Сызықты алгебралық теңдеулер жүйесін шешу әдістері

Есептеу математикасының негізгі мәселелерінің бірі – нақты коэффициентті сызықтық алгебралық теңдеулер жүйесін шешу мәселесі. Сызықтық алгебралық теңдеулер жүйесінің жуық шешімін табу үшін тура және итерациялық әдістер қолданылады. Сызықтық алгебраның математикалық аппараты вектор мен матрица нормалары, шарттылық саны ұғымдарына негізделген. Белгісіздерді жоюдың классикалық әдістері қарастырылады, симметриялы нақты матрицасы бар есептерді шешу ерекшеліктері атап өтіледі.

Сызықты алгебралық теңдеулер жүйесі есебінің қойылымы

Сызықты алгебралық теңдеулер жүйесі келесі түрде болады

$$\begin{cases} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n = b_1 \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n = b_2 \\ \vdots \\ A_{n1}x_1 + A_{n2}x_2 + \cdots + A_{nn}x_n = b_n \end{cases} \tag{1}$$

мұндағы a_{ij} коэффициенттері мен b_j тұрақтылары белгілі, ал x_i белгісіздерді көрсетеді. Матрицалық түрде теңдеулер жүйесі былай жазылады:

$$\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{21} & A_{22} & \dots & A_{2n} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{2}$$

немесе

$$A \cdot x = b \tag{3}$$

A −коэффициент матрицасы, b −бос мүше вектор бағаны, x − белгісіз вектор бағаны.

Есептің қойылымы: (3)-ті тепе-теңдікке айналдыратын $x = (x_1, x_2, \dots, x_n)^T$-вектор бағанын табу керек.

Шешімнің жалғыздығы

n белгісізді n сызықтық теңдеулер жүйесінің жалғыз шешімі бар, егер коэффициент матрицасының анықтауышы айрықша емес болса; яғни $|\mathrm{A}| \neq 0$. Ешбір жол (немесе баған) басқа жолдардың (немесе бағандардың) сызықтық комбинациясы емес деген мағынада, айрықша емес матрицаның жолдары мен бағандары сызықтық тәуелсіз.

Нашар шарттылық

Ашық сұрақ мынада: егер коэффициент матрицасының анықтауышы айрықша болғанда не болады (яғни, $|\mathrm{A}| = 0$ немесе өте аз болса). Коэффициент матрицасының анықтауышы «кіші» екенін анықтау үшін анықтауышты өлшеуге болатын аппарат қажет. Бұл аппарат матрицаның нормасы деп аталады және $\|\mathrm{A}\|$ арқылы белгіленеді. Сонда анықтауыш аз деп айта аламыз, егер $|\mathrm{A}| \ll \|\mathrm{A}\|$

Қолданыстағы әдебиеттерде матрицаның және векторлардың бірнеше нормалары анықталған, мысалы, Евклид нормасы

Векторлар үшін | Матрицалар үшін

$$\|x\|_{\mathrm{E}} = \sqrt{\sum_{j=1}^{n} |x_j|^2} \qquad \|A\|_{\mathrm{E}} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}^2} \tag{4}$$

және қатар-қосынды норманы шексіздік нормасы деп те атайды

Векторлар үшін | Матрицалар үшін

$$\|x\|_{\infty} = \max_{1 \le j \le n} |x_j| \qquad \|A\|_{\infty} = \max_{1 \le i \le n} \sum_{j=1}^{n} |A_{ij}| \tag{5}$$

Шарттылықтың формальды өлшемі ретінде келесі түрде анықталған матрицаның шарттылық саны болып табылады

$$cond(A) = \|A\| \cdot \|A^{-1}\|$$

Егер шарттылық сан бірлікке жақын болса, матрица жақсы шартталған. Шарттылық саны айрықша матрица үшін нашар шартталған дәрежесіне қарай өсе отырып, шексіздікке кетіп қалады.

