3. Rješavanje linearnih sustava

3.1. Koji se sustavi rješavaju u praksi

Zadani su matrica $A \in \mathbb{C}^{m\times n}$ i vektor $b \in \mathbb{C}^m$. Teorem Kronecker–Capelli daje odgovor na pitanje kad linearni sustav

$$Ax = b \tag{3.1.1}$$

ima rješenje $x \in \mathbb{C}^n$ i kad je ono jedinstveno.

U praksi se najčešće rješavaju linearni sustavi kad je matrica sustava A kvadratna i nesingularna. Tada znamo da sustav ima jedinstveno rješenje.

Kako bi se moglo izračunati rješenje takvog sustava? Na primjer, mogao bi se izračunati inverz A^{-1}, pa množenjem relacije (3.1.1) slijeva s A^{-1} dobivamo

$$x = A^{-1}b.$$

Time nam je odmah dana i jedna metoda za rješavanje linearnog sustava (3.1.1) koja je sasvim nepraktična, jer smo rješavanje linearnog sustava preveli u računanje inverza, što je teži problem. Naime, j-ti stupac inverza je rješenje sustava $Ax = e_j$.

Druga metoda, koja se često spominje u linearnoj algebri je Cramerovo pravilo. Prisjetimo se, j-ta komponenta rješenja sustava je

$$x_j = \frac{\det A_j}{\det A},$$

pri čemu je matrica A_j jednaka matrici A, osim što je j-ti stupac u A_j zamijenjen desnom stranom b. Složenost ovog načina rješavanja je eksponencijalna (dokažite to!) i nikad se ne koristi kao metoda numeričkog rješavanja.

Najjednostavnija metoda za rješavanje linearnog sustava (3.1.1) je njegovo svođenje na trokutastu formu $Rx = y$, gdje je R trokutasta matrica (recimo, gornja), iz koje se lako, tzv. povratnom supstitucijom, nalazi rješenje.

Gaussove eliminacije su metoda direktnog transformiranja linearnog sustava $Ax = b$, zajedno s desnom stranom, na trokutastu formu. Gaussove eliminacije možemo implementirati i tako da se desna strana ne transformira istovremeno kad i matrica A. Tada se formiraju dvije matrice L i R (R je trokutasta matrica iz Gaussovih eliminacija) i koristeći njih lako se dobije rješenje traženog sustava. Kad se Gaussove eliminacije tako implementiraju, metoda se obično zove LR (neki to zovu LU) faktorizacija matrice A. Ovaj pristup je posebno zgodan kad imamo više linearnih sustava s istom matricom A, a desne strane se razlikuju.

Mnogi sustavi linearnih jednadžbi imaju specijalna svojstva i specijalnu strukturu. U nekim od tih slučajeva, koji su za praksu jako bitni, mogu se primijeniti i iterativne metode, koje daju rješenje linearnog sustava (3.1.1) na zadovoljavajuću točnost mnogo brže nego LR faktorizacija.

3.2. Gaussove eliminacije

Elementarne transformacije su one koje ne mijenjaju rješenje linearnog sustava. Takve transformacije su: množenje jednadžbe konstantom različitom od 0, dodavanje jednadžbe (ili linearne kombinacije jednadžbi) drugim jednadžbama i zamjena poretka jednadžbi. Korištenjem elementarnih transformacija, svaki se linearni sustav s kvadratnom nesingularnom matricom može svesti na trokutasti oblik.

Označimo $A^{(1)} := A$, $b^{(1)} := b$. U skraćenoj notaciji, bez pisanja nepoznanica x_i, linearni sustav (3.1.1) možemo zapisati proširenom matricom, kao

$$\left[\begin{array}{cccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right].$$

Počnimo sa svođenjem matrice na trokutastu formu. Za prvi stupac to znači da u tom stupcu moramo poništiti sve elemente, osim prvog. Ako je element $a_{11}^{(1)} \neq 0$, onda redom, možemo od i-te jednadžbe oduzeti prvu jednadžbu pomnoženu s

$$m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, \quad i = 2, \ldots, n.$$

Prva jednadžba se ne mijenja. Time smo dobili ekvivalentni linearni sustav

$$\left[\begin{array}{cccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array}\right].$$

