz. értékadásához egyértelmű megoldás tartozik.

**Megf:** A lin. egyenletsz. megoldása tekinthető úgy, hogy a lin. egyenletsz. egy RLA kibővített egyhómx-szal van megadva.

## (Redukált) lépcsős alak

**Def:** Az  $M$  mátrix **lépcsős alakú (LA)**, ha

- (1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **v1**)
- (2) minden v1 feletti sorban van ettől a v1-től balra eső másik v1.

Az  $M$  mátrix **redukált lépcsős alakú (RLA)**, ha

- (3)  $M$  LA és (2)  $M$ -ben minden v1 felett csak nullák állnak.

**Def:** Kib. egyhómx **tilos sora**:  $0 \dots 0 \mid x$  alakú sor, ha  $x \neq 0$ .

**Def:** A RLA kib. egyhómx v1-hez tartozó változója **kötött**, a többi változó (amihez nem tartozik v1) **szabad** (vagy **szabad paraméter**).

**Megf:** Ha a kib. egyhómx RLA, akkor (1) minden sor vagy a v1-hez tartozó változó értékadása, vagy tilos sor, vagy csupa0 sor.

(2) Ha van tilos sor, akkor nincs megoldás.

(3) Ha nincs tilos sor, a szabad paraméterek tetsz. értékadásához egyértelmű megoldás tartozik.

**Megf:** A lin. egyenletrsz. megoldása tekinthető úgy, hogy a lin. egyenletrsz. egy RLA kibővített egyhómx-szal van megadva.

**Cél:** Olyan eljárás, ami ESÁ-okkal tetsz. mátrixot RLA-vá alakít.

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet v1-sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott v1 alatti elemeket.

**Példa:**

0	0	-1	3	5
3	6	-6	0	9
2	4	0	1	-1

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet v1-sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott v1 alatti elemeket.

**Példa:**

$$\begin{array}{ccccccccc} 0 & 0 & -1 & 3 & 5 & & 3 & 6 & -6 & 0 & 9 \\ 3 & 6 & -6 & 0 & 9 & \rightarrow & 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 & & 2 & 4 & 0 & 1 & -1 \end{array}$$



# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Példa:**

$$\begin{array}{cccccc} 0 & 0 & -1 & 3 & 5 & & 3 & 6 & -6 & 0 & 9 & & 1 & 2 & -2 & 0 & 3 \\ 3 & 6 & -6 & 0 & 9 & \rightarrow & 0 & 0 & -1 & 3 & 5 & \rightarrow & 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 & & 2 & 4 & 0 & 1 & -1 & & 2 & 4 & 0 & 1 & -1 \end{array}$$

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Példa:**

$$\begin{array}{cccccc} 0 & 0 & -1 & 3 & 5 & & 3 & 6 & -6 & 0 & 9 & & 1 & 2 & -2 & 0 & 3 \\ 3 & 6 & -6 & 0 & 9 & \rightarrow & 0 & 0 & -1 & 3 & 5 & \rightarrow & 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 & & 2 & 4 & 0 & 1 & -1 & & 2 & 4 & 0 & 1 & -1 \\ \rightarrow & 1 & 2 & -2 & 0 & 3 & & & & & & & & & & & \\ \rightarrow & 0 & 0 & -1 & 3 & 5 & & & & & & & & & & & \\ & 0 & 0 & 4 & 1 & -7 & & & & & & & & & & & \end{array}$$

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Példa:**

$$\begin{array}{cccccc} 0 & 0 & -1 & 3 & 5 & & 3 & 6 & -6 & 0 & 9 & & 1 & 2 & -2 & 0 & 3 \\ 3 & 6 & -6 & 0 & 9 & \rightarrow & 0 & 0 & -1 & 3 & 5 & \rightarrow & 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 & & 2 & 4 & 0 & 1 & -1 & & 2 & 4 & 0 & 1 & -1 \\ \rightarrow & 1 & 2 & -2 & 0 & 3 & & 1 & 2 & -2 & 0 & 3 & & & & & & \\ & 0 & 0 & -1 & 3 & 5 & \rightarrow & 0 & 0 & 1 & -3 & -5 & & & & & & \\ & 0 & 0 & 4 & 1 & -7 & & 0 & 0 & 4 & 1 & -7 & & & & & & \end{array}$$

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet v1-sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott v1 alatti elemeket.

