A számítástudomány alapjai

Lineáris egyenletrendszerek

2024. október 29.

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 igazzá tesz.

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

$$3x - 4z = 666$$

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

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

Példa:

$$\begin{aligned} 3x - 4z &= 666 \\ 33x - y + 77z &= 42 \\ \pi^{e^{\pi}} y - (\ln(\cos 42)) \cdot z &= \sqrt[\sqrt[\sqrt[\pi]{4}]{2}]{42} \end{aligned}$$

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

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.

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 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.

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 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 tartalmaznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás.

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 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 tartalmaznak ismeretlent. Utóbbi egyenletek mindegyike 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 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 tartalmaznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek bármely 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 igazzá 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 tartalmaznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek bármely értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.
Példa:

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

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 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 tartalmaznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek bármely 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 igazzá 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 tartalmaznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek bármely értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.

Példa:

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

$$x=\frac{29}{7} \qquad y=-\frac{10}{7}$$

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 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 tartalmaznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek bármely 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 igazzá 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 tartalmaznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek bármely értékadása egyértelműen meghatározza a kifejezett ismeretlenek értékét.
Példa:

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

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 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 tartalmaznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind azonosság, akkor a ki nem fejezett ismeretlenek bármely 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 igazzá 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 tartalmaznak ismeretlent. Utóbbi egyenletek mindegyike vagy azonosság, vagy ellentmondás. Ha van köztük ellentmondás, akkor nincs megoldás. Ha mind 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 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:

$$\begin{aligned} x_1 - 3x_3 + 5x_4 &= -6 \\ 7x_1 + 2x_2 + 3x_3 &= 9 \\ x_2 + 7x_3 - 2x_4 &= 11 \end{aligned} \qquad \mapsto \qquad \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \left(\begin{array}{cccc|c} 1 & 0 & -3 & 5 & -6 \\ 7 & 2 & 3 & 0 & 9 \\ 0 & 1 & 7 & -2 & 11 \end{array} \right) \end{array}$$

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:

$$\begin{aligned} x_1 - 3x_3 + 5x_4 &= -6 \\ 7x_1 + 2x_2 + 3x_3 &= 9 \\ x_2 + 7x_3 - 2x_4 &= 11 \end{aligned} \qquad \mapsto \qquad \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \left(\begin{array}{cccc|c} 1 & 0 & -3 & 5 & -6 \\ 7 & 2 & 3 & 0 & 9 \\ 0 & 1 & 7 & -2 & 11 \end{array} \right) \end{array}$$

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:

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

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, amelyek során a megoldások halmaza nem változik. Konkrétan: egyenleteket felcserélünk, egyenletet nemnullával végigszorzunk ill. az i-dik egyenletet kicseréljük az i-dik és j-dik egyenletek összegére. Erről szól a következő definíció.

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.

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 (az i-dik sor helyettesítése az i-dik sor és a j-dik sor konstansszorosának összegével, csupa0 sor hozzáadása/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 végigszorzása, (3) az i-dik sor helyettesítése az i-dik és j-dik sorok (koordinátánkénti) összegével (az i-dik sor helyettesítése az i-dik sor és a j-dik sor konstansszorosának összegével, csupa0 sor hozzáadása/elhagyása).

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 (az i-dik sor helyettesítése az i-dik sor és a j-dik sor konstansszorosának összegével, csupa0 sor hozzáadása/elhagyása).
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 (az i-dik sor helyettesítése az i-dik sor és a j-dik sor konstansszorosának összegével, csupa0 sor hozzáadása/elhagyása).
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 (az i-dik sor helyettesítése az i-dik sor és a j-dik sor konstansszorosának összegével, csupa0 sor hozzáadása/elhagyása).

Állítás: ESÁ hatására a megoldások halmaza nem változik.

Cél: Tetszőleges lineáris egyenletrendszerből kiindulva, ESÁ-okkal elérni, hogy a kapott rendszer a megoldások leírását adja.

Először leírjuk azt a kibővített együtthatómátrix-tulajdonságot, amit elérve minden megoldás könnyen kiolvashatóvá válik.

(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.

(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.

(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

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

(Redukált) lépcsős alak

Def: Az M mátrix lépcsős alakú (LA), ha
(1) minden sor első nemnulla eleme 1-es (ú.n. vezér1-es, avagy 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:

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

(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:

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

(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:

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

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

Példa: RLA kibővített egyhómx:

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

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

(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:

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

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

(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:

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

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

(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:

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

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

Def: Kib. egyhómx tilos sora: $\begin{array}{ccc|c} 0 & \dots & 0 & x \end{array}$ 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:

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

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

Def: Kib. egyhómx tilos sora: $\left.\begin{array}{ccc} 0 & \ldots & 0 \end{array}\right| x$ 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:

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

Def: Kib. egyhómx tilos sora: $\begin{array}{ccc|c} 0 & \dots & 0 & x \end{array}$ 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:

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

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.
(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:

