

A számítástudomány alapjai

Lineáris egyenletrendszerek

2025. november 4.

Munkaterv

Munkaterv

- ▶ Többismeretlenes, lineáris egyenletrendszerek megoldásával foglalkozunk.

Munkaterv

- ▶ Többismeretlenes, lineáris egyenletrendszerek megoldásával foglalkozunk.
- ▶ Az egyenletek sokszori leírása és másolása helyett egy tömörebb, táblázattal operáló módszert mutatunk be, aminek segítségével bármely lineáris egyenletrendszer összes megoldását megtalálhatjuk, és kompakt formában felírhatjuk.

Munkaterv

- ▶ Többismeretlenes, lineáris egyenletrendszerek megoldásával foglalkozunk.
- ▶ Az egyenletek sokszori leírása és másolása helyett egy tömörebb, táblázattal operáló módszert mutatunk be, aminek segítségével bármely lineáris egyenletrendszer összes megoldását megtalálhatjuk, és kompakt formában felírhatjuk.
- ▶ Megadunk egy algoritmust, amivel bármely lineáris egyenletrendszer hatékonyan megoldható.

Munkaterv

- ▶ Többismeretlenes, lineáris egyenletrendszerek megoldásával foglalkozunk.
- ▶ Az egyenletek sokszori leírása és másolása helyett egy tömörebb, táblázattal operáló módszert mutatunk be, aminek segítségével bármely lineáris egyenletrendszer összes megoldását megtalálhatjuk, és kompakt formában felírhatjuk.
- ▶ Megadunk egy algoritmust, amivel bármely lineáris egyenletrendszer hatékonyan megoldható.
- ▶ Hasznos megállapításokat teszünk a megoldások struktúrájáról.

Lineáris egyenletrendszerek naiv megoldása

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

Lineáris egyenletrendszerek naiv megoldása

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

Lineáris egyenletrendszerek naiv megoldása

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

Megoldás: Olyan értékadás, ami minden egyenletet igazzá tesz.

Lineáris egyenletrendszerek naiv megoldása

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

Megoldás: Olyan értékadás, ami minden egyenletet igazzá tesz.

Példa:

$$3x - 4z = 666$$

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

Lineáris egyenletrendszerek naiv megoldása

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

Megoldás: Olyan értékadás, ami minden egyenletet igazzá tesz.

Példa:

$$3x - 4z = 666$$

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

$$\pi^{e^{\pi}} y - (\ln(\cos 42)) \cdot z = \sqrt[\pi]{\sqrt[4]{2} \sqrt{42}}$$

Lineáris egyenletrendszerek naiv megoldása

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

Megoldás: Olyan értékadás, ami minden egyenletet igazgá tesz.

Lineáris egyenletrendszerek naiv megoldása

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.

Lineáris egyenletrendszerek naiv megoldása

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 egy egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe behelyettesítjük. Az így kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

Lineáris egyenletrendszerek naiv megoldása

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 egy egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe behelyettesítjük. Az így kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

A megoldás formája: Az eljárás végére vagy elfogynak az egyenletek vagy a kapott egyenletek nem tartalmazznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás.

Lineáris egyenletrendszerek naív megoldása

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.

Naív módszer: Minden fázisban egy egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe behelyettesítjük. Az így kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

A megoldás formája: Az eljárás végére vagy elfogynak az egyenletek vagy a kapott egyenletek nem tartalmazznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás.

Lineáris egyenletrendszerek naív megoldása

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.

Naív módszer: Minden fázisban egy egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe behelyettesítjük. Az így kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

A megoldás formája: Az eljárás végére vagy elfogynak az egyenletek vagy a kapott egyenletek nem tartalmazznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mindegyik azonosság, akkor a ki nem fejezett ismeretlenek bármely értékádása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

Lineáris egyenletrendszer naiv megoldása

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 egy egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe behelyettesítjük. Az így kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

A megoldás formája: Az eljárás végére vagy elfogynak az egyenletek vagy a kapott egyenletek nem tartalmazznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mindegyik azonosság, akkor a ki nem fejezett ismeretlenek bármely értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

Példa:

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

Lineáris egyenletrendszerek naiv megoldása

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 egy egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe behelyettesítjük. Az így kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

A megoldás formája: Az eljárás végére vagy elfogynak az egyenletek vagy a kapott egyenletek nem tartalmazznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mindegyik azonosság, akkor a ki nem fejezett ismeretlenek bármely értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

Példa:

Lineáris egyenletrendszer naiv megoldása

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 egy egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe behelyettesítjük. Az így kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