Postupak poništavanja možemo nastaviti s drugim stupcem matrice $A^{(2)}$, od dijagonale nadolje. Ako je $a_{22}^{(2)} \neq 0$, biramo faktore

$$m_{i2} = \frac{a_{i2}^{(2)}}{a_{22}^{(2)}}, \quad i = 3, \ldots, n,$$

tako da poništimo sve elemente drugog stupca ispod dijagonale. I tako redom. Konačno, ako su svi $a_{ii}^{(i)} \neq 0$, za $i = 1, \ldots, n-1$, završni linearni sustav, ekvivalentan polaznom, je

$$\left[\begin{array}{cccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ & & \ddots & \vdots & \vdots \\ & & & a_{nn}^{(n)} & b_n^{(n)} \end{array}\right].$$

Uz pretpostavku da je $a_{nn}^{(n)} \neq 0$, ovaj se linearni sustav lako rješava povratnom supstitucijom

$$\begin{aligned} x_n &= \frac{b_n^{(n)}}{a_{nn}^{(n)}}, \\ x_i &= \frac{1}{a_{ii}^{(i)}} \Big(b_i^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)} x_j \Big), \quad i = n-1, \ldots, 1. \end{aligned}$$

Prvo pitanje koje se nameće je moraju li svi $a_{ii}^{(i)}$ biti različiti od nule, ako je A regularna i kvadratna. Jasno je da ne moraju. Na primjer, matrica linearnog sustava

$$\left[\begin{array}{cc:c} 0 & 1 & 1 \\ 1 & 0 & 1 \end{array}\right]$$

je regularna ($\det A = -1$), sustav ima jedinstveno rješenje $x_1 = x_2 = 1$, a ipak ga ne možemo riješiti Gaussovim eliminacijama ako ne mijenjamo poredak jednadžbi.

Zamjena bilo koje dvije jednadžbe neće promijeniti rješenje sustava. Dakle, ako je $a_{11} = 0$, prije eliminacije elemenata prvog stupca, moramo izabrati ne-nula element u prvom stupcu, zovimo ga a_{r1}, a zatim zamijeniti prvu i r-tu jednadžbu.

Ponovno, nismo sigurni je li to uvijek moguće. No, ako u prvom stupcu ne postoji ne-nula element, matrica A ima nul-stupac za prvi stupac, pa ne može biti regularna. Pokažite da isti argument vrijedi za svaku od matrica u procesu eliminacije, tj. ako su u k-tom koraku eliminacije svi elementi matrice $A^{(k)}$ na ili ispod glavne dijagonale u k-tom stupcu jednaki 0, onda je matrica A singularna.

Dakle, ako je A nesingularna, onda u svakom koraku k ($k = 1, \ldots, n-1$) eliminacije, u matrici $A^{(k)}$ možemo naći element $a_{rk}^{(k)} \neq 0$, uz $r \geq k$, kojeg zamjenom jednadžbi r i k dovodimo na dijagonalu, tako da je $a_{kk}^{(k)} \neq 0$, a zatim računamo matricu $A^{(k+1)}$. Takve ne-nula elemente koje dovodimo na dijagonalu zovemo pivotnim elementima.

Drugo je pitanje, da li je dovoljno dobro birati pivotne elemente samo tako da budu ne-nula, što je, u principu, dovoljno da provedemo postupak eliminacije. Primjer 2.7.4 pokazuje da biranje veličine pivotnog elementa nije beznačajno.

Uobičajeno **parcijalno pivotiranje** kao pivotni element bira element koji je po apsolutnoj vrijednosti najveći u ostatku tog stupca — na glavnoj dijagonali ili ispod nje. Drugim riječima, ako je u k-tom koraku

$$|a_{rk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|,$$

onda ćemo zamijeniti r-ti i k-ti redak i početi korak eliminacije elemenata k-tog stupca.

Motivacija za takvo biranje pivotnih elemenata je jednostavna. Elementi "ostatka" linearnog sustava koje treba izračunati u matrici $A^{(k+1)}$ u k-tom koraku transformacije su

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \quad b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)}, \tag{3.2.1}$$

za $i, j = k+1, \ldots, n$, a multiplikatori m_{ik} su

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad i = k+1, \ldots, n. \tag{3.2.2}$$

Ako je multiplikator m_{ik} velik, u aritmetici pomičnog zareza može doći do kraćenja najmanje značajnih znamenki $a_{ij}^{(k)}$, tako da izračunati $a_{ij}^{(k+1)}$ može imati veliku relativnu grešku. Nažalost, to kraćenje može biti ekvivalentno relativno velikoj perturbaciji u originalnoj matrici A.

