ZESZYTY NAUKOWE WYŻSZEJ SZKOŁY PEDAGOGICZNEJ W BYDGOSZCZY
Problemy Matematyczne 1980 z. 2

Andrzej W. Mostowski
Uniwersytet Gdański

INFORMATYKA NA WYKŁADZIE ALGEBRY I ALGEBRY LINIOWEJ

W.W. Sawyer zauważył z pozoru paradoksalny fakt, że słabym uczniom w szkole o wiele łatwiej jest przyswajać wywody prowadzone przy użyciu bardzo nawet skomplikowanych rozważań ciągłościowych, niż z pozoru prostsze wywody algebraiczne.
Wydaje się, że przyczyną tego jest coś, co można by nazwać "stopniem odległości od praktyki". Przecież rozważania ciągłościowe stanowią, choć w myśli, wykonywanie czegoś.

Autor sądzi, że przyczyna trudności w nauczaniu algebry liniowej z geometrią na pierwszym roku i związany z tym fakt istnienia bardzo licznej grupy studentów z trudnością rozumiejącej materiał, leży również w odległości od praktyki.
Kilka pomysłów przedstawionych poniżej a związanych z wprowadzeniem elementów informatyki, choćby tylko do wykładu algebry liniowej z geometrią na sekcji nienauczycielskiej, wiąże się z chęcią zaradzenia tej sytuacji i przekonaniem, że pokazanie sposobu liczenia i danie możności, choćby w wyobraźni, policzenia czegoś, winno ułatwić przekroczenie /lub ominięcie/ progu abstrakcji, podobnie jak stworzenie uczniom w szkole możliwości rozważań ciągłościowych.

Autor tych słów nie wprowadził nigdy na żadnym wykładzie wszystkich na raz przedstawionych niżej innowacji i zawsze równolegle starał się wyjaśnić wszystko w sposób tradycyjny. Nie wszystkie jego pomysły są nowe i wiele przynależy po prostu do dobrego wykładania algebry.
Autor nie jest również w stanie stwierdzić, czy jego wysił-

ki odniosły pozytywny skutek. Z całą jednak pewnością nie odniosły skutku negatywnego i spotykały się z zaciekawieniem i życzliwym zainteresowaniem studentów.

Dla wygody wygłoszonego tu wykładu, autor podzielił proponowane przez siebie innowacje na kilka grup odpowiadających różnym kwestiom związanym z tym, co potocznie nazywa się informatyką a jest właściwie współczesną teorią obliczeń.
Kwestie te to: języki programowania, złożoność obliczeń, organizacja obliczeń, konstrukcja procedur rekurencyjnych z indukcyjnych dowodów istnienia. Poświęcone im są poniższe paragrafy.
Na zakończenie znajduje się kilka słów o tym, czego w wykładzie algebry nie robię choć ma to związek z informatyką.

1. Sposób oznaczeń bliższy językom programowania

Dlaczego nie pisać zamiast:
niech A będzie macierzą o m wierszach i n kolumnach po prostu:

array A[1 : m; 1 : n]

zaś element Ai,j macierzy oznaczać A[i,j].
Dlaczego nie powiedzieć, że zapisanie na taśmie danych elementów macierzy w postaci:
A[1,1],..., A[1,n],..., A[m,1],..., A[m,n] świadczy o izomorfizmie przestrzeni

$$K_n^m \text{ i } K^{m \times n}$$

Dlaczego nie użyć zapisów:

d[i,j] : = a[i,j] + b[i,j];
f[i,j] : = c × a[i,j];

a przede wszystkim zapisu:

e[i,j] : = **if** i = j **then** 1 **else** 0;

Dla mnożenia macierzy jest to wprawdzie trochę bardziej skomplikowane, gdyż trzeba napisać:

z : = 0;

```
        for k : = 1 step 1 until n do
Z := Z + a[i,k] × b[k,j];
        c[i,j] : = z;
```

ale ile rzeczy można przy tej okazji wytłumaczyć nie gubiąc wcale geometrycznych intuicji mnożenia macierzy.

Doskonale udaje się również wyjaśnienie pewnych rozróżnień związanych z wielomianami.

Przecież wielomian W to:

```
        array W [0 : pseudostopień];
```

zaś stopień wielomianu to maksimum zmiennej pseudostopień, dla której W [pseudostopień] ≠ 0.

Zauważmy, że stopień wielomianu zerowego jest przy tym nieokreślony . Wartość wielomianu możnaby określić podając jej sposób wyliczania np. przy pomocy następującej procedury:

```
procedure wartość (wielomian, zmienna);
          array wielomian [ 0 : pseudostopień];
          real zmienna;
               begin
                    real wartość;
                    integer k;
                    wartość : = 0;
                    for k : = pseudostopień step - 1
                             until 0 do
                    wartość : = zmienna × wartość + wielo-
                             mian [ k];
                    comment czytelniku, jak się nazywa użyty
                             wzór;
               end
```

W tym ujęciu doskonale widać rozróżnienie pomiędzy wielomianem a wartością wielomianu, gdyż ta ostatnia jest dwuparametrową procedurą.

Wzory na iloczyn h wielomianów f oraz g można łatwo zapisać w postaci skrótowej:

s : = stopień (f) + stopień (g);

```
begin
    array fnowe [1 : s], gnowe [1 : s];
    for i : = 0 step 1 until s do
        begin
            if ( i ≤ stopień ( f ))
                then fnowe [ i ] : = f[i]
                    else fnowe [ i ] : = 0;
                        if ( i ≤ stopień ( g ))
                            then
                        gnowe[i] : = g[i]
                                else
                            gnowe[i]: = 0
        end
comment czytelniku po co takie przygotowania;
for k : = 0 step 1 until s do
    begin
        z : = 0;
        for i : = 0 step 1 until k do
            z : = z + fnowe [i] × gnowe [ k - i];
        h[k] : = z
    end
  end
```

i przypomnieć na tym przykładzie:
że wielomian to ciąg <u>nieskończony</u> f[i];
dla którego tylko <u>skończona</u> a niewiadomo jaka ilość wyrazów
jest równa 0 a więc, zarówno f jak i fnowe są opisami tego
samego wielomianu.