A megoldás formája: Az eljárás végére vagy elfogynak az egyenletek vagy a kapott egyenletek nem tartalmazznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mindegyik azonosság, akkor a ki nem fejezett ismeretlenek bármely értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

Példa:

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

Lineáris egyenletrendszerek naiv megoldása

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 egy egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe behelyettesítjük. Az így kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

A megoldás formája: Az eljárás végére vagy elfogynak az egyenletek vagy a kapott egyenletek nem tartalmazznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mindegyik azonosság, akkor a ki nem fejezett ismeretlenek bármely értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

Példa:

Lineáris egyenletrendszer naiv megoldása

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 egy egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe behelyettesítjük. Az így kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

A megoldás formája: Az eljárás végére vagy elfogynak az egyenletek vagy a kapott egyenletek nem tartalmazznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mindegyik azonosság, akkor a ki nem fejezett ismeretlenek bármely értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

Példa:

$x - 2y = 7$	$\boxed{x} = 7 + 2y$	$\boxed{x} = 7 + 2y$	$y = \alpha \in \mathbb{R}$ $x = 7 + 2\alpha$
$3x - 6y = 21$	$21 + 6y - 6y = 21$	$21 = 21 \quad \checkmark$	
$-2x + 4y = -14$	$-14 - 4y + 4y = -14$	$-14 = -14 \quad \checkmark$	

Lineáris egyenletrendszerek naív megoldása

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.

Naív módszer: Minden fázisban egy egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe behelyettesítjük. Az így kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

A megoldás formája: Az eljárás végére vagy elfogynak az egyenletek vagy a kapott egyenletek nem tartalmazznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mindegyik azonosság, akkor a ki nem fejezett ismeretlenek bármely értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

Lineáris egyenletrendszer naiv megoldása

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 egy egyenletből kifejezünk egy ismeretlent, és ezt a többi egyenletbe behelyettesítjük. Az így kapott egyenletekkel végezzük a következő fázist. Minden fázisban 1-gyel csökken az ismeretlenek és az egyenletek száma is.

A megoldás formája: Az eljárás végére vagy elfogynak az egyenletek vagy a kapott egyenletek nem tartalmazznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mindegyik azonosság, akkor a ki nem fejezett ismeretlenek bármely értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

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

(Elemi) sorkvivalens 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:

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

(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:

$$\begin{array}{rcl} x_1 - 3x_3 + 5x_4 = -6 \\ 7x_1 + 2x_2 + 3x_3 = 9 \\ x_2 + 7x_3 - 2x_4 = 11 \end{array} \mapsto \begin{pmatrix} & x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 0 & -3 & 5 & -6 \\ 7 & 2 & 3 & 0 & 9 \\ 0 & 1 & 7 & -2 & 11 \end{pmatrix}$$

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:

$$\begin{array}{rcl} x_1 - 3x_3 + 5x_4 = -6 \\ 7x_1 + 2x_2 + 3x_3 = 9 \\ x_2 + 7x_3 - 2x_4 = 11 \end{array} \mapsto \begin{pmatrix} & x_1 & x_2 & x_3 & x_4 & \\ \begin{array}{c} 1 \\ 7 \\ 0 \end{array} & \begin{array}{c} 0 \\ 2 \\ 1 \end{array} & \begin{array}{c} -3 \\ 3 \\ 7 \end{array} & \begin{array}{c} 5 \\ 0 \\ -2 \end{array} & \left| \begin{array}{c} -6 \\ 9 \\ 11 \end{array} \right. \end{pmatrix}$$

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

Megoldás módszere: Ekvivalens átalakításokat végzünk, amelyek 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, (4) csupa0 sor hozzáadása vagy elhagyása.

(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, (4) csupa0 sor hozzáadása vagy elhagyása.

Megj: Sorekvivalens átalakítás (de nem ESÁ) az i -dik sor cseréje az i -dik sor és a j -dik sor konstansszorosának összegére.

(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, (4) csupa0 sor hozzáadása vagy elhagyása.

Megj: Sorekvivalens átalakítás (de nem ESÁ) az i -dik sor cseréje az i -dik sor és a j -dik sor konstansszorosának összegére.

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, (4) csupa0 sor hozzáadása vagy elhagyása.

Megj: Sorekvivalens átalakítás (de nem ESÁ) az i -dik sor cseréje az i -dik sor és a j -dik sor konstansszorosának összegére.

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 végigszorozása, (3) az i -dik sor helyettesítése az i -dik és j -dik sorok (koordinátánkénti) összegével, (4) csupa0 sor hozzáadása vagy elhagyása.

Megj: Sorekvivalens átalakítás (de nem ESÁ) az i -dik sor cseréje az i -dik sor és a j -dik sor konstansszorosának összegére.