Матрицалық норманы таңдауға байланысты шарттылық саны бірдей болмайтынын ескеріңіз. Өкінішке орай, үлкен матрицалар үшін шарттылық санын есептеу көп шығынға ұшыратады.

Нашар шартталған теңдеулердің сандық шешімдеріне сенуге болмайды. Себебі шешу үрдісінде жасалатын дөңгелектеу қателері коэффициент матрицасына аздаған өзгерістер енгізуге тура келеді. Бұл өз кезегінде шешімге үлкен қателер енгізеді, олардың мөлшері нашар шартталған деңгейіне байланысты.

Күдікті жағдайларда коэффициент матрицасының анықтауышын есептеу керек, ол нашар шартталған дәрежесін бағалауға болатындай. Мұны шешу кезінде немесе одан кейін аз ғана есептеу күшімен жасауға болады.

Сызықтық жүйелер

Сызықтық, алгебралық теңдеулер сандық талдаудың барлық салаларында кездеседі. Бірақ олардың инженериядағы ең көрнекті қолданылуы сызықтық жүйелерді талдау болып табылады (есептің жауабы бастапқы мәліметтерге пропорционал болатын кез келген жүйе сызықтық болып саналады).

Егер жүйе дискретті болса, онда оны талдау тікелей сызықтық алгебралық теңдеулерге әкеледі. Статикалық анықталатын ферма жағдайында, мысалы, теңдеулер қосылыстардың тепе-теңдік шарттарын жазғанда пайда болады. $x_1, x_2, \ldots, x_n$ белгісіздер мүшелердегі күштер мен тірек реакцияларын көрсетеді, ал $b_1, b_2, \ldots, b_n$ тұрақтылары берілген сыртқы жүктемелер болып табылады.

Үздіксіз жүйелердің әрекеті алгебралық теңдеулермен емес, дифференциалдық теңдеулермен сипатталады. Алайда, сандық талдау тек дискретті айнымалылармен айналысатындықтан, алдымен дифференциалдық теңдеуді жуықтап алгебралық теңдеулер жүйесі келтіріп алу қажет. Осыдан талдаудың белгілі ақырлы айырымдық, ақырлы элементтер және шекаралық элемент әдістері жұмыс істейді. Олар «дискретизацияға» қол жеткізу үшін әртүрлі жуықтауларды пайдаланады, бірақ әр жағдайда түпкілікті тапсырма бірдей: сызықтық, алгебралық теңдеулер жүйесін (көбінесе өте үлкен жүйе) шешу.

Қорытындылай келе, сызықтық жүйелерді модельдеу әрқашан $Ax = b$ түріндегі теңдеулерге алып келеді, мұнда $b-$ бастапқы мәліметтер және $x-$ жүйенің жауабын білдіреді. Жүйенің сипаттамаларын көрсететін коэффициент матрицасы A бастапқы мәліметтерге тәуелсіз. Басқаша айтқанда, егер бастапқы мәліметтер өзгертілсе, теңдеулерді басқа b-мен, бірақ сол A көмегімен қайтадан шешу керек. Сондықтан, минималды есептеу күшімен тұрақты векторлардың кез келген санын өңдей алатын теңдеулерді шешу алгоритмі болғаны жөн.

Шешу әдістері

Сызықтық алгебралық теңдеулер жүйесін шешу әдістерінің екі класы бар: Тура және итерациялық әдістер. Тура әдістердің жалпы сипаттамасы олар бастапқы теңдеулерді оңайырақ шешілетін эквивалентті теңдеулерге (шешімі бірдей теңдеулер) түрлендіру болып табылады. Түрлендіру келесі үш амалды қолдану арқылы жүзеге асырылады. Бұл элементар деп аталатын амалдар шешімді өзгертпейді, бірақ олар жақшада көрсетілген коэффициент матрицасының анықтауышына әсер етуі мүмкін.

