Matematika A2, VBK
8. előadás

Takács Balázs

Budapest, 2026. március 12.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel. Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel.Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  - a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  - b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  - c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel.Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  - a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  - b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  - c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel.Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel.Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel.Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel.Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  - a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  - b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  - c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel.Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  - a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  - b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  - c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel.Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  - a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  - b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  - c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel.Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  - a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  - b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  - c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel.Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/1

- Feltehető (szükség esetén sorok cseréjével), hogy $a_{11} \neq 0$. Ezt az elemet bekeretezzük és az első sort leosztjuk a_{11}-gyel.Ekkor a mátrix bal felső sarkában álló bekeretezett elem 1 lesz, neve: **vezéregyes**.
- A bekeretezett vezéregyes alatti elemeket kinullázzuk: az adott sorhoz az első sor megfelelő számszorosát adjuk.
- A következő lépésekben egy vezéregyes alatti nullázás elvégzése után a jelölőnégyzetet eggyel lefelé majd jobbra mozgatjuk. Ha a négyzet kilép az együtthatómátrixból, akkor az algoritmus véget ér. Egyébként pedig az alábbi esetek fordulhatnak elő:
  - a) a bekeretezett elem nem nulla: ezzel az elemmel osztjuk az adott sort, majd a keletkezett vezéregyes alatti elemeket kinullázzuk a fent leírt módon.
  - b) a bekeretezett elem 0 és alatta nem minden elem nulla: ekkor a legkisebb sorszámú ilyen sorral cserélünk, osztjuk az adott sort, majd az új vezéregyes alatt nullázunk.
  - c) a bekeretezett elem 0 és az alatta lévő elemek is 0-k: a jelölőnégyzetet eggyel jobbra mozgatjuk.
- Ha az algoritmus bármely lépése során csupa 0 sor keletkezik, akkor ezt elhagyjuk.
- Ha az algoritmus bármely lépése során keletkezik olyan sor, amelynek a vonaltól balra lévő elemei mind 0-k, a vonaltól jobba lévő eleme pedig nem nulla, akkor az algoritmus leáll. Az ilyen sor neve: **tilos sor**.

Gauss-elimináció/2

Az előbbi lépéseket elvégezve a kibővített együtthatómátrixot **lépcsős alakra** hoztuk, ami annyit jelent, hogy a vezéregyesek alatt 0-k állnak.
Az algoritmus tovább folytatható a következőképpen: a vezéregyeseken jobbról balra haladva a felettük álló elemeket is kinullázzuk. Ekkor azt mondjuk, hogy a kibővített együtthatómátrixot **redukált lépcsős alakra** hoztuk.
Az algoritmusnak tehát kétféle kimenetele lehet:

- A futás során keletkezik tilos sor: az egyenletrendszernek nincs megoldása, ugyanis egy tilos sor a $0x_1 + 0x_2 + \dots 0x_n = b$ egyenlőségnek felel meg (ahol $b \neq 0$), amelynek nincs megoldása.
- Az algoritmus futása során nem keletkezik tilos sor. Ekkor két további aleset lehetséges:
  a) Az együtthatómátrix minden oszlopa tartalmaz vezéregyest: az egyenletrendszer megoldása egyértelmű, és a redukált lépcsős alakból azonnal leolvasható.
  b) Az együtthatómátrixnak van olyan oszlopa, amely nem tartalmaz vezéregyest: ezen oszlopkhoz tartozó változókat szabad paraméternek tekintve az egyenletrendszer megoldásai a redukált lépcsős alakból átrendezés után leolvashatók. Ekkor az egyenletrendszernek végtelen sok megoldása van.

Gauss-elimináció/2

Az előbbi lépéseket elvégezve a kibővített együtthatómátrixot **lépcsős alakra** hoztuk, ami annyit jelent, hogy a vezéregyesek alatt 0-k állnak. Az algoritmus tovább folytatható a következőképpen: a vezéregyeseken jobbról balra haladva a felettük álló elemeket is kinullázzuk. Ekkor azt mondjuk, hogy a kibővített együtthatómátrixot **redukált lépcsős alakra** hoztuk.

Az algoritmusnak tehát kétféle kimenetele lehet:

- A futás során keletkezik tilos sor: az egyenletrendszernek nincs megoldása, ugyanis egy tilos sor a $0x_1 + 0x_2 + \dots 0x_n = b$ egyenlőségnek felel meg (ahol $b \neq 0$), amelynek nincs megoldása.
- Az algoritmus futása során nem keletkezik tilos sor. Ekkor két további aleset lehetséges:
    a) Az együtthatómátrix minden oszlopa tartalmaz vezéregyest: az egyenletrendszer megoldása egyértelmű, és a redukált lépcsős alakból azonnal leolvasható.
    b) Az együtthatómátrixnak van olyan oszlopa, amely nem tartalmaz vezéregyest: ezen oszlopkhoz tartozó változókat szabad paraméternek tekintve az egyenletrendszer megoldásai a redukált lépcsős alakból átrendezés után leolvashatók. Ekkor az egyenletrendszernek végtelen sok megoldása van.