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, (4) csupa0 sor hozzáadása vagy elhagyása.

Megj: Sorekvivalens átalakítás (de nem ESÁ) az i -dik sor cseréje az i -dik sor és a j -dik sor konstansszorosának összegére.

Állítás: ESÁ hatására a megoldások halmaza nem változik.

Fő cél: Tetszőleges lineáris egyenletrendszer esetén ESÁ-okkal elérni, hogy a kapott rendszer a megoldások leírását adja.

(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 végigszorítása, (3) az i -dik sor helyettesítése az i -dik és j -dik sorok (koordinátánkénti) összegével, (4) csupa0 sor hozzáadása vagy elhagyása.

Megj: Sorekvivalens átalakítás (de nem ESÁ) az i -dik sor cseréje az i -dik sor és a j -dik sor konstansszorosának összegére.

Állítás: ESÁ hatására a megoldások halmaza nem változik.

Fő cél: Tetszőleges lineáris egyenletrendszer esetén ESÁ-okkal elérni, hogy a kapott rendszer a megoldások leírását adja.

Részcél: Megfogalmazzuk a kibővített együtthatómátrixnak azt a tulajdonságát, amit elérve minden megoldás könnyen kiolvasható.

(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 $v1$.

Példa: LA mátrix

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

(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 (4) M -ben minden $v1$ felett csak nullák állnak.

Példa: LA mátrix

RLA mátrix

$$\begin{pmatrix} \textcolor{red}{1} & 3 & -2 & 0 & 1 & 1 & 7 \\ 0 & 0 & \textcolor{red}{1} & 2 & 0 & 5 & -2 \\ 0 & 0 & 0 & 0 & \textcolor{red}{1} & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & \textcolor{red}{1} & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ ill. } \begin{pmatrix} 0 & \textcolor{red}{1} & 0 & -2 & 0 & 1 & 0 \\ 0 & 0 & \textcolor{red}{1} & 2 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & \textcolor{red}{1} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \textcolor{red}{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(Redukált) lépcsős alak

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

- (1) minden sor első nemnulla eleme 1-es (ú.n. **vezér1-es**, avagy **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 (4) M -ben minden $v1$ 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 $v1$ feletti sorban van ettől a $v1$ -től balra eső másik $v1$.

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

- (3) M LA és (4) M -ben minden $v1$ felett csak nullák állnak.

Példa: RLA kibővített egyhómx:

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

(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 (4) M -ben minden $v1$ felett csak nullák állnak.

Példa: RLA kibővített egyhómx:

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

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

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

- (3) M LA és (4) M -ben minden $v1$ felett csak nullák állnak.

Példa: RLA kibővített egyhómx:

$$\begin{pmatrix} \overset{x_1}{1} & x_2 & \overset{x_3}{0} & \overset{x_4}{-2} & \bigg| & 1 \\ 0 & 0 & 1 & 5 & \bigg| & 7 \end{pmatrix} \mapsto \begin{array}{l} \overset{x_1}{1} + 3x_2 - 2x_4 = 1 \\ \qquad \qquad \qquad \overset{x_3}{1} + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2 = \alpha \in \mathbb{R} \text{ tetsz.} \\ x_4 = \beta \in \mathbb{R} \text{ tetsz.} \\ \overset{x_1}{1} = 1 - 3\alpha + 2\beta \\ \overset{x_3}{1} = 7 - 5\beta \end{array}$$

(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
- (4) M -ben minden v1 felett csak nullák állnak.

Példa: RLA kibővített egyhómx:

$$\left(\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \right) \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2 = \alpha \in \mathbb{R} \text{ tetsz.} \\ x_4 = \beta \in \mathbb{R} \text{ tetsz.} \\ x_1 = 1 - 3\alpha + 2\beta \\ x_3 = 7 - 5\beta \end{array}$$

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

(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
- (4) M -ben minden v1 felett csak nullák állnak.

Példa: RLA kibővített egyhómx:

$$\left(\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \right) \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2 = \alpha \in \mathbb{R} \text{ tetsz.} \\ x_4 = \beta \in \mathbb{R} \text{ tetsz.} \\ x_1 = 1 - 3\alpha + 2\beta \\ x_3 = 7 - 5\beta \end{array}$$

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

(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
- (4) M -ben minden v1 felett csak nullák állnak.

Példa: RLA kibővített egyhómx:

$$\left(\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \right) \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2 = \alpha \in \mathbb{R} \text{ tetsz.} \\ x_4 = \beta \in \mathbb{R} \text{ tetsz.} \\ x_1 = 1 - 3\alpha + 2\beta \\ x_3 = 7 - 5\beta \end{array}$$

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

(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
- (4) M -ben minden v1 felett csak nullák állnak.