Na primjer, neka je

$$A = \begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix}, \quad \varepsilon \leq u.$$

Eliminacija elementa a_{21} u aritmetici pomičnog zareza, umjesto elementa $a_{22}^{(2)}$, daje

$$f\ell(a_{22}^{(2)}) = f\ell\left(1 - \frac{1}{\varepsilon}\right) = -\frac{1}{\varepsilon}, \tag{3.2.3}$$

zbog $1 \ll 1/\varepsilon$. Kad bismo u originalnoj matrici A promijenili a_{22} s 1 u 0, dobili bismo isti rezultat za $f\ell(a_{22}^{(2)})$, s tim da je on sad i egzaktan. Drugim riječima, greška zaokruživanja napravljena u (3.2.3) ekvivalentna je velikoj relativnoj perturbaciji u originalnoj matrici A. Pogledajte da li bi se isto dogodilo da smo zamijenili jednadžbe prije početka eliminacije.

Sasvim općenito, ideja pivotiranja je minimizirati korekcije elemenata u (3.2.1) pri prijelazu s $A^{(k)}$ na $A^{(k+1)}$. Dakle, multiplikatori u (3.2.2) trebaju biti što manji. To se postiže izborom što je moguće većeg nazivnika (po apsolutnoj vrijednosti), a to je upravo pivotni element. Primijetite da za multiplikatore kod parcijalnog pivotiranja vrijedi

$$|m_{ik}| \leq 1, \quad i = k+1, \ldots, n.$$

U praksi, parcijalno pivotiranje funkcionira izvrsno, ali matematičari su konstruirali primjere kad ono "nije savršeno". Što to točno znači, bit će rečeno u jednom od sljedećih poglavlja.

Osim parcijalnog pivotiranja, može se provoditi i **potpuno pivotiranje**. U k-tom koraku, bira se maksimalni element u cijelom "ostatku" matrice $A^{(k)}$, a ne samo u k-tom stupcu. Ako je u k-tom koraku

$$|a_{rs}^{(k)}| = \max_{k \leq i,j \leq n} |a_{ij}^{(k)}|,$$

onda ćemo zamijeniti r-ti i k-ti redak, s-ti i k-ti stupac i početi korak eliminacije elemenata k-tog stupca. Ipak, trebamo biti malo oprezni. Zamjenom s-tog i k-tog stupca zamijenili smo ulogu varijabli x_s i x_k. Sve takve promjene treba pamtiti u vektoru permutacije varijabli. Osim toga, u usporedbi s parcijalnim pivotiranjem, imamo mnogo više pretraživanja u svakom koraku ($(n-k+1)^2$ elemenata, prema ranijih $n-k+1$), što usporava proces. Ipak, korištenjem potpunog pivotiranja mogu se izvesti bolje ocjene greške nego kod parcijalnog pivotiranja.

Ovo nisu jedine mogućnosti pivotiranja kod rješavanja linearnih sustava. Rutishauser je početkom sedamdesetih godina opisao relativno parcijalno pivotiranje, ali algoritam nije ušao u široku upotrebu.

Napišimo sad algoritam koji korištenjem Gaussovih eliminacija rješava linearni sustav $Ax = b$. Sve transformacije provodimo u istim poljima A i b koja na početku sadrže ulazne podatke.

Algoritam 3.2.1. (Gaussove eliminacije s parcijalnim pivotiranjem)

```
  {Trokutasta redukcija}
for k := 1 to n − 1 do
  begin
    {Nađi maksimalni element u ostatku stupca}
  max_elt := 0.0;
  ind_max := k;
  for i := k to n do
    if abs(A[i, k]) > max_elt then
      begin
      max_elt := abs(A[i, k]);
```

```
        ind_max := i;
        end;
  if max_elt > 0.0 then
    begin
    if ind_max <> k then
        {Zamijeni k-ti i ind_max-ti redak}
      begin
      for j := k to n do
        begin
        temp := A[ind_max, j];
        A[ind_max, j] := A[k, j];
        A[k, j] := temp;
        end;
      temp := b[ind_max];
      b[ind_max] := b[k];
      b[k] := temp;
      end;
    for i := k + 1 to n do
      begin
      mult := A[i, k]/A[k, k];
      A[i, k] := 0.0;   {Ne treba, ne koristi se kasnije}
      for j := k + 1 to n do
        A[i, j] := A[i, j] − mult ∗ A[k, j];
      b[i] := b[i] − mult ∗ b[k];
      end;
    end
  else
      {Matrica je singularna, stani s algoritmom}
    begin
    error := true;
    exit;
    end;
  end;
  {Povratna supstitucija, rješenje x ostavi u b}
b[n] := b[n]/A[n, n];
for i := n − 1 downto 1 do
  begin
  sum := b[i];
  for j := i + 1 to n do
    sum := sum − A[i, j] ∗ b[j];
  b[i] := sum/A[i, i];
  end;
error := false;
```