Gauss-elimináció/2

Az előbbi lépéseket elvégezve a kibővített együtthatómátrixot **lépcsős alakra** hoztuk, ami annyit jelent, hogy a vezéregyesek alatt 0-k állnak. Az algoritmus tovább folytatható a következőképpen: a vezéregyeseken jobbról balra haladva a felettük álló elemeket is kinullázzuk. Ekkor azt mondjuk, hogy a kibővített együtthatómátrixot **redukált lépcsős alakra** hoztuk.

Az algoritmusnak tehát kétféle kimenetele lehet:

- A futás során keletkezik tilos sor: az egyenletrendszernek nincs megoldása, ugyanis egy tilos sor a $0x_1 + 0x_2 + \ldots 0x_n = b$ egyenlőségnek felel meg (ahol $b \neq 0$), amelynek nincs megoldása.
- Az algoritmus futása során nem keletkezik tilos sor. Ekkor két további aleset lehetséges:
  - a) Az együtthatómátrix minden oszlopa tartalmaz vezéregyest: az egyenletrendszer megoldása egyértelmű, és a redukált lépcsős alakból azonnal leolvasható.
  - b) Az együtthatómátrixnak van olyan oszlopa, amely nem tartalmaz vezéregyest: ezen oszlopkhoz tartozó változókat szabad paraméternek tekintve az egyenletrendszer megoldásai a redukált lépcsős alakból átrendezés után leolvashatók. Ekkor az egyenletrendszernek végtelen sok megoldása van.

Gauss-elimináció/2

Az előbbi lépéseket elvégezve a kibővített együtthatómátrixot **lépcsős alakra** hoztuk, ami annyit jelent, hogy a vezéregyesek alatt 0-k állnak. Az algoritmus tovább folytatható a következőképpen: a vezéregyeseken jobbról balra haladva a felettük álló elemeket is kinullázzuk. Ekkor azt mondjuk, hogy a kibővített együtthatómátrixot **redukált lépcsős alakra** hoztuk.

Az algoritmusnak tehát kétféle kimenetele lehet:

- A futás során keletkezik tilos sor: az egyenletrendszernek nincs megoldása, ugyanis egy tilos sor a $0x_1 + 0x_2 + \dots 0x_n = b$ egyenlőségnek felel meg (ahol $b \neq 0$), amelynek nincs megoldása.
- Az algoritmus futása során nem keletkezik tilos sor. Ekkor két további aleset lehetséges:
  - a) Az együtthatómátrix minden oszlopa tartalmaz vezéregyest: az egyenletrendszer megoldása egyértelmű, és a redukált lépcsős alakból azonnal leolvasható.
  - b) Az együtthatómátrixnak van olyan oszlopa, amely nem tartalmaz vezéregyest: ezen oszlopkhoz tartozó változókat szabad paraméternek tekintve az egyenletrendszer megoldásai a redukált lépcsős alakból átrendezés után leolvashatók. Ekkor az egyenletrendszernek végtelen sok megoldása van.

Gauss-elimináció/2

Az előbbi lépéseket elvégezve a kibővített együtthatómátrixot **lépcsős alakra** hoztuk, ami annyit jelent, hogy a vezéregyesek alatt 0-k állnak. Az algoritmus tovább folytatható a következőképpen: a vezéregyeseken jobbról balra haladva a felettük álló elemeket is kinullázzuk. Ekkor azt mondjuk, hogy a kibővített együtthatómátrixot **redukált lépcsős alakra** hoztuk.
Az algoritmusnak tehát kétféle kimenetele lehet:

- A futás során keletkezik tilos sor: az egyenletrendszernek nincs megoldása, ugyanis egy tilos sor a $0x_1 + 0x_2 + \dots 0x_n = b$ egyenlőségnek felel meg (ahol $b \neq 0$), amelynek nincs megoldása.
- Az algoritmus futása során nem keletkezik tilos sor. Ekkor két további aleset lehetséges:
  - a) Az együtthatómátrix minden oszlopa tartalmaz vezéregyest: az egyenletrendszer megoldása egyértelmű, és a redukált lépcsős alakból azonnal leolvasható.
  - b) Az együtthatómátrixnak van olyan oszlopa, amely nem tartalmaz vezéregyest: ezen oszlopkhoz tartozó változókat szabad paraméternek tekintve az egyenletrendszer megoldásai a redukált lépcsős alakból átrendezés után leolvashatók. Ekkor az egyenletrendszernek végtelen sok megoldása van.

Gauss-elimináció/2

Az előbbi lépéseket elvégezve a kibővített együtthatómátrixot **lépcsős alakra** hoztuk, ami annyit jelent, hogy a vezéregyesek alatt 0-k állnak. Az algoritmus tovább folytatható a következőképpen: a vezéregyeseken jobbról balra haladva a felettük álló elemeket is kinullázzuk. Ekkor azt mondjuk, hogy a kibővített együtthatómátrixot **redukált lépcsős alakra** hoztuk.