1. Екі теңдеуді алмастыру ($|A|$ белгісін өзгертеді)

2. Теңдеуді нөлге тең емес тұрақтыға көбейту ($|A|$-ны сол тұрақтыға көбейтеді)

3. Теңдеуді нөлге тең емес тұрақтыға көбейту, содан кейін оны басқа теңдеуден алу ($|A|$ өзгеріссіз қалдырады).

Итерациялық әдістер x шешімін болжаудан басталады, содан кейін белгілі бір жинақтылық шартына жеткенше шешімді қайта-қайта нақтылайды. Итерациялық әдістер талап етілетін итерациялар санының көп болуына байланысты олардың тура әдістеріне қарағанда әдетте тиімділігі төмен. Дегенмен, егер коэффициент матрицасы өте үлкен және аз санды болса (коэффиценттердің көпшілігі нөлге тең) болса, олардың айтарлықтай есептеу артықшылықтары бар.

Тура әдістерге шолу

1-кестеде үш танымал тура әдістер келтірілген, олардың әрқайсысы өзінің оңай шешілетін теңдеулердің соңғы түрін шығару үшін қарапайым амалдарды қолданады.

Әдіс	Бастапқы түрі	Қорытынды түрі
Гаусс	$Ax = b$	$Ux = c$
LU жіктеу	$Ax = b$	$LUx = b$
Гаусс-Жордан	$Ax = b$	$Ix = c$

1-кесте. Үш танымал тура әдістер

1-кестеде U жоғарғы үшбұрышты матрицаны, L төменгі үшбұрышты матрицаны, және I бірлік матрицасын белгілейді. Квадрат матрица үшбұрышты деп аталады, егер оның басты диагональдың бір жағында тек нөлдік элементтер болса. Ендеше 3×3 жоғарғы үшбұрышты матрицаның түрі

$$\begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

және 3×3 төменгі үшбұрышты матрицаның түрі

$$\begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix}$$

Үшбұрышты матрицалар сызықтық алгебрада маңызды рөл атқарады, өйткені олар көптеген есептеулерді жеңілдетеді.

Үшбұрышты түрге клтіру процедурасы тура(алға) жүріс деген атаумен белгілі. Гаусстың біртіндеп жоюда $Ux = c$ қорытынды түріне келгеннен кейін, соңғы теңдеуден бастап, кері жүріс арқылы есептің шешімін оңай табады.

LU жіктеуімен байланысты $LUx = b$ теңдеулерін, егер оларды екі эквивалентті теңдеулер жинағымен ауыстырсақ, тез шешуге болады: $Ly = b$ және $Ux = y$. Енді $Ly = b$ y үшін тура жүріс арқылы шешіледі, содан кейін кері жүріс арқылы $Ux = y$ шешіледі.

Гаусс-Жорданның біртіндеп жою арқылы алынған $Ix = c$ теңдеулері $x = c$ тең ($Ix = x$ бірлік матрица екенін еске түсіріңіз), сондықтан c шешім болып табылады.

Гаусстың біртіндеп жою әдісінің алгоритмі

Біртіндеп жою кезеңі. Біртіндеп жою кезеңінде қандай да бір сәттегі теңдеулерді қарастырайық. A матрицасының алғашқы k жолы жоғарғы үшбұрышты пішінге түрлендірілді делік. Демек, ағымдағы тірек теңдеу k-ші теңдеу болып табылады және одан төмен барлық теңдеулер әлі түрленеді. Бұл жағдай келесі кеңейтілген коэффициент матрицасы арқылы бейнеленген.