Można następnie w zależności od chęci uczestników zaproponować albo napisanie pełnej procedury

iloczyn wielomianów (f, g);

albo wyznaczenie ilości działań przy przyjętym sposobie obliczania.

2. Złożoność obliczeń i organizacja obliczeń:

Dobrym przykładem szacowania ilości operacji jest teoria układów równań liniowych prowadzona przez sprowadzenie macierzy układu do postaci trójkątnej:

Redukcje macierzy do postaci trójkątnej:

$$
\begin{array}{l}
x[1] + a[1,2] \times x[2] + \ldots + a[1,n] \times x[n] = b[1] \\
\qquad a[2,2] \times x[2] + \ldots + a[2,n] \times x[n] = b[2] \\
\qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\qquad a[m,2] \times x[2] + \ldots + a[m,n] \times x[n] = b[m]
\end{array}
$$

wymaga: dla pierwszego równania n + 1 dzieleń dla następnych równań: n + 1 mnożeń i n + 1 odejmowań, razem działań:

$$(m + 1) \times 2 \times (n + 1) + n + 1 \approx 2\, m \times n$$

wynika stąd, że redukcja macierzy do postaci trójkątnej wymaga:

$$\sum_{i=1}^{n} 2\, i^2 = \frac{1}{3}\, n\,(n + 1)(2n + 1) \text{ działań.}$$

Rozwiązanie równania w postaci trójkątnej wymaga wtedy już tylko

$n\,(n - 1)$ mnożeń i dodawań.

Daje to łącznie:

$$\frac{1}{3}\left[n\,(n + 1)(2n + 1) + 3n(n - 1)\right] \leq n^3.$$

Rachunki takie jak wyżej na wykładzie zawsze robię i porównuję wykładaną metodę rozwiązywania układów równań, ze sposobem opartym o wzory Cramera wymagającym w istocie $nx(n + 1)!$ działań.

Żałuję tylko zawsze, że nie mogę poprosić sali o wyjęcie kalkulatorów celem rozwiązania jakiegoś konkretnego układu równań i policzenia jednocześnie, ile razy musimy nastawiać liczbę na klawiaturze i ile razy musimy liczbę notować na kawałku papieru.
Natomiast, jeżeli zdarza mi się mówić o wzorach Cramera, to robiąc przykład obliczenia np. $x[1] = A[1]/A$, $x[2] = A[2]/A$, $x[3] = A[3]/A$, dzielę salę na cztery części, mówiąc, że obliczenia będziemy robić równolegle, wy stanowicie cztery procesory, to jest jeden, to drugi itd., a ja jestem jed-

nostką centralną koordynującą przebieg rachunków.

Staram się również akcentować różnice pomiędzy rachunkiem na liczbach i na literach. Doskonale nadaje się do tego obliczanie rzędu macierzy.

3. Procedury rekurencyjne a dowody indukcyjne

Doskonałym przykładem pokazywania, że dowody indukcyjne istnienia jakiejś wielkości stanowią procedury rekurencyjne dla obliczania tej wielkości, jest wspomniane już wyżej twierdzenie o sprowadzaniu układu równań liniowych do postaci trójkątnej.
Inne doskonałe przykłady to dzielenie wielomianów; znajdowanie n.w.d metodą algorytmu Euklidesa.
Mówiąc o tym ostatnim, winno się, jak uważam, wspomnieć o nieużyteczności jego dla celów obliczeń przy pomocy e.m.c., o roli zera w maszynie i kumulowaniu się błędów zaokrągleń. Należy, uważam, wspomnieć również w jakimś miejscu o problemie obliczania pierwiatków wielomianów wyższych stopni i braku zadawalających programów dla znajdywania wszystkich pierwiastków.

4. Czego nie robię. Nie mówię oczywiście /lub niestety, jak kto woli/ nic ani o wyspecjalizowanych sposobach obliczeń ani o metodach numerycznych algebry liniowej. Nie mówię również nic o metodach optymalizacji liniowej, uważając to za zupełnie oddzielne zagadnienie specjalistyczne. Nie staram się również specjalnie zapisywać wszystkich wzorów i procedur w sposób syntaktycznie poprawny, dostarczając choć z rzadka słuchaczom radości zauważenia, że czegoś brakuje - najczęściej zresztą średnika.

Trzeba niestety natomiast, w wykładzie algebry liniowej starannie podawać, traktowane zwykle po macoszemu, twierdzenia i pojęcia potrzebne do metod numerycznych algebry liniowej. Są to w pierwszym rzędzie dwie grupy zagadnień:

a - twierdzenia o macierzach symetrycznych i wartościach własnych,

b - o dzielnikach elementarnych i postaci normalnej zarówno Jordana jak i Fro beniusa.

THE INFORMATIES AN THE LECTURES OF THE ALGEBRA AND THE LINEAR ALGEBRA

Summary

The article contains the autor's view that:

a/ a programming language notation of vectors and matrices;

b/ procedure type notation of algebraic operations;

c/ an estimation of computational complexity for linear equations;

when used during a first course lecture of linear algebra on the section of numerical methods with programmation can ease a beter understanding of algebra and train a proper programmation skills as well.