Példa: RLA kibővített egyhómx:

$$\left(\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \right) \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2 = \alpha \in \mathbb{R} \text{ tetsz.} \\ x_4 = \beta \in \mathbb{R} \text{ tetsz.} \\ x_1 = 1 - 3\alpha + 2\beta \\ x_3 = 7 - 5\beta \end{array}$$

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

Def: Kib. egyhómx **tilos sora**: $0 \quad \dots \quad 0 \mid x$ 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
- (4) M -ben minden v1 felett csak nullák állnak.

Példa: RLA kibővített egyhómx:

$$\left(\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \right) \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2 = \alpha \in \mathbb{R} \text{ tetsz.} \\ x_4 = \beta \in \mathbb{R} \text{ tetsz.} \\ x_1 = 1 - 3\alpha + 2\beta \\ x_3 = 7 - 5\beta \end{array}$$

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

Def: Kib. egyhómx **tilos sora**: $0 \quad \dots \quad 0 \mid x$ 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
- (4) M -ben minden v1 felett csak nullák állnak.

Példa: RLA kibővített egyhómx:

$$\left(\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \right) \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2 = \alpha \in \mathbb{R} \text{ tetsz.} \\ x_4 = \beta \in \mathbb{R} \text{ tetsz.} \\ x_1 = 1 - 3\alpha + 2\beta \\ x_3 = 7 - 5\beta \end{array}$$

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
- (4) M -ben minden v1 felett csak nullák állnak.

Példa: RLA kibővített egyhómx:

$$\left(\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \right) \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2 = \alpha \in \mathbb{R} \text{ tetsz.} \\ x_4 = \beta \in \mathbb{R} \text{ tetsz.} \\ x_1 = 1 - 3\alpha + 2\beta \\ x_3 = 7 - 5\beta \end{array}$$

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
- (4) M -ben minden v1 felett csak nullák állnak.

Példa: RLA kibővített egyhómx:

$$\left(\begin{array}{cccc|c} x_1 & x_2 & x_3 & x_4 & \\ \hline 1 & 3 & 0 & -2 & 1 \\ 0 & 0 & 1 & 5 & 7 \end{array} \right) \mapsto \begin{array}{l} x_1 + 3x_2 - 2x_4 = 1 \\ x_3 + 5x_4 = 7 \end{array} \mapsto \begin{array}{l} x_2 = \alpha \in \mathbb{R} \text{ tetsz.} \\ x_4 = \beta \in \mathbb{R} \text{ tetsz.} \\ x_1 = 1 - 3\alpha + 2\beta \\ x_3 = 7 - 5\beta \end{array}$$

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, akkor a szabad paraméterek minden lehetséges értékadásához pontosan egy 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 $v1$ feletti sorban van ettől a $v1$ -től balra eső másik $v1$.

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

- (3) M LA és (4) 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 változó **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, akkor a szabad paraméterek minden lehetséges értékadásához pontosan egy 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 $v1$ feletti sorban van ettől a $v1$ -től balra eső másik $v1$.

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

(3) M LA és (4) 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, akkor a szabad paraméterek minden lehetséges értékadásához pontosan egy megoldás tartozik.

Megf: A lin. egyenletsrz. megoldása azt jelenti, hogy a lin. egyenletsrz. 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 (4) 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, akkor a szabad paraméterek minden lehetséges értékadásához pontosan egy megoldás tartozik.

Megf: A lin. egyenletshr. megoldása azt jelenti, hogy a lin. egyenletshr. egy RLA kibővített egyhómx-szal van megadva.

Új cél: Egy eljárás, ami tetsz. mátrixot ESÁ-okkal 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. Ha van, 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.

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. Ha van, 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:

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

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. Ha van, 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:

$$\begin{pmatrix} 0 & 0 & -1 & 3 & 5 \\ 3 & 6 & -6 & 0 & 9 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \sim \begin{pmatrix} 3 & 6 & -6 & 0 & 9 \\ 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix}$$

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. Ha van, 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:

$$\begin{pmatrix} 0 & 0 & -1 & 3 & 5 \\ 3 & 6 & -6 & 0 & 9 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \sim \begin{pmatrix} 3 & 6 & -6 & 0 & 9 \\ 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix}$$

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. Ha van, 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:

$$\begin{pmatrix} 0 & 0 & -1 & 3 & 5 \\ 3 & 6 & -6 & 0 & 9 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \sim \begin{pmatrix} 3 & 6 & -6 & 0 & 9 \\ 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \\ \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & -1 & 3 & 5 \\ 0 & 0 & 4 & 1 & -7 \end{pmatrix}$$

