4 Mnożenie macierzy a rozwiązywanie równań liniowych

4.1 Sformułowanie problemu

Zadania z tytułu są równoważne w tym sensie, że mając algorytm o złożoności $c_1 \cdot n^\alpha$ dla $\alpha \geqslant 2$ rozwiązujący jedno z tych zadań potrafimy rozwiązać drugie za pomocą algorytmu o złożoności $c_2 \cdot n^\alpha$ dla odpowiednio dobranej stałej c_2. Pojęcie złożoności rozumiemy tu jako złożoność algebraiczną. Tak więc algorytmem o złożoności f jest taki algorytm, który dla danych rozmiaru n wykonuje nie więcej niż $f(n)$ operacji arytmetycznych.

Oprócz wymienionych zadań będziemy także zajmować się kilkoma innymi. Wykażemy równoważność podanym sensie następujących zadań:

1) obliczania iloczynu macierzy,

2) odwracania macierzy trójkątnych,

3) odwracania dowolnych macierzy,

4) rozwiązywania układów równań liniowych.

Sformułowania zadań będą jeszcze uściślane. Będziemy też zajmować się obliczaniem tzw. rozkładu LUP.

Dowodząc równoważności będziemy często konstruować algorytmy rekurencyjne, które będzie można łatwiej opisać i analizować przy założeniu, że dane są rozmiaru będącego potęgą dwójki. Zauważmy, że podobnie jak w przypadku algorytmu mnożenia macierzy, przy pewnych założeniach algorytm o złożoności $c_1 \cdot n^\alpha$ dobrze działający dla potęg dwójki można przekształcić w algorytm rozwiązujący zadanie dla dowolnych danych bez istotnego pogarszania złożoności. Przypuśćmy, że

1) w określony sposób powiększamy dane do możliwie małego rozmiaru będącego potęgą dwójki nie wykonując żadnych działań arytmetycznych (zauważmy, że w najgorszym razie podwoimy rozmiar danych),

2) następnie uruchamiamy algorytm o złożoności $c_1 \cdot n^\alpha$ dla powiększonych danych

3) a na koniec zwracamy część obliczonych wartości (a więc też nie wykonujemy operacji arytmetycznych).

Nietrudno zauważyć, że taki algorytm ma złożoność $c_1 \cdot (2 \cdot n)^\alpha = (2^\alpha \cdot c_1) \cdot n^\alpha$.

4.2 Odwracanie macierzy trójkątnych

Zadanie 4.1 Odwracanie macierzy trójkątnej (górnej).

Dane: macierz kwadratowa A górnie trójkątna o wymiarach $n \times n$,

Rozmiar zadania: liczba wierszy macierzy A, Obliczyć macierz odwrotną do A lub stwierdzić, że macierz odwrotna do A nie istnieje.

Zauważmy, że jeżeli istnieją macierze A^{-1} i B^{-1}, to

$$\begin{bmatrix} A & C \\ 0 & B \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & -A^{-1} \cdot C \cdot B^{-1} \\ 0 & B^{-1} \end{bmatrix}.$$

Stąd, w szczególności, wynika, że

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

dla dowolnej odwracalnej macierzy A.

Z podanych wzorów wynika następujący algorytm odwracania macierzy trójkątnych:

Algorytm $\mathcal{O}(A)$ odwracania macierzy trójkątnej (o wymiarach będących potęgą dwójki)

1) $n :=$ wymiar A;
2) jeżeli $n = 1$ i $A = [0]$, to zwróć informację, że macierz A nie jest odwracalna i zakończ działanie programu;
3) jeżeli $n = 1$ i $A = [a]$, to zwróć macierz $[a^{-1}]$ i zakończ działanie programu;
4) niech X, Y, Z będą macierzami takimi, że
$$i_{2,n/2}(A) = \begin{bmatrix} X & Z \\ 0 & Y \end{bmatrix}$$
(pamiętajmy, że n jest potęgą dwójki, a A jest macierzą trójkątną, stąd m.in. wynika, że X i Y są macierzami trójkątnymi);
5) $U := \mathcal{O}(X)$;
6) $V := \mathcal{O}(Y)$;
7) $W := -(U \cdot Z) \cdot V$;
8) zwróć macierz
$$i^{-1}_{2,n/2}\left(\begin{bmatrix} U & W \\ 0 & V \end{bmatrix}\right).$$