Zadatak 3.2.1. *Pokušajte samostalno napisati algoritam koji koristi potpuno pivotiranje. Posebnu pažnju obratite na efikasno pamćenje zamjena varijabli koje su posljedica zamjena stupaca. Može li se isti princip efikasno primijeniti i za pamćenje zamjena redaka, tako da se potpuno izbjegnu eksplicitne zamjene elemenata u matrici A i vektoru b?*

Prebrojimo sve aritmetičke operacije ovog algoritma da bismo dobili jednostavnu mjeru složenosti Gaussovih eliminacija. U prvom koraku trokutaste redukcije obavlja se:

- $n-1$ dijeljenje — računanje *mult*,
- $n(n-1)$ množenje — za svaki od $n-1$ redaka po $n-1$ množenje za računanje elemenata matrice A i jedno množenje za računanje elementa vektora b,
- $n(n-1)$ oduzimanje — javlja se u istoj naredbi gdje i prethodna množenja.

Na sličan način zaključujemo da se u k-tom koraku obavlja:

- $n-k$ dijeljenja,
- $(n-k+1)\,(n-k)$ množenja i $(n-k+1)\,(n-k)$ oduzimanja.

Ukupno, u k-tom koraku imamo

$$n-k+2(n-k+1)\,(n-k) = 2(n-k)^2 + 3(n-k)$$

aritmetičkih operacija.

Broj koraka k varira od 1 do $n-1$, pa je ukupan broj operacija potrebnih za svođenje na trokutastu formu jednak

$$\sum_{k=1}^{n-1} [2(n-k)^2 + 3(n-k)] = \sum_{k=1}^{n-1} (2k^2 + 3k) = \frac{1}{6}(4n^3 + 3n^2 - 7n).$$

Druga suma u prošloj jednakosti dobije se iz prve zamjenom indeksa $n-k \to k$.

Potpuno istim zaključivanjem dobivamo da u povratnoj supstituciji ima:

- $(n-1)\,n/2$ množenja i $(n-1)\,n/2$ zbrajanja,
- n dijeljenja,

što je zajedno točno n^2 operacija.

Dakle, ukupan broj operacija u Gaussovim eliminacijama je

$$OP(n) = \frac{1}{6}(4n^3 + 9n^2 - 7n),$$

što je približno $2n^3/3$, za malo veće n.

Ovaj broj je najjednostavnija mjera efikasnosti ili složenosti Gaussovih eliminacija. Uočimo da ova mjera ignorira pivotiranje, jer tamo nema vidljivih aritmetičkih operacija. Međutim, uspoređivanje dva realna broja u floating point aritmetici se obično radi oduzimanjem ta dva broja i usporedbom rezultata s nulom. U tom smislu, sve takve usporedbe bi, također, trebalo brojati. Nađite njihov broj za parcijalno i potpuno pivotiranje.

Ako se u Gaussovim eliminacijama poništavaju ne samo elementi ispod dijagonale, nego i iznad nje, dobivamo tzv. Gauss–Jordanovu metodu, koja linearni sustav svodi na ekvivalentni dijagonalni sustav. Gauss–Jordanove eliminacije se danas rijetko koriste u praksi, jer zahtijevaju previše računskih operacija.

Zadatak 3.2.2. *Napišite taj algoritam i pokažite da je broj računskih operacija, ne brojeći uspoređivanja, u tom slučaju jednak*

$$OP(n) = n^3 + n^2 - n.$$

To je skoro 50% *više računskih operacija nego u običnim Gaussovim eliminacijama.*

3.3. LR faktorizacija

U praksi se linearni sustavi najčešće rješavaju korištenjem LR faktorizacije. Pretpostavimo da smo dobili matricu A faktoriziranu u obliku

$$A = LR, \tag{3.3.1}$$

pri čemu je L donjetrokutasta matrica s jedinicama na dijagonali, a R gornjetrokutasta. Matrica L je regularna i vrijedi $\det L = 1$, pa regularnost matrice A povlači i regularnost matrice R, jer iz (3.3.1) slijedi

$$\det A = \det L \cdot \det R = \det R.$$

Rješenje linearnog sustava (3.1.1) sad se svodi samo na dva rješavanja trokutastih sustava. Kako? Polazni sustav u faktoriziranoj formi ima oblik