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. Ha van, 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:

$$\begin{pmatrix} 0 & 0 & -1 & 3 & 5 \\ 3 & 6 & -6 & 0 & 9 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \sim \begin{pmatrix} 3 & 6 & -6 & 0 & 9 \\ 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & 2 & -2 & 0 & 3 \\ 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \\ \sim \begin{pmatrix} \color{red}{1} & 2 & -2 & 0 & 3 \\ 0 & 0 & -1 & 3 & 5 \\ 0 & 0 & 4 & 1 & -7 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & 2 & -2 & 0 & 3 \\ 0 & 0 & \color{red}{1} & -3 & -5 \\ 0 & 0 & 4 & 1 & -7 \end{pmatrix}$$

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. Ha van, 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:

$$\begin{pmatrix} 0 & 0 & -1 & 3 & 5 \\ 3 & 6 & -6 & 0 & 9 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \sim \begin{pmatrix} 3 & 6 & -6 & 0 & 9 \\ 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \\ \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & -1 & 3 & 5 \\ 0 & 0 & 4 & 1 & -7 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & 1 & -3 & -5 \\ 0 & 0 & 4 & 1 & -7 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & 1 & -3 & -5 \\ 0 & 0 & 0 & 13 & 13 \end{pmatrix}$$

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. Ha van, 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:

$$\begin{pmatrix} 0 & 0 & -1 & 3 & 5 \\ 3 & 6 & -6 & 0 & 9 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \sim \begin{pmatrix} 3 & 6 & -6 & 0 & 9 \\ 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & -1 & 3 & 5 \\ 2 & 4 & 0 & 1 & -1 \end{pmatrix} \\ \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & -1 & 3 & 5 \\ 0 & 0 & 4 & 1 & -7 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & 1 & -3 & -5 \\ 0 & 0 & 4 & 1 & -7 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & 1 & -3 & -5 \\ 0 & 0 & 0 & 13 & 13 \end{pmatrix} \\ \sim \begin{pmatrix} 1 & 2 & -2 & 0 & 3 \\ 0 & 0 & 1 & -3 & -5 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \checkmark$$

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. Ha van, 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.

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. Ha van, 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.

Megf: A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden v1 felett kinullázhatók az elemek, ha a v1 sorának konstansszorosait a v1 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. Ha van, 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.

Megf: A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden v1 felett kinullázhatók az elemek, ha a v1 sorának konstansszorosait a v1 feletti sorokhoz adjuk.

Példa:

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

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. Ha van, 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.

Megf: A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden v1 felett kinullázhatók az elemek, ha a v1 sorának konstansszorosait a v1 feletti sorokhoz adjuk.

Példa:

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

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. Ha van, 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.

Megf: A Gauss-elimináció outputja LA. Az RLA-hoz további lépésekre van szükség: minden v1 felett kinullázhatók az elemek, ha a v1 sorának konstansszorosait a v1 feletti sorokhoz adjuk.

Példa:

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

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ 2 & 1 & -3 & 1 \end{pmatrix}$$

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \textcolor{red}{1} & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 2 & 1 & -3 & 1 \end{pmatrix}$$

$$\boxed{I} \mapsto \boxed{I}/13$$

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 0 & \frac{9}{13} & -\frac{53}{13} & \frac{21}{13} \end{pmatrix} \quad \text{Yeah.}$$

$\boxed{\text{II}} \mapsto \boxed{\text{II}} - 2 \cdot \boxed{\text{I}}$

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 0 & \frac{9}{13} & -\frac{53}{13} & \frac{21}{13} \end{pmatrix} \quad \text{Yeah.}$$

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 0 & \frac{9}{13} & -\frac{53}{13} & \frac{21}{13} \end{pmatrix} \quad \text{Yeah.}$$

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ 2 & 1 & -3 & 1 \end{pmatrix}$$

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 0 & \frac{9}{13} & -\frac{53}{13} & \frac{21}{13} \end{pmatrix} \quad \text{Yeah.}$$

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & -4 & 25 & -10 \\ 2 & 1 & -3 & 1 \end{pmatrix}$$

$$\boxed{\text{I}} \mapsto \boxed{\text{I}} - 6 \cdot \boxed{\text{II}}$$

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 0 & \frac{9}{13} & -\frac{53}{13} & \frac{21}{13} \end{pmatrix} \quad \text{Yeah.}$$

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & -4 & 25 & -10 \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} \color{red}{1} & -4 & 18 & -10 \\ 0 & 9 & -53 & 21 \end{pmatrix} \quad \checkmark$$

$\boxed{\text{II}} \mapsto \boxed{\text{II}} - 2 \cdot \boxed{\text{I}}$

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

(2) A Gauss-elimináció megfogalmazható rekurzív algoritmusként is.