**Példa:**

$$\begin{array}{cccccc} 0 & 0 & -1 & 3 & 5 & 3 & 6 & -6 & 0 & 9 & 1 & 2 & -2 & 0 & 3 \\ 3 & 6 & -6 & 0 & 9 & \rightarrow & 0 & 0 & -1 & 3 & 5 & \rightarrow & 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 & & 2 & 4 & 0 & 1 & -1 & & 2 & 4 & 0 & 1 & -1 \\ \rightarrow & 1 & 2 & -2 & 0 & 3 & & 1 & 2 & -2 & 0 & 3 & & & & & \\ \rightarrow & 0 & 0 & -1 & 3 & 5 & \rightarrow & 0 & 0 & 1 & -3 & -5 & & & & & \\ & 0 & 0 & 4 & 1 & -7 & & 0 & 0 & 4 & 1 & -7 & & & & & \\ \rightarrow & 1 & 2 & -2 & 0 & 3 & & & & & & & & & & & \\ \rightarrow & 0 & 0 & 1 & -3 & -5 & & & & & & & & & & & \\ & 0 & 0 & 0 & 13 & 13 & & & & & & & & & & & \end{array}$$

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Példa:**

$$\begin{array}{cccccc} 0 & 0 & -1 & 3 & 5 & 3 & 6 & -6 & 0 & 9 & 1 & 2 & -2 & 0 & 3 \\ 3 & 6 & -6 & 0 & 9 & \rightarrow & 0 & 0 & -1 & 3 & 5 & \rightarrow & 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 & & 2 & 4 & 0 & 1 & -1 & & 2 & 4 & 0 & 1 & -1 \\ \rightarrow & 1 & 2 & -2 & 0 & 3 & & 1 & 2 & -2 & 0 & 3 & & & & & & \\ \rightarrow & 0 & 0 & -1 & 3 & 5 & \rightarrow & 0 & 0 & 1 & -3 & -5 & & & & & & \\ & 0 & 0 & 4 & 1 & -7 & & 0 & 0 & 4 & 1 & -7 & & & & & & \\ \rightarrow & 1 & 2 & -2 & 0 & 3 & & 1 & 2 & -2 & 0 & 3 & & & & & & \\ \rightarrow & 0 & 0 & 1 & -3 & -5 & \rightarrow & 0 & 0 & 1 & -3 & -5 & \checkmark & & & & & \\ & 0 & 0 & 0 & 13 & 13 & & 0 & 0 & 0 & 1 & 1 & & & & & & \end{array}$$

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

**Példa:**

$$\begin{array}{cccccc} 1 & 2 & -2 & 0 & 3 & \\ 0 & 0 & 1 & -3 & -5 & \\ 0 & 0 & 0 & 1 & 1 & \end{array}$$



# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

**Példa:**

$$\begin{array}{cccccc} 1 & 2 & -2 & 0 & 3 & \rightarrow & 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & 1 & -3 & -5 & & 0 & 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 1 & 1 & & 0 & 0 & 0 & 1 & 1 \end{array}$$

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

**Példa:**

$$\begin{array}{cccccc} 1 & 2 & -2 & 0 & 3 & \rightarrow & 1 & 2 & -2 & 0 & 3 & \rightarrow & 1 & 2 & 0 & 0 & -1 \\ 0 & 0 & 1 & -3 & -5 & \rightarrow & 0 & 0 & 1 & 0 & -2 & \rightarrow & 0 & 0 & 1 & 0 & -2 \\ 0 & 0 & 0 & 1 & 1 & \rightarrow & 0 & 0 & 0 & 1 & 1 & \rightarrow & 0 & 0 & 0 & 1 & 1 \end{array} \quad \checkmark$$

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

(2) Ha csupán LA (vagy RLA) a cél, eltérhetünk a Gauss-eliminációtól, feltéve, hogy ESÁ-okkal dolgozunk.