$$LRx = b.$$

Označimo li $y = Rx$, dobivamo dva sustava

$$Ly = b, \quad Rx = y.$$

Oba sustava lako se rješavaju: prvi — supstitucijom unaprijed

$$\begin{aligned} y_1 &= b_1 \\ y_i &= b_i - \sum_{j=1}^{i-1} \ell_{ij}\, y_j, \quad i = 2, \ldots, n, \end{aligned}$$

a drugi — povratnom supstitucijom

$$
\begin{aligned}
x_n &= \frac{y_n}{r_{nn}} \\
x_i &= \frac{1}{r_{ii}} \Big(y_i - \sum_{j=i+1}^{n} r_{ij}\, x_j \Big), \quad i = n-1, \ldots, 1.
\end{aligned}
$$

Zašto je takva faktorizacija korisna? Na primjer, ako se rješavaju linearni sustavi kojima se mijenjaju samo desne strane, onda je dovoljno imati A spremljenu u faktoriziranom obliku, a zatim riješiti već navedena dva trokutasta sustava. Naravno, prvo treba naći LR faktorizaciju matrice A.

Relacije za elemente ℓ_{ij} i r_{ij} matrica L i R dobivamo ako iskoristimo njihovu poznatu strukturu i činjenicu da njihov produkt daje A. Onda je

$$
a_{ij} = \sum_{k=1}^{\min\{i,j\}} \ell_{ik}\, r_{kj},
$$

s tim da je $\ell_{ii} = 1$. Iz ovih relacija računamo redom one elemente koje možemo izraziti preko poznatih veličina. Tako dobivamo rekurziju za elemente matrica L i R

$$
\begin{aligned}
& r_{1j} = a_{1j}, \quad j = 1, \ldots, n, \\
& \ell_{j1} = \frac{a_{j1}}{r_{11}}, \quad j = 2, \ldots, n, \\
\text{za } & i = 2, \ldots, n: \\
& r_{ij} = a_{ij} - \sum_{k=1}^{i-1} \ell_{ik}\, r_{kj}, \quad j = i, \ldots, n, \\
& \ell_{ji} = \frac{1}{r_{ii}} \Big(a_{ji} - \sum_{k=1}^{i-1} \ell_{jk}\, r_{ki} \Big), \quad j = i+1, \ldots, n.
\end{aligned}
$$

U zadnjem koraku, za $i = n$, računamo samo r_{nn}. Jedini problem u provedbi ovog algoritma je osigurati da je $r_{ii} \neq 0$. Ako znamo da to vrijedi, onda prethodne relacije daju egzistenciju i jedinstvenost matrica L i R. Sljedeći teorem daje potrebni kriterij u terminima polazne matrice A.

Teorem 3.3.1. *Postoji jedinstvena LR faktorizacija matrice A ako i samo ako su vodeće glavne podmatrice $A_k := A(1:k, 1:k)$, $k = 1, \ldots, n-1$, regularne. Ako je A_k singularna za neki k, faktorizacija može postojati, ali nije jedinstvena.*

Dokaz:

Dokaz se provodi indukcijom po dimenziji matrice. Pretpostavimo da su sve matrice A_k regularne. Za $k = 1$, postoji jedinstvena LR faktorizacija

$$
A_1 = [1]\,[a_{11}].
$$

Pretpostavimo da A_{k-1} ima jedinstvenu faktorizaciju

$$A_{k-1} = L_{k-1}\, R_{k-1}.$$

Tražimo faktorizaciju matrice A_k, gdje je

$$A_k = \begin{bmatrix} A_{k-1} & b \\ c^T & a_{kk} \end{bmatrix} = \begin{bmatrix} L_{k-1} & 0 \\ \ell^T & 1 \end{bmatrix} \begin{bmatrix} R_{k-1} & r \\ 0 & r_{kk} \end{bmatrix} := L_k\, R_k.$$

Da bi jednadžbe bile zadovoljene, mora vrijediti

$$L_{k-1} r = b, \quad R_{k-1}^T \ell = c, \quad a_{kk} = \ell^T r + r_{kk}.$$

Matrice L_{k-1} i R_{k-1} su regularne, pa postoji jedinstveno rješenje r, ℓ, pa onda i jedinstveni r_{kk}.

Pokažimo obrat, uz pretpostavku da je A nesingularna i da postoji LR faktorizacija od A. Tada je $A_k = L_k\, R_k$, za $k = 1, \ldots, n$. Budući da je A regularna, vrijedi

$$\det A = \det R = r_{11}\, r_{22} \cdots r_{nn} \neq 0.$$

Odatle slijedi

$$\det A_k = r_{11}\, r_{22} \cdots r_{kk} \neq 0,$$

tj. sve matrice A_k su regularne.