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

(2) A Gauss-elimináció megfogalmazható 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.

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

(2) A Gauss-elimináció megfogalmazható 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 konstanssal szorzásával 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.

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

(2) A Gauss-elimináció megfogalmazható rekurzív algoritmusként is.

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami mindig működik, de néha igen küzdelmes. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

(2) A Gauss-elimináció megfogalmazható rekurzív algoritmusként is.

(3) 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 $\text{konst} \cdot nk$ lépést igényel, az összlépésszám legfeljebb $\text{konst} \cdot n^2k$. 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 egyenletrendszer 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 egyenletrendszer 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 egyenletrendszer 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 egyenletrendszer 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. □

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

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.

$$\begin{array}{l} -2x_1 - x_2 + 7x_3 = 0 \\ 5x_1 + 3x_2 - 2x_3 = 0 \\ 2x_1 - 4x_2 + x_3 = 0 \end{array} \mapsto \left(\begin{array}{ccc|c} x_1 & x_2 & x_3 & \\ -2 & -1 & 7 & 0 \\ 5 & 3 & -2 & 0 \\ 2 & -4 & 1 & 0 \end{array} \right) \mapsto \left(\begin{array}{ccc|c} x_1 & x_2 & x_3 & \\ -2 & -1 & 7 & \\ 5 & 3 & -2 & \\ 2 & -4 & 1 & \end{array} \right)$$

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.
(2) HLE megoldása során sosem jelenhet meg tilos sor, ezért mindig van megoldás.

Homogén lineáris egyenletrendszer

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.
(2) HLE megoldása során sosem jelenhet meg tilos sor, ezért mindig van megoldás. És csakugyan: a $(0, 0, \dots, 0)$ mindig jó.
Ha nincs szabad paraméter, akkor a csupa-0 az egyetlen megoldás.
Ha van, akkor minden kötött változó a szabad paraméterek egy rögzített összegével egyenlő.

$$\begin{array}{ccccc|c} x_1 & x_2 & x_3 & x_4 & x_5 & \\ \left(\begin{array}{ccccc|c} 1 & 0 & -2 & 0 & 0 & 0 \\ 0 & 1 & 7 & 0 & -2 & 0 \\ 0 & 0 & & 1 & 7 & 0 \end{array} \right) & \begin{array}{l} x_3 = \alpha \in \mathbb{R} \text{ tetsz.} \\ x_5 = \beta \in \mathbb{R} \text{ tetsz.} \\ x_1 = 2\alpha \\ x_3 = -7\alpha + 2\beta \\ x_4 = -7\beta \end{array} \end{array}$$

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.
(2) HLE megoldása során sosem jelenhet meg tilos sor, ezért mindig van megoldás. És csakugyan: a $(0, 0, \dots, 0)$ mindig jó.
Ha nincs szabad paraméter, akkor a csupa-0 az egyetlen megoldás.
Ha van, akkor minden kötött változó a szabad paraméterek egy rögzített összegével egyenlő.

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.

(2) HLE megoldása során sosem jelenhet meg tilos sor, ezért mindig van megoldás. És csakugyan: a $(0, 0, \dots, 0)$ mindig jó. Ha nincs szabad paraméter, akkor a csupa-0 az egyetlen megoldás. Ha van, akkor minden kötött változó a szabad paraméterek egy rögzített összegével egyenlő.

(3) Tfh $(x_1, \dots, x_n)$ és $(y_1, \dots, y_n)$ is megoldja ugyanazt a HLE-t. Ekkor $(\lambda x_1, \dots, \lambda x_n)$ és $(x_1 + y_1, \dots, x_n + y_n)$ is a HLE megoldásai.

$$-2x_1 - x_2 + 7x_3 = 0 \quad -2(\lambda x_1) - (\lambda x_2) + 7(\lambda x_3) = 0$$

$$5x_1 + 3x_2 - 2x_3 = 0 \Rightarrow 5(\lambda x_1) + 3(\lambda x_2) - 2(\lambda x_3) = 0$$

$$2x_1 - 4x_2 + x_3 = 0 \quad 2(\lambda x_1) - 4(\lambda x_2) + (\lambda x_3) = 0$$

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.

(2) HLE megoldása során sosem jelenhet meg tilos sor, ezért mindig van megoldás. És csakugyan: a $(0, 0, \dots, 0)$ mindig jó. Ha nincs szabad paraméter, akkor a csupa-0 az egyetlen megoldás. Ha van, akkor minden kötött változó a szabad paraméterek egy rögzített összegével egyenlő.