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

(2) Ha csupán LA (vagy RLA) a cél, eltérhetünk a Gauss-eliminációtól, feltéve, hogy ESÁ-okkal dolgozunk.

**Példa:**

$$\begin{array}{ccccc} 3 & 2 & -2 & 0 & 3 \\ 2 & 1 & 0 & \sqrt{42} & \sqrt[3]{\pi} \end{array}$$

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

(2) Ha csupán LA (vagy RLA) a cél, eltérhetünk a Gauss-eliminációtól, feltéve, hogy ESÁ-okkal dolgozunk.

**Példa:** 
$$\begin{array}{cccccc} 3 & 2 & -2 & 0 & 3 & 1 & 1 & -2 & -\sqrt{42} & 3 - \frac{\varepsilon}{\sqrt{\pi}} \\ 2 & 1 & 0 & \sqrt{42} & \frac{\varepsilon}{\sqrt{\pi}} & 2 & 1 & 0 & \sqrt{42} & \frac{\varepsilon}{\sqrt{\pi}} \end{array} \rightarrow$$
 ✓

# Gauss-elimináció

**Gauss-elimináció:** Input:  $M \in \mathbb{R}^{n \times k}$  mátrix.

Output: Egy  $M$ -ből ESÁ-okkal kapható  $M' \in \mathbb{R}^{n \times k}$  LA mátrix.

Működés: Az algoritmus fázisokból áll. Az  $i$ -dik fázisban keresünk egy nemnulla elemet az  $(i - 1)$ -dik sor alatt a lehető legkisebb sorszámú oszlopban. Ha nincs ilyen elem, az algoritmus véget ér. Sorcserével ezt a nemnulla elemet az  $i$ -dik sorba visszük.

Az  $i$ -dik sor konstanssal szorzásával ezt az elemet  $v_1$ -sé alakítjuk.

Az  $i$ -dik sor alatti sorokhoz az  $i$ -dik sor konstansszorosát hozzáadva kinullázzuk a kapott  $v_1$  alatti elemeket.

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

(2) Ha csupán LA (vagy RLA) a cél, eltérhetünk a Gauss-eliminációtól, feltéve, hogy ESÁ-okkal dolgozunk.

# Gauss-elimináció

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

(2) Ha csupán LA (vagy RLA) a cél, eltérhetünk a Gauss-eliminációtól, feltéve, hogy ESÁ-okkal dolgozunk.



# Gauss-elimináció

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

(2) Ha csupán LA (vagy RLA) a cél, eltérhetünk a Gauss-eliminációtól, feltéve, hogy ESÁ-okkal dolgozunk.

(3) A Gauss-elimináció megvalósítható rekurzív algoritmusként is. A  $GE(M)$  (az  $M$  mátrixot lépcsős alakra hozó eljárás):

**1.** Ha  $M$  első oszlopa csupa0, akkor  $M'$  az első oszlop törlésével keletkező mátrix. Output:  $GE(M')$  elé írunk egy csupa0 oszlopot.

## Gauss-elimináció

**Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.

(2) Ha csupán LA (vagy RLA) a cél, eltérhetünk a Gauss-eliminációtól, feltéve, hogy ESÁ-okkal dolgozunk.

(3) A Gauss-elimináció megvalósítható rekurzív algoritmusként is. A  $GE(M)$  (az  $M$  mátrixot lépcsős alakra hozó eljárás):

**1.** Ha  $M$  első oszlopa csupa0, akkor  $M'$  az első oszlop törlésével keletkező mátrix. Output:  $GE(M')$  elé írunk egy csupa0 oszlopot.