Primjer koji ilustrira da LR faktorizacija može postojati u slučaju singularne matrice A, ali da nije jedinstvena, je faktorizacija nul-matrice

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ l & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

S druge strane, matrica

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

nema LR faktorizaciju, iako je regularna. ■

Pokažimo da je matrica R dobivena LR faktorizacijom jednaka gornjetrokutastoj matrici R dobivenoj Gaussovim eliminacijama. Pretpostavimo da je $A^{(k)}$ matrica dobivena u k-tom koraku Gaussovih eliminacija. Njezina blok forma ima oblik

$$A^{(k)} = \begin{bmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{bmatrix},$$

pri čemu je $A_{11}^{(k)}$ trokutasta matrica reda $k-1$ (tj. dosad sređena matrica), dok su preostale dvije matrice, generalno, pune. U matričnoj notaciji, sljedeći korak eliminacija možemo izraziti u obliku produkta

$$A^{(k+1)} = M_k A^{(k)} := \left[\begin{array}{c|ccccc} I_{k-1} & & & & & \\ \hline & 1 & & & & \\ & -m_{k+1,k} & 1 & & & \\ & -m_{k+2,k} & & \ddots & & \\ & \vdots & & & \ddots & \\ & -m_{n,k} & & & & 1 \end{array}\right] A^{(k)},$$

gdje su m_{ik} multiplikatori iz relacije (3.2.2). Matricu M_k možemo i kompaktno napisati kao

$$M_k = I - m_k e_k^T,$$

gdje je e_k, k-ti vektor kanonske baze, a m_k vektor s n komponenti,

$$m_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ m_{k+1,k} \\ \vdots \\ m_{n,k} \end{bmatrix}.$$

Primijetite da je

$$M_k^{-1} = I + m_k e_k^T,$$

jer je $e_i^T m_k = 0$ za $i \leq k$.

Prema tome je

$$M_{n-1}\, M_{n-2} \cdots M_1\, A = A^{(n)} := \widetilde{R}.$$

S druge strane, možemo dobiti i sam A

$$\begin{aligned} A &= M_1^{-1}\, M_2^{-1} \cdots M_{n-1}^{-1}\, \widetilde{R} = (I + m_1 e_1^T)\,(I + m_2 e_2^T) \cdots (I + m_{n-1} e_{n-1}^T)\, \widetilde{R} \\ &= \left(I + \sum_{i=1}^{n-1} m_i e_i^T\right) \widetilde{R} = \begin{bmatrix} 1 & & & & \\ m_{21} & 1 & & & \\ \vdots & m_{32} & \ddots & & \\ \vdots & \vdots & & \ddots & \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & 1 \end{bmatrix} \widetilde{R} := \widetilde{L}\, \widetilde{R}. \end{aligned}$$

Iz jedinstvenosti LR faktorizacije slijedi da je $\widetilde{R} = R$.

Teorem 3.3.1. i činjenica da je R iz LR faktorizacije jednak onom iz Gaussovih eliminacija, upućuju nas da pivotiranje vršimo na isti način kao i kod Gaussovih eliminacija.

Ako vršimo parcijalno pivotiranje, onda se LR faktorizacija tako dobivene matrice (permutiranih redaka) može zapisati kao

$$PA = LR,$$

pri čemu je P matrica permutacije — u svakom retku i stupcu ima točno jednu jedinicu, a ostalo su nule. Ako znamo "permutiranu" faktorizaciju, kako ćemo riješiti linearni sustav $Ax = b$? Najjednostavnije je lijevu i desnu stranu (slijeva) pomnožiti s P (P je uvijek regularna — pokažite to), pa dobivamo

$$PAx = LRx = Pb.$$

Ako vršimo potpuno pivotiranje, na kraju dobivamo LR faktorizaciju matrice koja ima permutirane retke i stupce obzirom na A, tj.

$$PAQ = LR,$$

gdje su P i Q matrice permutacije. U ovom je slučaju rješavanje linearnog sustava malo kompliciranije (skicirajte kako).

3.4. Teorija perturbacije linearnih sustava

U ovom odjeljku prezentirat ćemo rezultate klasične teorije perturbacije po normi linearnih sustava. Pitanje na koje odgovaraju takve teorije perturbacije je koliko se (po normi) rješenje linearnog sustava (3.1.1) promijeni ako se po normi/po komponentama malo promijene A, b ili oba.