$$\begin{array}{c@{\quad}l}
\left[\begin{array}{ccccccccc|c}
A_{11} & A_{12} & A_{13} & \cdots & A_{1k} & \cdots & A_{1j} & \cdots & A_{1n} & b_1 \\
0 & A_{22} & A_{23} & \cdots & A_{2k} & \cdots & A_{2j} & \cdots & A_{2n} & b_2 \\
0 & 0 & A_{33} & \cdots & A_{3k} & \cdots & A_{3j} & \cdots & A_{3n} & b_3 \\
\vdots & \vdots & \vdots & & \vdots & & \vdots & & \vdots & \vdots \\
0 & 0 & 0 & \cdots & A_{kk} & \cdots & A_{kj} & \cdots & A_{kn} & b_k \\
\hline
\vdots & \vdots & \vdots & & \vdots & & \vdots & & \vdots & \vdots \\
0 & 0 & 0 & \cdots & A_{ik} & \cdots & A_{ij} & \cdots & A_{in} & b_i \\
\vdots & \vdots & \vdots & & \vdots & & \vdots & & \vdots & \vdots \\
0 & 0 & 0 & \cdots & A_{nk} & \cdots & A_{nj} & \cdots & A_{nn} & b_n
\end{array}\right]
&
\begin{array}{l}
\\ \\ \\ \\
\leftarrow \text{Тірек жол} \\
\\
\leftarrow \text{Түрленетін жол} \\
\\ \\
\end{array}
\end{array}$$

А матрицасының элементтері бастапқы теңдеулердің коэффициенттері емес екенін ескеріңіз (бірінші жолды қоспағанда), себебі олар біртіндеп жою процедурасы арқылы өзгертілген. Бұл b тұрақты векторының элементтеріне де қатысты.

Түрлендірілетін i-ші жол тірек теңдеудің астындағы әдеттегі жол болсын, яғни A_{ik} элементі жойылады. Біз бұған тірек жолды $\lambda = A_{ik}/A_{kk}$ көбейтіп, оны i-ші жолдан шегеру арқылы қол жеткізе аламыз. i-ші жолдағы сәйкес өзгерістер

$$A_{ij} \leftarrow A_{ij} - \lambda \cdot A_{kj}, \quad j = k, k+1, \dots, n \tag{6}$$

$$b_i \leftarrow b_i - \lambda \cdot b_k \tag{7}$$

Бүкіл коэффициент матрицасын жоғарғы үшбұрыш түріне түрлендіру үшін (6)-(7) формуларында k және i коэффициенттері келесі диапазонда жүріп өтуі керек: $k = 1,2, \dots, n-1$ (тірек жол); $i = k+1, k+2, \dots, n$ (түрлендірілетін жол). Біртіндеп жою кезеңінің алгоритмі:

```
for k in range(0,n-1):
    for i in range(k+1,n):
        if a[i,k] != 0.0:
            lam = a[i,k]/a[k,k]
            a[i,k+1:n] = a[i,k+1:n] - lam*a[k,k+1:n]
            b[i] = b[i] - lam*b[k]
```

Қажетсіз әрекеттерді болдырмау үшін алдыңғы алгоритм мынадай жолдармен бастапқы (6)-(7) амалдардан сәл ауытқиды:

Егер A_{ik} нөлге тең болса, і жолының түрлендіруі өткізіліп жіберіледі.

(6) формуладағы j индексі k емес, $k+1$-ден басталады. Сондықтан A_{ik} нөлге ауыстырылмайды, бірақ өзінің бастапқы мәнін сақтайды. Шешімді табу фазасы коэффициент матрицасының төменгі үшбұрышты бөлігіне ешқашан кірмейтіндіктен, оның мазмұны маңызды емес.

Кері жүріс кезеңі. Гаусстың біртіндеп жоюдан кейін кеңейтілген коэффициент матрицасы келесі түрге ие болады

$$\left[\mathbf{A}\,\middle|\,\mathbf{b}\right]=\left[\begin{array}{ccccc|c} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} & b_1 \\ 0 & A_{22} & A_{23} & \cdots & A_{2n} & b_2 \\ 0 & 0 & A_{33} & \cdots & A_{3n} & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_{nn} & b_n \end{array}\right]$$

Соңғы теңдеу $A_{nn} \cdot x_n = b_n$ бірінші шешіліп,

$$x_n = b_n / A_{nn} \tag{8}$$

шығады.