Przypuśćmy, że algorytm $\mathcal{O}$ korzysta z algorytmu mnożenia macierzy, który dla pewnego $\alpha \geqslant 2$ i dla macierzy o rozmiarze $n \times n$ wykonuje $\leqslant c \cdot n^\alpha$ mnożeń. Niech $O'_M(m)$ oznacza mnożeń wykonanych podczas odwracania macierzy o wymiarach $m \times m$ zgodnie z algorytmem $\mathcal{O}$. Dla argumentów będących potęgami dwójki funkcja O'_M spełnia nierówność

$$O'_M(m) \leqslant 2 \cdot O'_M(\frac{m}{2}) + 2 \cdot c \cdot (\frac{m}{2})^\alpha.$$

Udowodnimy, że

$$O'_M(m) \leqslant c \cdot m^\alpha.$$

Na podstawie zasady indukcji i założenia $\alpha \geqslant 2$ otrzymujemy to z następujących nierówności:

$$O'_M(m) \leqslant 2 \cdot c \cdot (\frac{m}{2})^\alpha + 2 \cdot c \cdot (\frac{m}{2})^\alpha \leqslant \frac{4}{2^\alpha} \cdot c \cdot m^\alpha \leqslant c \cdot m^\alpha.$$

Analogiczne oszacowania są prawdziwe dla liczby dodawań i liczby działań wykonywanych podczas realizacji algorytmu $\mathcal{O}$.

Dowolne macierze trójkątne odwracamy stosując następujący algorytm:

Algorytm odwracania macierzy trójkątnej (A jest daną macierzą)

1) Niech B oznacza możliwie małą macierz postaci

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

o wymiarach będących potęgą dwójki;

2) $C := \mathcal{O}(B)$;

3) nietrudno zauważyć, że macierz C jest postaci

$$\begin{bmatrix} D & 0 \\ 0 & 1 \end{bmatrix}$$

dla pewnej macierzy D o wymiarach macierzy A);

4) zwróć macierz D.

4.3 Rozkład LUP

Niech A będzie macierzą o n kolumnach i $m \leqslant n$ liniowo niezależnych wierszach. Pokażemy, że $A = L \cdot U \cdot P$ dla pewnych macierzy L, U i P takich, że

1) L i P są macierzami kwadratowymi,

2) L jest dolną macierzą trójkątną, tzn. jeżeli $l_{i,j}$ jest j-tym wyrazem i-tego wiersza tej macierzy i $j > i$, to $l_{i,j} = 0$,

3) wyrazy L znajdujące się na przekątnej są różne od 0 ($l_{i,i} \neq 0$),

4) U jest górną macierzą trójkątną, tzn. jeżeli $u_{i,j}$ jest j-tym wyrazem i-tego wiersza tej macierzy oraz $j < i$, to $u_{i,j} = 0$,

5) wyrazy U znajdujące się na przekątnej są różne od 0 ($u_{i,i} \neq 0$),

6) P jest macierzą permutującą, tzn. w każdym wierszu i w każdej kolumniej tej macierzy znajduje się dokładnie jeden wyraz równy 1, a pozostałe są równe 0.

Przedstawienie macierzy w postaci iloczynu macierzy o wyżej podanych własnościach nazywamy rozkładem LUP. Łatwo dowodzi się, że jeżeli macierz ma rozkład LUP, to jej wiersze są liniowo niezależne.

Zadanie 4.2 Znajdowanie rozkładu LUP.

Dane: macierz A o n kolumnach i $m = 2^l \leqslant n$ wierszach,

Rozmiar zadania: liczby wierszy i kolumn macierzy A,

Znaleźć rozkład LUP macierzy A lub stwierdzić, że dla macierzy A taki rozkład nie istnieje.