Postoje razne vrste normi (pogledajte Osnovni udžbenik za detalje), ali će nama biti dovoljno da za normu vektora x (s n komponenti) uzmemo tzv. 2-normu, koja se vrlo često označava i kao $\|x\|_2$

$$\|x\| = \sqrt{\sum_{i=1}^{n} |x_i|^2},$$

a za normu matrice A s m redaka i n stupaca tzv. Frobeniusovu ili euklidsku normu

$$\|A\| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2}.$$

Pretpostavimo da, umjesto sustava (3.1.1), egzaktno rješavamo sustav

$$(A + \Delta A)(x + \Delta x) = b, \tag{3.4.1}$$

tj. samo je matrica sustava malo perturbirana. Možemo pretpostaviti da je norma perturbacije mala prema normi polazne matrice

$$\|\Delta A\| \leq \varepsilon \|A\|.$$

Zbog toga, umjesto x, dobili smo rješenje $x + \Delta x$.

Raspišimo (3.4.1) i iskoristimo (3.1.1). Izlazi

$$A\,\Delta x + \Delta A\,(x + \Delta x) = 0.$$

Množenjem slijeva s A^{-1} i sređivanjem dobivamo

$$\Delta x = -A^{-1}\,\Delta A\,(x + \Delta x).$$

Uzimanjem norme lijeve i desne strane, a zatim ocjenjivanjem odozgo, dobivamo

$$\begin{aligned}\|\Delta x\| &\leq \|A^{-1}\|\,\|\Delta A\|\,\|x + \Delta x\| \leq \varepsilon\,\|A^{-1}\|\,\|A\|\,\|x + \Delta x\| \\ &\leq \varepsilon\,\kappa(A)\,(\|x\| + \|\Delta x\|),\end{aligned}$$

pri čemu je $\kappa(A) = \|A\|\,\|A^{-1}\|$ standardna oznaka za uvjetovanost matrice A. Premještanjem na lijevu stranu svih pribrojnika koji sadrže Δx dobivamo

$$(1 - \varepsilon\,\kappa(A))\,\|\Delta x\| \leq \varepsilon\,\kappa(A)\,\|x\|.$$

Ako je $\varepsilon\,\kappa(A) < 1$, a to znači i $\|\Delta A\|\,\|A^{-1}\| < 1$, onda je

$$\|\Delta x\| \leq \frac{\varepsilon\,\kappa(A)}{1 - \varepsilon\,\kappa(A)}\,\|x\|, \tag{3.4.2}$$

što pokazuje da je pogreška u rješenju približno proporcionalna uvjetovanosti matrice A.

Pretpostavimo sad da, umjesto sustava (3.1.1), egzaktno rješavamo sustav

$$A(x + \Delta x) = b + \Delta b, \tag{3.4.3}$$

tj. samo je desna strana sustava malo perturbirana. Možemo pretpostaviti da je norma perturbacije mala prema normi vektora b

$$\|\Delta b\| \leq \varepsilon \|b\|.$$

Zbog te pertutrbacije, umjesto x, dobili smo rješenje $x + \Delta x$.

Raspišimo (3.4.3) i iskoristimo (3.1.1). Izlazi

$$A\,\Delta x = \Delta b.$$

Množenjem slijeva s A^{-1} dobivamo

$$\Delta x = A^{-1}\,\Delta b.$$

Uzimanjem norme lijeve i desne strane, a zatim ocjenjivanjem odozgo, dobivamo

$$\begin{aligned}\|\Delta x\| &\leq \|A^{-1}\|\,\|\Delta b\| \leq \varepsilon\,\|A^{-1}\|\,\|b\| \leq \varepsilon\,\|A^{-1}\|\,\|Ax\| \\ &\leq \varepsilon\,\|A^{-1}\|\,\|A\|\,\|x\| \leq \varepsilon\,\kappa(A)\,\|x\|,\end{aligned}$$

što pokazuje da je pogreška u rješenju, ponovno, proporcionalna uvjetovanosti matrice A.

Neke matrice koje izgledaju dosta "nevino" imaju vrlo visoku uvjetovanost čak i za ne preveliku dimenziju. Najpoznatije takve matrice su Hilbertove matrice, koje se često javljaju u praksi.