Енді кері жүріс кезеңін қарастырайық, мұнда $x_n, x_{n-1}, \dots, x_{k+1}$ есептеліп қойылсын делік (осы ретпен), және біз k-ші теңдеуден x_k анықтағалы жатырмыз.

$$A_{kk}x_k + A_{k,k+1}x_{k+1} + \cdots + A_{kn}x_n = b_k$$

Шешімі

$$x_k = \left(b_k - \sum_{j=k+1}^{n} A_{kj}x_j\right)\frac{1}{A_{kk}}, \qquad k = n-1, n-2, \dots, 1 \tag{9}$$

Кері жүрістің сәйкес алгоритмі

```
for k in range(n-1,-1,-1):
    x[k]=(b[k] - dot(a[k,k+1:n],x[k+1:n]))/a[k,k]
```

Операциялар саны. Алгоритмнің орындалу уақыты негізінен орындалатын ұзақ амалдардың (көбейту және бөлу) санына байланысты. Гаусстың біртіндеп жою әдісінде тура жүріс кезеңінде шамамен $n^3/3$ осындай амалдар (n-теңдеулер саны) және кері жүріс $n^2/2$ амалдар бар екенін көрсетуге болады. Бұл сандар есептеу уақытының көп бөлігі тура жүріс кезеңіне кететінін көрсетеді. Сонымен қатар, уақыт теңдеулер санымен өте жылдам артады.

Гаусстың біртінден жою әдісімен есепті шешу коды.

```
import numpy as np
def gaussElimin(a,b):
    n = len(b)
    # Tura juris
    for k in range(0,n-1):
        for i in range(k+1,n):
            if a[i,k] != 0.0:
                lam = a [i,k]/a[k,k]
                a[i,k+1:n] = a[i,k+1:n] - lam*a[k,k+1:n]
                b[i] = b[i] - lam*b[k]
    # keri juris
    for k in range(n-1,-1,-1):
        b[k] = (b[k] - np.dot(a[k,k+1:n],b[k+1:n]))/a[k,k]
    return b
```

gaussElimin функциясы тура мен кері жүріс кезеңдерін біріктіреді. Кері жүріс кезінде шешім x векторы b арқылы қайта жазылады, осылайша b шығу кезінде шешімді қамтиды.

```
In [3]: import numpy as np
def gaussElimin(a,b):
    n = len(b)
    # Tura juris
    for k in range(0,n-1):
        for i in range(k+1,n):
            if a[i,k] != 0.0:
                lam = a [i,k]/a[k,k]
                a[i,k+1:n] = a[i,k+1:n] - lam*a[k,k+1:n]
                b[i] = b[i] - lam*b[k]
    # keri juris
    for k in range(n-1,-1,-1):
        b[k] = (b[k] - np.dot(a[k,k+1:n],b[k+1:n]))/a[k,k]
    return b


a = np.array([[ 5.0, 0.0, 1.0], \
              [2.0, 6.0, -2.0], \
              [-3.0, 2.0, 10.0]])
b = np.array([11.0, 8.0, 6.0])
det = np.prod(np.diagonal(a))
print("\nAnyqtauyshy =",det)
x = gaussElimin(a,b)
print("x=",x)
```

```
Anyqtauyshy = 300.0
x= [2. 1. 1.]
```

LU жіктеу әдісі

Кез келген А квадрат матрицасын төменгі L үшбұрышты матрица мен жоғарғы U үшбұрышты матрицаның көбейтіндісі ретінде көрсетуге болатынын көрсетуге болады:

$$A = LU \tag{10}$$

Берілген А үшін L және U есептеу үрдісі LU жіктеуі немесе LU ыдырауы ретінде белгілі. LU жіктеуі бірмәнді емес (тағайындалған A үшін L және U комбинациялары шексіз), егер L немесе U-ға белгілі бір шектеулер қойылмаса. Бұл шектеулер жіктеудің бір түрін екіншісінен ажыратады.