Algorytm $\mathcal{R}(A)$ (rozkładu macierzy, których liczba wierszy jest potęgą dwójki)

1) $m :=$ liczba wierszy A;

2) jeżeli $m = 1$ i wiersz składa się z samych zer, to zwracamy informację, że macierz A ma liniowo zależne wiersze i przerywamy wykonywanie algorytmu;

3) jeżeli $m = 1$, to ustalamy indeks i wyrazu kierunkowego macierzy (wektora) A i zwracamy $L = [1]$, P powstające z macierzy jednostkowej przez zamianę pierwszego wiersza z i-tym oraz $U = AP$ (U powstaje przez zamianę pierwszego elementu z kierunkowym, zauważmy, że PP jest macierzą jednostkową, a więc $UP = A$) i przerwamy działanie algorytmu;

4) niech B i C będą macierzami o $m/2$ wierszach takimi, że
$$A = \begin{bmatrix} B \\ C \end{bmatrix};$$

5) wywołując rekurencyjnie algorytm $\mathcal{R}$ znajdujemy L_1, U_1 i P_1 takie, że $B = L_1U_1P_1$;

6) niech $U_1 = [X, Y]$ oraz $CP_1^{-1} = [Z, T]$ dla pewnych kwadratowych macierzy X i Z.

7) obliczamy $E = ZX^{-1}$, $F = EY$ oraz $G = T - F$;

8) przedstawiamy G w postaci $L_2U_2P_2$ korzystając z rekurencyjnego wywołania algorytmu $\mathcal{R}$;

9) zwracamy
$$L = \begin{bmatrix} L_1 & 0 \\ ZX^{-1} & L_2 \end{bmatrix},$$
$$U = \begin{bmatrix} X & YP_2^{-1} \\ 0 & U_2 \end{bmatrix},$$
$$P = \begin{bmatrix} 1 & 0 \\ 0 & P_2 \end{bmatrix} P_1.$$

Udowodnimy teraz lematy 4.1 i 4.2, z których wynika poprawność algorytmu $\mathcal{R}(A)$. Zauważmy, że z drugiego lematu wynika także, że macierz A ma rozkład LUP wtedy i tylko wtedy, gdy jej wiersze są liniowo niezależne.

Lemat 4.1 *Jeżeli algorytm $\mathcal{R}$ zwróci macierze L, U i P, to $A = L \cdot U \cdot P$.*

Dowód. Dowodzimy ten fakt przez indukcję. Zauważmy, że

$$
\begin{aligned}
LUP &= \begin{bmatrix} L_1 & 0 \\ ZX^{-1} & L_2 \end{bmatrix} \begin{bmatrix} X & YP_2^{-1} \\ 0 & U_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & P_2 \end{bmatrix} P_1 \\
&= \begin{bmatrix} L_1X & L_1YP_2^{-1} \\ Z & ZX^{-1}YP_2^{-1} + L_2U_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & P_2 \end{bmatrix} P_1 \\
&= \begin{bmatrix} L_1X & L_1Y \\ Z & ZX^{-1}Y + L_2U_2P_2 \end{bmatrix} P_1 = \begin{bmatrix} L_1X & L_1Y \\ Z & F+G \end{bmatrix} P_1 \\
&= \begin{bmatrix} L_1X & L_1Y \\ Z & T \end{bmatrix} P_1 = \begin{bmatrix} L_1U_1 \\ CP_1^{-1} \end{bmatrix} P_1 = \begin{bmatrix} L_1U_1P_1 \\ C \end{bmatrix} = \begin{bmatrix} B \\ C \end{bmatrix} = A. \ \square
\end{aligned}
$$

Lemat 4.2 *Algorytm $\mathcal{R}$ stwierdza liniową zależność wierszy macierzy A wtedy i tylko wtedy, gdy A ma liniowo zależne wiersze.*

Dowód. Najpierw ustalimy kilka pomocniczych faktów. Przypuśćmy, że mamy dwie macierze A i B. Wierszami tych macierzy są odpowiednio wektory $\vec{a}_i \in R^k$ dla $i = 1, \ldots, l$ oraz $\vec{b}_j \in R^l$. Niech $f : R^k \to R^l$ będzie funkcją zdefiniowaną wzorem

$$f(\vec{x}) = \sum_{j=1}^{l} x_j \cdot \vec{b}_j.$$

Fakt 4.3 Wektor $f(\vec{a}_i)$ jest i-tym wierszem macierzy AB. $\square$

Fakt 4.4 Wektory $\vec{b}_1, \ldots \vec{b}_l$ są liniowo niezależne wtedy i tylko wtedy, gdy funkcja f jest wzajemnie jednoznaczna. $\square$

Fakt 4.5 Jeżeli funkcja f jest wzajemnie jednoznaczna, to wektory $\vec{a}_1, \ldots, \vec{a}_k$ są liniowo niezależne wtedy i tylko wtedy, gdy wektory $f(\vec{a}_1), \ldots, f(\vec{a}_k)$ są liniowo niezależne. $\square$