**2.** Ha  $M$  első oszlopa tartalmaz nemnulla elemet, sorcserével és az első sor konstansszorosításával az első sor első elemét  $v_1$ -sé tesszük, majd a  $v_1$  alatti elemeket ESÁ-okkal kinullázzuk. Legyen  $M'$  az első sor és első oszlop törlésével keletkező mátrix. Output:  $GE(M')$  elé írunk egy csupa0 oszlopot és az így kapott mátrix fölé a korábban törölt első sort.

# Gauss-elimináció

- Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.
- (2) Ha csupán LA (vagy RLA) a cél, eltérhetünk a Gauss-eliminációtól, feltéve, hogy ESÁ-okkal dolgozunk.
- (3) A Gauss-elimináció megvalósítható rekurzív algoritmusként is.

## Gauss-elimináció

- Megj:** (1) A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden  $v_1$  felett kinullázhatók az elemek, ha a  $v_1$  sorának konstansszorosait a  $v_1$  feletti sorokhoz adjuk.
- (2) Ha csupán LA (vagy RLA) a cél, eltérhetünk a Gauss-eliminációtól, feltéve, hogy ESÁ-okkal dolgozunk.
- (3) A Gauss-elimináció megvalósítható rekurzív algoritmusként is.
- (4) Az  $M \in R^{n \times k}$  Gauss-eliminációja során minden fázisában legfeljebb egy sorcserét, legfeljebb  $2n$  sorszorzást és legfeljebb  $n$  sorösszeadást hajtunk végre. Ezért minden fázis legfeljebb  $konst \cdot nk$  lépést igényel, az összlépésszám legfeljebb  $konst \cdot n^2 k$ . Az input  $M$  mátrix  $n \cdot k$  elemet tartalmaz, az eljárás hatékony.

# Lineáris egyenletrendszerek megoldásszáma

**Láttuk:**

# Lineáris egyenletrendszerek megoldásszáma

## Láttuk:

1. Lineáris egyenletrendszer kib. egyhómxként is megadható.

# Lineáris egyenletrendszerek megoldásszáma

## Láttuk:

1. Lineáris egyenletrendszer kib. egyhómxként is megadható.
2. ESÁ nem változtat a megoldásokon

# Lineáris egyenletrendszerek megoldásszáma

## Láttuk:

1. Lineáris egyenletrendszer kib. egyhómxként is megadható.
2. ESÁ nem változtat a megoldásokon
3. ESÁ-okkal elérhető a RLA



# Lineáris egyenletrendszerek megoldásszáma

## Láttuk:

1. Lineáris egyenletrendszer kib. egyhómxként is megadható.
2. ESÁ nem változtat a megoldásokon
3. ESÁ-okkal elérhető a RLA
4. A RLA-ból azonnal adódik a megoldás

# Lineáris egyenletrendszerek megoldásszáma

## Láttuk:

1. Lineáris egyenletrendszer kib. egyhómxként is megadható.
2. ESÁ nem változtat a megoldásokon
3. ESÁ-okkal elérhető a RLA
4. A RLA-ból azonnal adódik a megoldás
  - ▶ Ha az utolsó oszlopban van  $v_1$ , akkor nincs megoldás

# Lineáris egyenletrendszerek megoldásszáma

## Láttuk:

1. Lineáris egyenletrendszer kib. egyhómxként is megadható.
2. ESÁ nem változtat a megoldásokon
3. ESÁ-okkal elérhető a RLA
4. A RLA-ból azonnal adódik a megoldás
  - ▶ Ha az utolsó oszlopban van  $v_1$ , akkor nincs megoldás
  - ▶ Ha az utolsó kivételével minden oszlopban van  $v_1$ , akkor egyetlen megoldás van