А ыдырағаннан кейін жоғарыда көрсетілгендей $Ax = b$ теңдеулерін шешу оңай. Алдымен теңдеулерді $LUx = b$ деп қайта жазамыз. $Ux = y$ белгісін қолданғаннан кейін теңдеулер болады

$$\boldsymbol{Ly = b}$$

оны y үшін тура жүріс арқылы шешуге болады. Содан кейін

$$\boldsymbol{Ux = y}$$

кері жүріс үрдісі арқылы x береді.

LU жіктеуінің Гаусстың біртіндеп жою әдісіне қарағанда артықшылығы мынада: А жіктелгеннен кейін біз қалағанша b тұрақты векторлары үшін $Ax = b$ мәнін шеше аламыз. Әрбір қосымша шешімді табудың уақыты салыстырмалы түрде аз, өйткені тура және кері жүріс амалдары жіктеу үрдісіне қарағанда әлдеқайда аз уақытты алады.

Дулитлдің жіктеу әдісі(Doolittle’s Decomposition Method)

Жіктеу кезеңі. Дулитлдің жіктеуі Гаусстың біртіндеп жою әдісімен тығыз байланысты.

Жіктеуді көрсету үшін 3×3 А матрицасын қарастырамыз және $A = LU$ болатындай үшбұрышты матрицалар бар деп есептейік

$$L = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \quad U = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

Көбейтуді орындағаннан кейін біз

$$A = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ U_{11}L_{21} & U_{12}L_{21} + U_{22} & U_{13}L_{21} + U_{23} \\ U_{11}L_{31} & U_{12}L_{31} + U_{22}L_{32} & U_{13}L_{31} + U_{23}L_{32} + U_{33} \end{bmatrix} \tag{11}$$

аламыз. Енді Гаусстың біртіндеп жою әдісін (11)-ге қолданайық. Біртіндеп жою процедурасының бірінші жүрісі бірінші жолды тірек жол ретінде таңдаудан және келесі қарапайым амалдарды жасаудан тұрады.

$$2\text{жол} \leftarrow 2\text{жол} - L21 \times 1\text{жол}\ (A21\text{ді жояды})$$
$$3\text{жол} \leftarrow 3\text{жол} - L31 \times 1\text{жол}\ (A31\text{ді жояды})$$

Нәтижесінде

$$A' = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & U_{22}L_{32} & U_{23}L_{32} + U_{33} \end{bmatrix}$$

Келесі жүрісте біз екінші жолды тірек жол ретінде алып, келесі соңғы амалды қолданамыз

$$3\text{жол} \leftarrow 3\text{жол} - L32 \times 2\text{жол}\ (A32\text{ні жояды})$$

Жоғарыда келтірілген Дулитлдің жіктелуінің екі маңызды ерекшелігін көрсетеді:

1. U матрицасы Гаусстың біртіндеп жою нәтижесінде пайда болатын жоғарғы үшбұрышты матрицамен бірдей.

2. L диагональдан тыс элементтері Гаусстың біртіндеп жою кезінде қолданылатын тірек жолдардың көбейткіштері болып табылады; яғни $L_{ij} - A_{ij}$элементтері жойылған көбейткіш.

Көбейткіштерді коэффициент матрицасының төменгі үшбұрышты бөлігінде сақтау әдеттегі тәжірибе болып табылады, олар жойылған кезде коэффициенттерді ауыстырады (L_{ij} орнына A_{ij}).