Fakt 4.6 Jeżeli $|A| \neq 0$, to wiersze macierzy B są liniowo niezależne wtedy i tylko wtedy, gdy wiersze macierzy AB są liniowo niezależne. Jeżeli $|B| \neq 0$, to wiersze macierzy A są liniowo niezależne wtedy i tylko wtedy, gdy wiersze macierzy AB są liniowo niezależne. $\square$

Możemy wrócić do dowodu lematu. Będzie to dowód indukcyjny. Najpierw załóżmy, że wiersze danej macierzy A są liniowo niezależne. Wtedy macierze B i C (patrz opis algorytmu $\mathcal{R}$) są liniowo niezależne. Z założenia indukcyjnego wynika, że algorytm $\mathcal{R}$ nie stwierdzi liniowej zależności wierszy macierzy B i zwróci jej rozkład.

Będziemy teraz porównywać liniową niezależność wierszy macierzy A i G przy założeniu, że macierz $B = L_1U_1P_1$ ma wiersze liniowo niezależne. Zauważmy, że

$$\begin{bmatrix} L_1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X & Y \\ Z & T \end{bmatrix} P_1 = \begin{bmatrix} L_1X & L_1Y \\ Z & T \end{bmatrix} P_1 = \begin{bmatrix} L_1U_1 \\ CP_1^{-1} \end{bmatrix} P_1 = \begin{bmatrix} B \\ C \end{bmatrix} = A.$$

Stąd i z faktu 4.6 wynika, że macierz A ma liniowo niezależne wiersze wtedy i tylko wtedy, gdy ta własność przysługuje macierzy

$$\begin{bmatrix} X & Y \\ Z & T \end{bmatrix}.$$

Mamy także

$$\begin{bmatrix} -ZX^{-1} & 1 \end{bmatrix} \begin{bmatrix} X & Y \\ Z & T \end{bmatrix} = \begin{bmatrix} 0 & T - ZX^{-1}Y \end{bmatrix} = \begin{bmatrix} 0 & G \end{bmatrix}.$$

Pierwsza z powyższych macierzy ma liniowo niezależne wiersze (ponieważ zawiera kwadratową macierz jednostkową 1). Z podanych równości otrzymujemy, że macierz A ma liniowo niezależne wiersze wtedy i tylko wtedy, gdy macierz $[0\ G]$ ma wiersze liniowo niezależne, a to jest oczywiście równoważne liniowej niezależności wierszy macierzy G.

Wobec tego i na podstawie założenia indukcyjnego drugie rekurencyjne wywołanie algorytmu $\mathcal{R}$ (z macierzą G jako daną) także nie spowoduje stwierdzenia liniowej zależności wierszy macierzy A.

Dla dowodu implikacji odwrotnej załóżmy, że macierz A ma liniowo zależne wiersze. Możemy założyć, że wiersze macierzy B są liniowo niezależne, gdyż w przeciwnym razie pierwsze wywołanie rekurencyjne algorytmu $\mathcal{R}$ spowoduje stwierdzenie liniowej zależności wierszy macierzy A. Z dodatkowego założenia, stosując równoważność wykazaną w dowodzie poprzedniej implikacji pokazujemy, że także wiersze macierzy G są liniowo zależne. W tym przypadku drugie wywołanie rekurencyjne algorytmu $\mathcal{R}$ stwierdzi liniową zależność wierszy macierzy A. □

Oszacujemy złożoność algorytmu $\mathcal{R}$. Niech $R(n,m)$ oznacza liczbę operacji arytmetycznych wykonywanych podczas rozkładania macierzy o m wierszach i n kolumnach za pomocą algorytmu $\mathcal{R}$ (m powinno więc być potęgą dwójki). Analizując algorytm stwierdzamy, że $R(n,1) = 0$ oraz

$$R(n,m) \leqslant R(n,\frac{m}{2}) + O(\frac{m}{2}) + M(\frac{m}{2}) + \left\lceil \frac{2\cdot n - m}{m} \right\rceil \cdot M(\frac{m}{2}) + \frac{m}{2}(n - \frac{m}{2}) + R(n - \frac{m}{2}, \frac{m}{2}),$$

gdzie $O(m)$ i $M(m)$ oznaczają odpowiednio liczby operacji wymaganych do odwrócenia macierzy trójkątnej o rozmiarze $m \times m$ i do pomnożenia dwóch takich macierzy (współczynnik przy drugiej wartości funkcji M to liczba wywołań procedury mnożenia macierzy kwadratowych potrzebna do obliczenia iloczynu EY).