# Lineáris egyenletrendszerek megoldásszáma

## Láttuk:

1. Lineáris egyenletrendszer kib. egyhómxként is megadható.
2. ESÁ nem változtat a megoldásokon
3. ESÁ-okkal elérhető a RLA
4. A RLA-ból azonnal adódik a megoldás
  - ▶ Ha az utolsó oszlopban van  $v_1$ , akkor nincs megoldás
  - ▶ Ha az utolsó kivételével minden oszlopban van  $v_1$ , akkor egyetlen megoldás van
  - ▶ Ha az utolsón kívül más oszlopban sincs  $v_1$ , akkor van szabad paraméter, így végtelen sok különböző megoldás van.

# Lineáris egyenletrendszerek megoldásszáma

## Láttuk:

1. Lineáris egyenletrendszer kib. egyhómxként is megadható.
2. ESÁ nem változtat a megoldásokon
3. ESÁ-okkal elérhető a RLA
4. A RLA-ból azonnal adódik a megoldás
  - ▶ Ha az utolsó oszlopban van  $v_1$ , akkor nincs megoldás
  - ▶ Ha az utolsó kivételével minden oszlopban van  $v_1$ , akkor egyetlen megoldás van
  - ▶ Ha az utolsón kívül más oszlopban sincs  $v_1$ , akkor van szabad paraméter, így végtelen sok különböző megoldás van.

**Köv:** Ha a lineáris egyenletrendszernek pontosan egy megoldása van, akkor legalább annyi egyenlet van, mint ahány ismeretlen.

# Lineáris egyenletrendszerek megoldásszáma

## Láttuk:

1. Lineáris egyenletrendszer kib. egyhómxként is megadható.
2. ESÁ nem változtat a megoldásokon
3. ESÁ-okkal elérhető a RLA
4. A RLA-ból azonnal adódik a megoldás
  - ▶ Ha az utolsó oszlopban van  $v_1$ , akkor nincs megoldás
  - ▶ Ha az utolsó kivételével minden oszlopban van  $v_1$ , akkor egyetlen megoldás van
  - ▶ Ha az utolsón kívül más oszlopban sincs  $v_1$ , akkor van szabad paraméter, így végtelen sok különböző megoldás van.

**Köv:** Ha a lineáris egyenletrendszernek pontosan egy megoldása van, akkor legalább annyi egyenlet van, mint ahány ismeretlen.

**Biz:** Az RLA-ra hozás után nincs szabad paraméter, tehát minden változóhoz tartozik  $v_1$ . Ezért a kibővített együtthatómátrixnak legalább annyi sora van, mint a változók száma. □

# Lineáris egyenletrendszerek megoldásszáma

## Láttuk:

1. Lineáris egyenletrendszer kib. egyhómxként is megadható.
2. ESÁ nem változtat a megoldásokon
3. ESÁ-okkal elérhető a RLA
4. A RLA-ból azonnal adódik a megoldás
  - ▶ Ha az utolsó oszlopban van  $v_1$ , akkor nincs megoldás
  - ▶ Ha az utolsó kivételével minden oszlopban van  $v_1$ , akkor egyetlen megoldás van
  - ▶ Ha az utolsó kívül más oszlopban sincs  $v_1$ , akkor van szabad paraméter, így végtelen sok különböző megoldás van.

**Köv:** Ha a lineáris egyenletrendszernek pontosan egy megoldása van, akkor legalább annyi egyenlet van, mint ahány ismeretlen.

**Biz:** Az RLA-ra hozás után nincs szabad paraméter, tehát minden változóhoz tartozik  $v_1$ . Ezért a kibővített együtthatómátrixnak legalább annyi sora van, mint a változók száma. □

**Megj:** A fenti következmény fordított irányban nem igaz, és lényegében nincs más összefüggés az egyértelmű megoldhatóság, az ismeretlenek és egyenletek száma között.

**Köszönöm a figyelmet!**