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

Def: Kib. egyhómx tilos sora: $\begin{array}{ccc|c} 0 & \ldots & 0 & x \end{array}$ 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 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: $\begin{array}{ccc|c} 0 & \ldots & 0 & x \end{array}$ 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 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: $\begin{array}{ccc|c} 0 & \dots & 0 & x \end{array}$ 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 azt jelenti, hogy a lin. egyenletrsz. 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: $\begin{array}{ccc|c} 0 & \dots & 0 & x \end{array}$ 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 azt jenti, hogy a lin. egyenletrsz. 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 nemulla 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 nemulla 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 nemulla 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 nemulla 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 nemulla 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 nemulla 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:

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

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

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 nemulla 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 nemulla 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 nemulla 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 nemulla 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 nemulla 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 nemulla 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 nemulla 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 garantáltan működik, bár néha sokat kell számolni. 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 garantáltan működik, bár néha sokat kell számolni. 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 garantáltan működik, bár néha sokat kell számolni. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

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

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

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami garantáltan működik, bár néha sokat kell számolni. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \underset{\boxed{\text{II}} \mapsto \boxed{\text{II}} - 2\cdot \boxed{\text{I}}}{\begin{pmatrix} 1 & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ 0 & \frac{9}{13} & -\frac{53}{13} & \frac{21}{13} \end{pmatrix}}$$

Yeah.

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami garantáltan működik, bár néha sokat kell számolni. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

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

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami garantáltan működik, bár néha sokat kell számolni. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

$$\begin{pmatrix} 13 & 2 & 7 & -4 \\ \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ \\ 2 & 1 & -3 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & \frac{2}{13} & \frac{7}{13} & -\frac{4}{13} \\ \\ 0 & \frac{9}{13} & -\frac{53}{13} & \frac{21}{13} \end{pmatrix}$$

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 garantáltan működik, bár néha sokat kell számolni. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

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

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

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

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami garantáltan működik, bár néha sokat kell számolni. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.

Példa: Mechanikus Gauss-elimináció vs agysejtkímélő eljárás

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

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

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

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami garantáltan működik, bár néha sokat kell számolni. 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 garantáltan működik, bár néha sokat kell számolni. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.
(2) A Gauss-elimináció megvalósítható rekurzív algoritmusként is.

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami garantáltan működik, bár néha sokat kell számolni. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.
(2) A Gauss-elimináció megvalósítható rekurzív algoritmusként is.
A GE(M) (az M mátrixot lépcsős alakra hozó eljárás):
$\boxed{\textbf{1.}}$ 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 garantáltan működik, bár néha sokat kell számolni. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.
(2) A Gauss-elimináció megvalósítható rekurzív algoritmusként is. A GE(M) (az M mátrixot lépcsős alakra hozó eljárás):
$\boxed{\textbf{1.}}$ Ha M első oszlopa csupa0, akkor M' az első oszlop törlésével keletkező mátrix. Output: GE(M') elé írunk egy csupa0 oszlopot.
$\boxed{\textbf{2.}}$ Ha M első oszlopa tartalmaz nemnulla elemet, sorcserével és az első sor konstanssal szorzásával az első sor első elemét v1-sé tesszük, majd a v1 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 garantáltan működik, bár néha sokat kell számolni. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.
(2) A Gauss-elimináció megvalósítható rekurzív algoritmusként is.

A Gauss-elimináció további tulajdonságai

Megf: (1) A Gauss-elimináció olyan eljárás, ami garantáltan működik, bár néha sokat kell számolni. A LA vagy RLA eléréséhez nem muszáj a Gauss-eliminációt követni: dolgozhatunk ügyes ESÁ-okkal.
(2) A Gauss-elimináció megvalósítható 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 $konst \cdot nk$ lépést igényel, az összlépésszám legfeljebb $konst \cdot n^2k$. 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 v1, 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 v1, akkor nincs megoldás
   - Ha az utolsó kivételével minden oszlopban van v1, 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 v1, akkor nincs megoldás
   - Ha az utolsó kivételével minden oszlopban van v1, akkor egyetlen megoldás van
   - Ha az utolsón kívül más oszlopban sincs v1, 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 v1, akkor nincs megoldás
   - Ha az utolsó kivételével minden oszlopban van v1, akkor egyetlen megoldás van
   - Ha az utolsón kívül más oszlopban sincs v1, 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 v1, akkor nincs megoldás
    - Ha az utolsó kivételével minden oszlopban van v1, akkor egyetlen megoldás van
    - Ha az utolsón kívül más oszlopban sincs v1, 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 v1. Ezért a kibővített együtthatómátrixnak legalább annyi sora van, mint a változók száma. $\square$

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 v1, akkor nincs megoldás
    - Ha az utolsó kivételével minden oszlopban van v1, akkor egyetlen megoldás van
    - Ha az utolsón kívül más oszlopban sincs v1, 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 v1. 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.

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

Köszönöm a figyelmet!