Przypuśćmy, że podczas wykonywania algorytmu $\mathcal{R}$ stosujemy algorytmy odwracające macierze trójkątne i mnożące macierze kwadratowe o złożoności $c\cdot m^\alpha$ dla pewnego $\alpha > 2$. Weźmy liczbę

$$\gamma = \max(2, \frac{4}{2^\alpha - 4}).$$

Zauważmy, że
$$\frac{2\gamma + 2}{2^{\alpha-1}} \leqslant \gamma.$$
Przez indukcję udowodnimy, że $R(n,m) \leqslant \gamma \cdot c \cdot m^{\alpha-1} \cdot n$. Nierówność ta wynika z następujących rachunków:

$$\begin{aligned}
& R(n,m) \\
& \leqslant R(n, \frac{m}{2}) + O(\frac{m}{2}) + M(\frac{m}{2}) + \left\lceil \frac{2 \cdot n - m}{m} \right\rceil \cdot M(\frac{m}{2}) + \frac{m}{2}(n - \frac{m}{2}) + R(n - \frac{m}{2}, \frac{m}{2}) \\
& \leqslant \gamma \cdot c \cdot (\frac{m}{2})^{\alpha-1} \cdot n + 2 \cdot c \cdot (\frac{m}{2})^{\alpha} + \frac{2 \cdot n}{m} \cdot c \cdot (\frac{m}{2})^{\alpha} + \frac{m}{2} \cdot n + \gamma \cdot c \cdot (\frac{m}{2})^{\alpha-1} \cdot (n - \frac{m}{2}) \\
& \leqslant c \cdot (\frac{m}{2})^{\alpha-1} \cdot n \cdot (\gamma + 1 + 1 + \gamma) + 2 \cdot c \cdot (\frac{m}{2})^{\alpha} - \gamma \cdot c \cdot (\frac{m}{2})^{\alpha} \\
& \leqslant c \cdot m^{\alpha-1} \cdot n \cdot \frac{2 \cdot \gamma + 2}{2^{\alpha-1}} \leqslant \gamma \cdot c \cdot m^{\alpha-1} \cdot n.
\end{aligned}$$

4.4 Zastosowania rozkładu LUP: obliczanie wyznaczników

Algorytm obliczający wyznacznik:

1) daną macierz A zastępujemy macierzą
$$B = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix}$$
o wymiarze będącym potęgą dwójki (zauważmy, że macierze A i B mają te same wyznaczniki);

2) znajdujemy przedstawienie macierzy B w postaci LUP (uruchamiamy algorytm rozwiązujący zadanie 4.2),

3) jeżeli macierz B została przedstawiona w postaci LUP, to obliczamy wyznaczniki macierzy L, U oraz P, mnożymy je przez siebie i zwracamy obliczony iloczyn, w przeciwnym razie zwracamy 0.

Ponieważ macierze L i U są trójkątne, więc ich wyznaczniki obliczamy mnożąc elementy z przekątnych, macierz P ma wyznacznik równy 1 lub -1 i można go znaleźć ustalając liczbę przestawień niezbędną do przekształcenia P w macierz jednostkową.

4.5 Zastosowania rozkładu LUP: odwracanie macierzy

Zadanie 4.3 Odwracanie macierzy.

Dane: macierz kwadratowa A,

Rozmiar zadania: liczba wierszy macierzy A,

Obliczyć macierz A^{-1} lub stwierdzić, że macierz odwrotna do A nie istnieje.

Algorytm obliczający macierz odwrotną:

1) daną macierz A zastępujemy macierzą
$$B = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix}$$
o wymiarze będącym potęgą dwójki,

2) obliczamy przedstawienie macierzy B w postaci LUP uruchamiając algorytm rozwiązujący zadanie 4.2,

3) jeżeli macierz B ma przedstawienie postaci LUP, to znajdujemy B^{-1} korzystając ze wzoru $B^{-1} = P^{-1}U^{-1}L^{-1}$, w przeciwnym razie stwierdzamy, że macierz B nie jest odwracalna.