(3) Tfh $(x_1, \dots, x_n)$ és $(y_1, \dots, y_n)$ is megoldja ugyanazt a HLE-t. Ekkor $(\lambda x_1, \dots, \lambda x_n)$ és $(x_1 + y_1, \dots, x_n + y_n)$ is a HLE megoldásai.

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.

(2) HLE megoldása során sosem jelenhet meg tilos sor, ezért mindig van megoldás. És csakugyan: a $(0, 0, \dots, 0)$ mindig jó. Ha nincs szabad paraméter, akkor a csupa-0 az egyetlen megoldás. Ha van, akkor minden kötött változó a szabad paraméterek egy rögzített összegével egyenlő.

(3) Tfh $(x_1, \dots, x_n)$ és $(y_1, \dots, y_n)$ is megoldja ugyanazt a HLE-t. Ekkor $(\lambda x_1, \dots, \lambda x_n)$ és $(x_1 + y_1, \dots, x_n + y_n)$ is a HLE megoldásai.

$$\begin{array}{lll} -2x_1 - x_2 + 7x_3 = 0 & -2y_1 - y_2 + 7y_3 = 0 & -2(x_1 + y_1) - (x_2 + y_2) + 7(x_3 + y_3) = 0 \\ 5x_1 + 3x_2 - 2x_3 = 0, & 5y_1 + 3y_2 - 2y_3 = 0 \Rightarrow & 5(x_1 + y_1) + 3(x_2 + y_2) - 2(x_3 + y_3) = 0 \\ 2x_1 - 4x_2 + x_3 = 0 & 2y_1 - 4y_2 + y_3 = 0 & 2(x_1 + y_1) - 4(x_2 + y_2) + (x_3 + y_3) = 0 \end{array}$$

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.

(2) HLE megoldása során sosem jelenhet meg tilos sor, ezért mindig van megoldás. És csakugyan: a $(0, 0, \dots, 0)$ mindig jó. Ha nincs szabad paraméter, akkor a csupa-0 az egyetlen megoldás. Ha van, akkor minden kötött változó a szabad paraméterek egy rögzített összegével egyenlő.

(3) Tfh $(x_1, \dots, x_n)$ és $(y_1, \dots, y_n)$ is megoldja ugyanazt a HLE-t. Ekkor $(\lambda x_1, \dots, \lambda x_n)$ és $(x_1 + y_1, \dots, x_n + y_n)$ is a HLE megoldásai.

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.

(2) HLE megoldása során sosem jelenhet meg tilos sor, ezért mindig van megoldás. És csakugyan: a $(0, 0, \dots, 0)$ mindig jó. Ha nincs szabad paraméter, akkor a csupa-0 az egyetlen megoldás. Ha van, akkor minden kötött változó a szabad paraméterek egy rögzített összegével egyenlő.

(3) Tfh $(x_1, \dots, x_n)$ és $(y_1, \dots, y_n)$ is megoldja ugyanazt a HLE-t. Ekkor $(\lambda x_1, \dots, \lambda x_n)$ és $(x_1 + y_1, \dots, x_n + y_n)$ is a HLE megoldásai.

(4) Tfh $(x_1, \dots, x_n)$ és $(y_1, \dots, y_n)$ is megoldja ugyanazt a lin. egy.rsz-t. Ekkor $(x_1 - y_1, \dots, x_n - y_n)$ megoldja a megfelelő HLE-t.

$$\begin{array}{lll} -2x_1 - x_2 + 7x_3 = \pi & -2y_1 - y_2 + 7y_3 = \pi & -2(x_1 - y_1) - (x_2 - y_2) + 7(x_3 - y_3) = 0 \\ 5x_1 + 3x_2 - 2x_3 = \sqrt{5}, & 5y_1 + 3y_2 - 2y_3 = \sqrt{5} \Rightarrow & 5(x_1 - y_1) + 3(x_2 - y_2) - 2(x_3 - y_3) = 0 \\ 2x_1 - 4x_2 + x_3 = 42 & 2y_1 - 4y_2 + y_3 = 42 & 2(x_1 - y_1) - 4(x_2 - y_2) + (x_3 - y_3) = 0 \end{array}$$

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.

(2) HLE megoldása során sosem jelenhet meg tilos sor, ezért mindig van megoldás. És csakugyan: a $(0, 0, \dots, 0)$ mindig jó. Ha nincs szabad paraméter, akkor a csupa-0 az egyetlen megoldás. Ha van, akkor minden kötött változó a szabad paraméterek egy rögzített összegével egyenlő.

(3) Tfh $(x_1, \dots, x_n)$ és $(y_1, \dots, y_n)$ is megoldja ugyanazt a HLE-t. Ekkor $(\lambda x_1, \dots, \lambda x_n)$ és $(x_1 + y_1, \dots, x_n + y_n)$ is a HLE megoldásai.