L диагональ элементтерін сақтаудың қажеті жоқ, өйткені олардың әрқайсысы бірге тең екендігі белгілі. Осылайша, коэффициент матрицасының соңғы түрі келесі L және U бірігіп былай жазылады:

$$[L\backslash U] = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ L_{21} & U_{22} & U_{23} \\ L_{31} & L_{32} & U_{33} \end{bmatrix} \tag{12}$$

Дулитлдің жіктеу алгоритмі осылайша gaussElimin функциясындағы Гаусстың біртіндеп жою процедурасымен бірдей, тек әрбір көбейткіш λ енді A-ның төменгі үшбұрышты бөлігінде сақталады:

```
for k in range(0,n-1):
    for i in range(k+1,n):
        if a[i,k] != 0.0:
            lam = a[i,k]/a[k,k]
            a[i,k+1:n] = a[i,k+1:n] - lam*a[k,k+1:n]
            a[i,k] = lam
```

Шешімді табу кезеңі. Енді $Ly = b$ шешімін тура жүріспен шешу процедурасын қарастырайық. Теңдеулердің скаляр түрі ($L_{ii} = 1$ екенін еске түсірейік)

$$\begin{aligned} y_1 &= b_1 \\ L_{21}y_1 + y_2 &= b_2 \\ &\vdots \\ L_{k1}y_1 + L_{k2}y_2 + \cdots + L_{k,k-1}y_{k-1} + y_k &= b_k \\ &\vdots \end{aligned}$$

k-ші теңдеуден y_k табамыз

$$y_k = b_k - \sum_{j=1}^{k-1} L_{kj} y_j , \qquad k = 2,3, \dots , n \tag{13}$$

Сондықтан тура жүріс алгоритмі келесідей болады

```
y[0] = b[0]
for k in range(1,n):
    y[k] = b[k] - dot(a[k,0:k],y[0:k])
```

$Ux = y$ шешуге арналған кері жүріс кезеңі Гаусстың біртіндеп жою әдісінде қолданылғанмен бірдей.

LU жіктеу әдісімен есепті шешу коды

```
import numpy as np
def LUdecomp(a):
    n = len(a)
    for k in range(0,n-1):
        for i in range(k+1,n):
            if a[i,k]!= 0.0:
                lam = a [i,k]/a[k,k]
                a[i,k+1:n] = a[i,k+1:n] - lam*a[k,k+1:n]
                a[i,k] = lam
    return a
def LUsolve(a,b):
    n = len(a)
    for k in range(1,n):
        b[k] = b[k] - np.dot(a[k,0:k],b[0:k])
        b[n-1] = b[n-1]/a[n-1,n-1]
    for k in range(n-2,-1,-1):
        b[k] = (b[k] - np.dot(a[k,k+1:n],b[k+1:n]))/a[k,k]
return b
```

Бұл модульде жіктеу және шешу кезеңдері бар. Жіктеу кезеңінде (12) формуласында көрсетілген $[L\backslash U]$ матрицасын қайтарады. Шешім кезеңінде тура жүріс кезінде b векторы y-ке ауыстырылады. Сол сияқты, кері жүрісте *y* шешімін *x* шешімімен қайта жазады.

Қолданылған әдебиеттер тізімі

1) Шакенов Қ.Қ. Есептеу математикасы әдістері лекциялар курсы.Алматы,2019. – 193б
2) P. Dechaumphai, N. Wansophark. Numerical Methods in Science and Engineering Theories with MATLAB, Mathematica, Fortran, C and Python Programs. Alpha Science International Ltd. 2022
3) Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков Численные методы: Классический университетский учебник. —М.: Издательство «Бином. Лаб. знаний»,2020. — 636 с.
4) Вабищевич П.Н. Численные методы: Вычислительный практикум. — М.: Книжный дом «ЛИБРОКОМ», 2020. — 320 с.

Интернет ресурстар

1) https://docs.python.org/3/

2) http://math-hse.info/f/2018-19/py-polit/instruction_JN.pdf
3) https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html
4) https://colab.research.google.com/
5) https://planetcalc.ru/search/?tag=2874