Zauważmy, że
$$\begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & 0 \\ 0 & 1 \end{bmatrix}.$$
Wzór $(AB)^T = B^T A^T$ implikuje, że $A^{-1} = ((A^T)^{-1})^T$ i pozwala obliczyć macierz odwrotną do macierzy dolnej trójkątnej stosując algorytm obliczania macierzy odwrotnej do górnej macierzy trójkątnej. Ponadto $P^{-1} = P^T$. Oczywiście, złożoność tego algorytmu jest tego samego rzędu, co złożoność algorytmu mnożenia macierzy.

4.6 Mnożenie macierzy za pomocą odwracania

Wzór
$$\begin{bmatrix} 1 & A & 0 \\ 0 & 1 & B \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -A & AB \\ 0 & 1 & -B \\ 0 & 0 & 1 \end{bmatrix}$$
pozwala zredukować mnożenie macierzy do obliczania macierzy odwrotnej. Jeżeli posłużymy się algorytmem odwracającym macierz o złożoności $c \cdot n^\alpha$, to oparty o podany wzór algorytm obliczający iloczyn macierzy będzie miał złożoność $(c \cdot 3^\alpha) \cdot n^\alpha$.

Przedstawiony rezultat dowodzi, że równoważne są wszystkie zadania wymienione w rozdziale 4.1 z wyjątkiem ostatniego (rozwiązywania równań liniowych). Równoważnosć ostatniego zadania i pozostałych wynika z następnego rozdziału.

4.7 Rozwiązywanie układów równań liniowych

Ustalmy najpierw czym jest rozwiązanie układu równań liniowych. Zgodnie z rozdziałem 1.3, aby rozwiązać układ $A\vec{x} = \vec{b}$ powinniśmy znaleźć jedno rozwiązanie tego układu (jeżeli istnieje) i bazę (lub zbiór generatorów) przestrzeni $\{\vec{x} : A\vec{x} = \vec{0}\}$.

Rozważmy układ równań liniowych postaci $\vec{x} = A\vec{y} + \vec{b}$, gdzie A jest macierzą o m wierszach długości n. Z postaci tego układu, bez żadnych obliczeń, możemy odczytać jego rozwiązanie. Tak więc wektor $(\vec{x}, \vec{y})$ równy $(\vec{b}, \vec{0}) \in R^{n+m}$ spełnia to równanie, a przestrzeń

$\{(\vec{x}, \vec{y}) \in R^{n+m} : \vec{x} = A\vec{y}\}$ jest generowana przez wektory $(\vec{a}_i, \vec{e}_i)$, gdzie $\vec{a}_i$ jest i-tą kolumną macierzy A, dla $1 \leqslant i \leqslant n$. Albo inaczej, przestrzeń ta jest generowana przez kolumny macierzy $\begin{bmatrix} A \\ 1 \end{bmatrix}$. Nietrudno też zauważyć, że opisany zbiór generatorów składa się z wektorów liniowo niezależnych. Wobec tego, zadanie rozwiązania danego układu równań może być rozumiane jako polecenie znalezienia równoważnego mu układu postaci $\vec{x} = A\vec{y} + \vec{b}$, a sam ten układ możemy uważać za rozwiązanie danego.

Będzie interesować nas kilka zadań. Najbardziej szczegółowym jest następujące.

Zadanie 4.4 Rozwiązać układ równań liniowych wyrażając określone zmienne przez pozostałe.

> Dane: układ $A\vec{x} = \vec{b}$ złożony z $m \leqslant n$ równań liniowych z n niewiadomymi,
>
> Rozmiar: liczba równań m i liczba niewiadomych n,
>
> Znaleźć macierz C i wektor $\vec{d}$ takie, że dany układ równań jest równoważny układowi postaci $(x_1, \ldots, x_m) = C(x_{m+1}, \ldots, x_n) + (d_1, \ldots, d_m)$, gdzie C oznacza tu pewną macierz o m wierszach i $n - m$ kolumnach (wektory utożsamiamy z jednokolumnowymi macierzami) lub stwierdzić, że nie jest to możliwe (albo mówiąc inaczej: wyrazić pierwsze m zmiennych przez pozostałe lub stwierdzić, że rozwiązania nie da się tak wyrazić).