(4) Tfh $(x_1, \dots, x_n)$ és $(y_1, \dots, y_n)$ is megoldja ugyanazt a lin. egy.rsz-t. Ekkor $(x_1 - y_1, \dots, x_n - y_n)$ megoldja a megfelelő HLE-t.

Homogén lineáris egyenletrendszerek

Def: A **homogén lineáris egyenletrendszer (HLE)** olyan egyenletrendszer, amiben minden egyenlet jobb oldalán 0 áll.

Megf: (1) HLE kib. egyhómx-ának kibővítő oszlopa csupa 0-ból áll. ESÁ nyomán itt soha nem jelenhet meg 0-tól különböző érték, ezért HLE megoldásakor sokszor ki se írjuk a kibővítő oszlopot.

(2) HLE megoldása során sosem jelenhet meg tilos sor, ezért mindig van megoldás. És csakugyan: a $(0, 0, \dots, 0)$ mindig jó. Ha nincs szabad paraméter, akkor a csupa-0 az egyetlen megoldás. Ha van, akkor minden kötött változó a szabad paraméterek egy rögzített összegével egyenlő.

(3) Tfh $(x_1, \dots, x_n)$ és $(y_1, \dots, y_n)$ is megoldja ugyanazt a HLE-t. Ekkor $(\lambda x_1, \dots, \lambda x_n)$ és $(x_1 + y_1, \dots, x_n + y_n)$ is a HLE megoldásai.

(4) Tfh $(x_1, \dots, x_n)$ és $(y_1, \dots, y_n)$ is megoldja ugyanazt a lin. egy.rsz-t. Ekkor $(x_1 - y_1, \dots, x_n - y_n)$ megoldja a megfelelő HLE-t.

(5) Lin. egy.rsz. minden egyes megoldása előáll egy rögzített megoldásából a megfelelő HLE egy megoldását hozzáadva.

Mit tanultunk ma?

Mit tanultunk ma?

- ▶ Lineáris egyenletrendszerek naiv megoldása

Mit tanultunk ma?

- ▶ Lineáris egyenletrendszerek naiv megoldása
- ▶ Kibővített együtthatómátrix, ESÁ

Mit tanultunk ma?

- ▶ Lineáris egyenletrendszerek naiv megoldása
- ▶ Kibővített együtthatómátrix, ESÁ
- ▶ (Redukált) lépcsős alak, és kapcsolata a megoldásokkal

Mit tanultunk ma?

- ▶ Lineáris egyenletrendszerek naiv megoldása
- ▶ Kibővített együtthatómátrix, ESÁ
- ▶ (Redukált) lépcsős alak, és kapcsolata a megoldásokkal
- ▶ Tilos sor, szabad paraméter, kötött változó

Mit tanultunk ma?

- ▶ Lineáris egyenletrendszerek naiv megoldása
- ▶ Kibővített együtthatómátrix, ESÁ
- ▶ (Redukált) lépcsős alak, és kapcsolata a megoldásokkal
- ▶ Tilos sor, szabad paraméter, kötött változó
- ▶ Gauss-elimináció a LA eléréséhez, RLA képzése

Mit tanultunk ma?

- ▶ Lineáris egyenletrendszerek naiv megoldása
- ▶ Kibővített együtthatómátrix, ESÁ
- ▶ (Redukált) lépcsős alak, és kapcsolata a megoldásokkal
- ▶ Tilos sor, szabad paraméter, kötött változó
- ▶ Gauss-elimináció a LA eléréséhez, RLA képzése
- ▶ Ismeretlenek, egyenletek és megoldások számának kapcsolata

Mit tanultunk ma?

- ▶ Lineáris egyenletrendszerek naiv megoldása
- ▶ Kibővített együtthatómátrix, ESÁ
- ▶ (Redukált) lépcsős alak, és kapcsolata a megoldásokkal
- ▶ Tilos sor, szabad paraméter, kötött változó
- ▶ Gauss-elimináció a LA eléréséhez, RLA képzése
- ▶ Ismeretlenek, egyenletek és megoldások számának kapcsolata
- ▶ Homogén lineáris egyenletrendszerek

Mit tanultunk ma?

- ▶ Lineáris egyenletrendszerek naiv megoldása
- ▶ Kibővített együtthatómátrix, ESÁ
- ▶ (Redukált) lépcsős alak, és kapcsolata a megoldásokkal
- ▶ Tilos sor, szabad paraméter, kötött változó
- ▶ Gauss-elimináció a LA eléréséhez, RLA képzése
- ▶ Ismeretlenek, egyenletek és megoldások számának kapcsolata
- ▶ Homogén lineáris egyenletrendszerek

Köszönöm a figyelmet!