Az algoritmusnak tehát kétféle kimenetele lehet:

- A futás során keletkezik tilos sor: az egyenletrendszernek nincs megoldása, ugyanis egy tilos sor a $0x_1 + 0x_2 + \ldots 0x_n = b$ egyenlőségnek felel meg (ahol $b \neq 0$), amelynek nincs megoldása.
- Az algoritmus futása során nem keletkezik tilos sor. Ekkor két további aleset lehetséges:
    a) Az együtthatómátrix minden oszlopa tartalmaz vezéregyest: az egyenletrendszer megoldása egyértelmű, és a redukált lépcsős alakból azonnal leolvasható.
    b) Az együtthatómátrixnak van olyan oszlopa, amely nem tartalmaz vezéregyest: ezen oszlopkhoz tartozó változókat szabad paraméternek tekintve az egyenletrendszer megoldásai a redukált lépcsős alakból átrendezés után leolvashatók. Ekkor az egyenletrendszernek végtelen sok megoldása van.

Gauss-elimináció/2

Az előbbi lépéseket elvégezve a kibővített együtthatómátrixot **lépcsős alakra** hoztuk, ami annyit jelent, hogy a vezéregyesek alatt 0-k állnak.
Az algoritmus tovább folytatható a következőképpen: a vezéregyeseken jobbról balra haladva a felettük álló elemeket is kinullázzuk. Ekkor azt mondjuk, hogy a kibővített együtthatómátrixot **redukált lépcsős alakra** hoztuk.
Az algoritmusnak tehát kétféle kimenetele lehet:

- A futás során keletkezik tilos sor: az egyenletrendszernek nincs megoldása, ugyanis egy tilos sor a $0x_1 + 0x_2 + \dots 0x_n = b$ egyenlőségnek felel meg (ahol $b \neq 0$), amelynek nincs megoldása.
- Az algoritmus futása során nem keletkezik tilos sor. Ekkor két további aleset lehetséges:
  a) Az együtthatómátrix minden oszlopa tartalmaz vezéregyest: az egyenletrendszer megoldása egyértelmű, és a redukált lépcsős alakból azonnal leolvasható.
  b) Az együtthatómátrixnak van olyan oszlopa, amely nem tartalmaz vezéregyest: ezen oszlopkhoz tartozó változókat szabad paraméternek tekintve az egyenletrendszer megoldásai a redukált lépcsős alakból átrendezés után leolvashatók. Ekkor az egyenletrendszernek végtelen sok megoldása van.

Gauss-elimináció/2

Az előbbi lépéseket elvégezve a kibővített együtthatómátrixot **lépcsős alakra** hoztuk, ami annyit jelent, hogy a vezéregyesek alatt 0-k állnak.
Az algoritmus tovább folytatható a következőképpen: a vezéregyeseken jobbról balra haladva a felettük álló elemeket is kinullázzuk. Ekkor azt mondjuk, hogy a kibővített együtthatómátrixot **redukált lépcsős alakra** hoztuk.
Az algoritmusnak tehát kétféle kimenetele lehet:

- A futás során keletkezik tilos sor: az egyenletrendszernek nincs megoldása, ugyanis egy tilos sor a $0x_1 + 0x_2 + \dots 0x_n = b$ egyenlőségnek felel meg (ahol $b \neq 0$), amelynek nincs megoldása.
- Az algoritmus futása során nem keletkezik tilos sor. Ekkor két további aleset lehetséges:
  a) Az együtthatómátrix minden oszlopa tartalmaz vezéregyest: az egyenletrendszer megoldása egyértelmű, és a redukált lépcsős alakból azonnal leolvasható.
  b) Az együtthatómátrixnak van olyan oszlopa, amely nem tartalmaz vezéregyest: ezen oszlopkhoz tartozó változókat szabad paraméternek tekintve az egyenletrendszer megoldásai a redukált lépcsős alakból átrendezés után leolvashatók. Ekkor az egyenletrendszernek végtelen sok megoldása van.

Emlékeztető

Tétel (A determináns tulajdonságai)

Legyen $A := (a_{i,j})_{i,j=1}^{n} \in \mathbb{R}^{n\times n}$*. Ekkor az alábbi állítások teljesülnek.*

- *Ha A alsó- vagy felső háromszögmátrix, akkor*

  $$\det(A) = a_{1,1} \cdot a_{2,2} \cdot \ldots \cdot a_{n,n},$$

  azaz a determináns a főátlóbeli elemek szorzata
- *Ha A egy tetszőleges sorához egy másik sor számszorosát adjuk, akkor a kapott* $\widetilde{A}$ *mátrix determinánsa megegyezik az eredeti mátrix determinánsával:* $\det(\widetilde{A}) = \det(A)$
- *Az A mátrix egy tetszőleges sorát a* λ *valós számmal szorozva, a kapott* $\tilde{A}$ *mátrix determinánsa megegyezik az eredeti mátrix determinánsának* λ*-szorosával:* $\det(\widetilde{A}) = \lambda \det(A)$.
- *Az A mátrix két sorának felcserélésével kapott* $\widetilde{A}$ *mátrix determinánsa megegyezik az eredeti mátrix determinánsának ellentettjével:* $\det(\widetilde{A}) = -\det(A)$.