Zadanie 4.4 można rozwiązać stosując następujący algorytm:
Algorytm rozwiązujący zadanie 4.4

1) Macierz A przedstawiamy w postaci $[A_1, A_2]$, gdzie A_1 jest macierzą kwadratową,

2) obliczamy macierz odwrotną do macierzy A_1 uruchamiając algorytm rozwiązujący zadanie 4.3,

3) jeżeli A_1 ma macierz odwrotną, to obliczamy i zwracamy iloczyny $C = A_1^{-1}A_2$ i $\vec{d} = A_1^{-1}\vec{b}$, w przeciwnym razie stwierdzamy, że dany układ nie ma rozwiązania żądanej postaci.

Nietrudno zauważyć, że złożoność tego algorytmu wyraża się wzorem $c \cdot m^{\alpha-1} \cdot n$, gdzie α jest wykładnikiem z wzoru szacującego złożoność algorytmu mnożenia macierzy. Co więcej, algorytm ten świadczy o tym, zadanie 4.4 można zredudować do problemu odwracania macierzy (zadanie 4.3). Wątpliwości może jednak budzić jego poprawność. Wynika ona z następującego lematu:

Lemat 4.7 *Jeżeli A jest macierzą kwadratową, a układy $A\vec{x} + B\vec{y} = \vec{c}$ oraz $\vec{x} + D\vec{y} = \vec{e}$ są równoważne, to A jest macierzą odwracalną.*

Dowód. Przypuśćmy, że $A\vec{a} = \vec{0}$ (a więc rozważamy liniową kombinację kolumn macierzy A ze współczynnikami $\vec{a}$, równą zero). Każde równanie takiej postaci, jak drugie, ma rozwiązanie. Wobec tego, rozwiązanie musi mieć też pierwsze równanie. Weźmy dowolne jego rozwiązanie

$(\vec{x}_0, \vec{y}_0)$. Rozwiązaniem pierwszego równania jest też $(\vec{x}_0+\vec{a}, \vec{y}_0)$. Korzystając z równoważności obu równań, otrzymujemy, że rozwiązania te spełniają drugie równanie. Tak więc

$$\vec{x}_0 + \vec{a} + D\vec{y}_0 = \vec{e} = \vec{x}_0 + D\vec{y}_0.$$

Stąd wynika, że $\vec{a} = \vec{0}$. W ten sposób dowiedliśmy, że kolumny macierzy A są liniowo niezależne, a to implikuje tezę. $\square$

Zauważmy, że prawdziwy jest też następujący lemat:

Lemat 4.8 *Jeżeli układy $\vec{x} + B_1\vec{y} = \vec{c}_1$ oraz $\vec{x} + B_2\vec{y} = \vec{c}_2$ są równoważne, to $B_1 = B_2$ i $\vec{c}_1 = \vec{c}_2$.*

Dowód. Oczywiście, wektor $(\vec{c}_1, \vec{0})$ jest rozwiązaniem pierwszego równania, a więc jest także rozwiązaniem drugiego. Podstawiając ten wektor do drugiego równania otrzymujemy, że $\vec{c}_1 = \vec{c}_2$. Z tej równości z kolei wynika, że $B_1\vec{y} = B_2\vec{y}$ dla wszystkich $\vec{y}$. Tak więc $B_1\vec{e}_i = B_2\vec{e}_i$ dla wszystkich dopuszczalnych i. Równość ta oznacza, że i-te kolumny macierzy B_1 i B_2 są identyczne. $\square$

Z lematu 4.8 wynika następujący algorytm odwracania macierzy (zadanie 4.3):

Algorytm odwracania macierzy

1) Dla danej macierzy A tworzymy układ równań $A\vec{x} - I\vec{y} = \vec{0}$, gdzie I jest macierzą jednostkową o takich samych wymiarach, jak macierz A,
2) obliczamy rozwiązanie równania $A\vec{x} - I\vec{y} = \vec{0}$ postaci $\vec{x} - B\vec{y} = \vec{c}$,
3) Jeżeli znajdziemy takie rozwiązanie, to zwracamy macierz B, w przeciwnym razie stwierdzamy, że macierz A nie jest odwracalna.