Primjer 3.4.1. *Riješite linearni sustav $H_n x = b_n$, gdje je H_n Hilbertova matrica reda n,*

$$H_n = \begin{bmatrix} 1 & \dfrac{1}{2} & \cdots & \dfrac{1}{n} \\ \dfrac{1}{2} & \dfrac{1}{3} & \cdots & \dfrac{1}{n+1} \\ \vdots & \cdots & & \vdots \\ \dfrac{1}{n} & \dfrac{1}{n+1} & \cdots & \dfrac{1}{2n-1} \end{bmatrix},$$

a vektor b_n je jednak

$$b_n = \begin{bmatrix} 1 + 2 + \cdots + \dfrac{1}{n} \\ \dfrac{1}{2} + \dfrac{1}{3} + \cdots + \dfrac{1}{n+1} \\ \vdots \\ \dfrac{1}{n} + \dfrac{1}{n+1} + \cdots + \dfrac{1}{2n-1} \end{bmatrix},$$

tako da je rješenje sustava uvijek vektor x kojemu su komponente jedinice.

Uvjetovanost Hilbertovih matrica je vrlo visoka:

n	$\kappa_2(H_n)$	n	$\kappa_2(H_n)$
2	$1.93 \cdot 10^{1}$	10	$1.60 \cdot 10^{13}$
3	$5.24 \cdot 10^{2}$	11	$5.23 \cdot 10^{14}$
4	$1.56 \cdot 10^{4}$	12	$1.71 \cdot 10^{16}$
5	$4.66 \cdot 10^{5}$	13	$5.63 \cdot 10^{17}$
6	$1.50 \cdot 10^{7}$	14	$1.85 \cdot 10^{19}$
7	$4.75 \cdot 10^{8}$	15	$6.12 \cdot 10^{20}$
8	$1.53 \cdot 10^{10}$	16	$2.02 \cdot 10^{22}$
9	$4.93 \cdot 10^{11}$		

Ovdje je napisana uvjetovanost u tzv. 2-normi (vidjeti Osnovni udžbenik), a uvjetovanost u Frobeniusovoj normi je samo veća!

Kao rezultat rješavanja sustava, umjesto x koji ima sve jedinice, ako računamo u `extended` *točnosti, dobivamo*

```
Hilbertova matrica reda  2  - izracunato rjesenje sustava
---------------------------------------------------------

x( 1) =     1.0000000000000000       x( 2) =     1.0000000000000000


Hilbertova matrica reda  5  - izracunato rjesenje sustava
---------------------------------------------------------

x( 1) =     1.0000000000000000       x( 4) =     0.9999999999999990
x( 2) =     0.9999999999999999       x( 5) =     1.0000000000000005
x( 3) =     1.0000000000000007


Hilbertova matrica reda 10  - izracunato rjesenje sustava
---------------------------------------------------------

x( 1) =     1.0000000000003436       x( 6) =     0.9999999294831902
x( 2) =     0.9999999999710395       x( 7) =     1.0000001151701616
x( 3) =     1.0000000006068386       x( 8) =     0.9999998890931838
x( 4) =     0.9999999945453735       x( 9) =     1.0000000580638087
x( 5) =     1.0000000258066880       x(10) =     0.9999999872591526
```

```
Hilbertova matrica reda 15  - izracunato rjesenje sustava
---------------------------------------------------------

x( 1) =     1.0000000005406387        x( 9) =     1.0952919444304200
x( 2) =     0.9999999069805858        x(10) =     0.8797820363884070
x( 3) =     1.0000039790948573        x(11) =     1.0994671444236333
x( 4) =     0.9999257525660447        x(12) =     0.9508102511158300
x( 5) =     1.0007543452271621        x(13) =     1.0106027108940050
x( 6) =     0.9953234190795597        x(14) =     1.0012346841153261
x( 7) =     1.0188643674562383        x(15) =     0.9992252029377023
x( 8) =     0.9487142544341838


Hilbertova matrica reda 20  - izracunato rjesenje sustava
---------------------------------------------------------

x( 1) =     1.0000000486333029        x(11) =   231.3608002738048500
x( 2) =     0.9999865995557111        x(12) =   -60.5143391625873562
x( 3) =     1.0008720556363132        x(13) =   -57.6674972682886125
x( 4) =     0.9760210562677670        x(14) =     5.1760567992057506
x( 5) =     1.3512820600312678        x(15) =     8.7242780841976215
x( 6) =    -2.0883247796748707        x(16) =   210.1722288687690970
x( 7) =    18.4001541798146106        x(17) =  -413.9544667202651170
x( 8) =   -63.8982130462650081        x(18) =   349.7671855031355400
x( 9) =   161.8392478869777220        x(19) =  -142.9134532513063250
x(10) =  -254.7902985140752950        x(20) =    25.0584794423327874
```