Z lematu 4.7 wynika, że powyższy algorytm uruchomiony z macierzą nieodwracalną stwierdzi ten fakt, a jeżeli uruchomimy go z macierzą odwracalną A, to znajdzie rozwiązanie równania $A\vec{x} - I\vec{y} = \vec{0}$ i obliczy odpowiednią macierz B. Mnożąc to równanie stronami przez A^{-1} otrzymujemy $\vec{x} - A^{-1}\vec{y} = \vec{0}$. Z lematu 4.8 wynika, że $B = A^{-1}$, a więc w tym przypadku algorytm zwróci macierz odwrotną do A. Zauważmy także, że dwa ostatnie algorytmy świadczą o równoważności zadań 4.4 i 4.3.

Polecenie rozwiązania układu równań liniowych może też być rozumiane jako zadanie 4.5 i rozwiązane w inny sposób.

Zadanie 4.5 Rozwiązywanie układu niezależnych równań liniowych

Dane: układ $A\vec{x} = \vec{b}$ złożony z m liniowo niezależnych równań (liniowych) z n niewiadomymi (tak więc $m \leqslant n$),

Rozmiar: liczba równań i liczba niewiadomych,

Znaleźć rozwiązanie danego układu równań.

Algorytm rozwiązujący zadanie 4.5

1) dany układ m równań z n zmiennymi powiększamy do układu złożonego 2^l równań dopisując możliwie mało równań postaci $y_i = 0$, używając nowych zmiennych y_i; przypuśćmy, że w ten sposób otrzymaliśmy równanie $A'\vec{x}' = \vec{b}'$

2) macierz A' danego układu o μ wierszach długości ν przedstawiamy w postaci LUP stosując algorytm $\mathcal{R}$,

3) rozwiązujemy układ $L\vec{y} = \vec{b}'$ stosując metodę macierzy równoważnych wierszowo; niech $\vec{y}_0$ będzie znalezionym rozwiązaniem,

4) obliczamy $\vec{z} = P\vec{x}'$ (czyli przestawiamy odpowiednio zmienne w ciągu $\vec{x}'$),

5) rozwiązujemy układ $U\vec{z} = \vec{y}_0$ stosując algorytm rozwiązujący zadanie 4.4 i wyrażając pierwsze m zmiennych ciągu $\vec{z}$ przez pozostałe,

6) zwracamy rozwiązanie znalezione w poprzednim punkcie usuwając z niego informacje o dodatkowych zmiennych y_i.

Zauważmy, że rozwiązanie metodę macierzy równoważnych wierszowo układu $L\vec{y} = \vec{b}$ takiego, że A jest macierzą trójkątną (górną lub dolną) o wymiarach $\mu \times \mu$ wymaga wykonania najwyżej $c \cdot \mu^2$ operacji arytmetycznych. Przestawienie zmiennych w ciągu $\vec{x}'$ (obliczenie $P\vec{x}'$ dla permutacyjnej macierzy P) nie wymaga żadnych operacji arytmetycznych. Stosując odpowiednie algorytmy rozkładania macierzy i rozwiązywania zadania 4.4, punkty 1 i 4 rozważanego algorytmy można zrealizować wykonującą $c \cdot \mu^{\alpha-1} \cdot \nu$ operacji arytmetycznych. Wobec tego podany algorytm wykonuje najwyżej

$$c \cdot \mu^{\alpha-1} \cdot \nu + c_1 \cdot \mu^2 + c \cdot \mu^{\alpha-1} \cdot \nu \leqslant (2 \cdot c + c_1) \cdot \mu^{\alpha-1} \cdot \nu$$

operacji arytmetycznych. Ponieważ $\mu < 2 \cdot m$ i $\nu < 2 \cdot n$, więc analizowany algorytm ma złożoność

$$\leqslant (2 \cdot c + c_1) \cdot 2^\alpha \cdot m^{\alpha-1} \cdot n.$$

Polecenie rozwiązania układu równań liniowych najbardziej ogólnie może być rozumiane jako

Zadanie 4.6 Rozwiązywanie układu równań liniowych.

Dane: układ $A\vec{x} = \vec{b}$ złożony z m równań liniowych z n niewiadomymi,

Rozmiar: liczba równań i liczba niewiadomych,

Znaleźć rozwiązanie danego układu równań lub stwierdzić, że układ nie ma rozwiązania.

Zadanie 4.6 może zostać zredukowane do zadania 4.5. Można to zrobić w sposób opisany w kolejnych rozdziałach.