

VI. Tablice, macierze i wyeksponowane równania

Wiesław Krakowiak

16 marca 2015

Środowisko array

Środowisko array służy do tworzenia struktur tablicowych zawierających wyrażenia matematyczne. Jest ono bardzo podobne do środowiska tabular i ma analogiczne własności i możliwości. Zasadniczą różnicą jest konieczność używania środowiska array w trybie matematycznym.

Otoczenie **array** ma ogólną postać:

```
\begin{array}[poz]{preambuła}  
lista-wierszy  
\end{array}
```

Obowiązkowy parametr preambuła

Parametr preambuła określa liczbę kolumn tabeli i sposób ich justowania. Przyjmuje on następujące wartości:

- `l` – kolumna wyrównywania jest do lewej;
- `r` – kolumna wyrównywania jest do prawej;
- `c` – kolumna wyrównywania jest centrowana;
- `|` – wstawiana jest pojedyncza pionowa linia oddzielająca sąsiednie kolumny tabeli;
- `\|` – wstawiana jest podwójna pionowa linia oddzielająca kolumny tabeli;

Parametry środowiska `array` (`cd`)

Dla każdej kolumny należy w argumencie wstawić jedna z liter: `l`, `r` lub `c`, określając w ten sposób justowanie zawartości kolumny. Dosunięcie zawartości kolumny do lewej oznaczamy literą `l`, do prawej – znakiem `r`, a wyśrodkowanie — znakiem `c`. Zapis `*{3}{r}` jest równoważny zapisowi `rrr`. Wewnątrz środowiska `array` poszczególne komórki w wierszu znakiem `&` a wiersze oddzielamy instrukcją `\\`.

Opcjonalny parametr poz

Z każdym wyrażeniem matematycznym stowarzyszona jest hipotetyczna linia, nosząca nazwę osi na wysokości której umieszcza się znak minus. Parametrem `poz` określa położenie osi środowiska `array`. Dostępne są następujące jego wartości.

- `c` – wskazuje domyślne ułożenie osi (pozycja osi gdy parametr nie jest podany);
- `t` – oznacza, że oś stowarzyszona z pierwszym wierszem tablicy ma się pokrywać z osią całej tablicy;
- `b` – oznacza, że oś stowarzyszona z ostatnim wierszem tablicy ma się pokrywać z osią całej tablicy.

Przykłady tablic

Pisząc kod źródłowy

```
\[
\begin{array}{ccc}
1 & 2 & 3 \\
1 & 2 & 3 \\
1 & 2 & 3 \\
\end{array}
\]
```

otrzymujemy:

1	2	3
1	2	3
1	2	3

Przykłady tablic (cd)

Pisząc kod źródłowy

```
\[
\begin{array}{rcl}
1 & & 2 & & 3 \\
11 & & 22 & & 33 \\
111 & & 222 & & 333
\end{array}
\]
```

otrzymujemy:

1	2	3
11	22	33
111	222	333

Zastosowanie środowiska array

Pisząc

```
\[
f(x) = \left\{ \begin{array}{rcl}
5 - x^2 & \text{dla} & x \leqslant 0, \\
6 \sqrt{x} + \sin x & \text{dla} & x > 0.
\end{array} \right.
\]
```

otrzymujemy

$$f(x) = \begin{cases} 5 - x^2 & \text{dla } x \leqslant 0, \\ 6\sqrt{x} + \sin x & \text{dla } x > 0. \end{cases}$$

Siatka tabeli (cd)

Ponieważ \LaTeX nie tworzy automatycznie siatki tablicy, aby ją utworzyć musimy:

- 1 w parametrze `preambuła` wstawić znak `|`, aby sąsiednie kolumny oddzielić pojedynczą pionową linią lub znak `\|`, aby je oddzielić podwójną pionową linią;
- 2 w celu utworzenia poziomej linii użyć polecenia `\hline`, które można umieścić albo przed pierwszym wierszem tabeli, albo bezpośrednio za poleceniem `\\` kończącym wiersz. Jeżeli umieszczone jest przed pierwszym wierszem, to tworzy poziomą linię nad tabelą na całą jej szerokość. W przeciwnym razie tworzy ona poziomą linię pod wierszem, za którego zakończeniem została umieszczona. Dwa bezpośrednio po sobie umieszczone polecenia `\hline` tworzą podwójną linię.

Przykłady tablic z siatką

Pisząc kod źródłowy

```
\[  
\begin{array}{|c|c|c|}\hline  
1 & 2 & 3 \\\hline  
1 & 2 & 3 \\\hline  
1 & 2 & 3 \\\hline  
\end{array}  
\]
```

otrzymujemy:

1	2	3
1	2	3
1	2	3

Przykłady tablic z siatką (cd)

Pisząc kod źródłowy

```
\[
\begin{array}{\|r|c|l\|}\hline \hline
1 & 2 & 3 \\\hline
11 & 22 & 33 \\\hline
111 & 222 & 333 \\\hline \hline
\end{array}
\]
```

otrzymujemy:

1	2	3
11	22	33
111	222	333

Macierze

Środowisko **array** można wykorzystać do pisania macierzy.

Mianowicie, pisząc kod:

```
\left[\begin{array}{cc}
```

```
1 & 2 \\
```

```
3 & 4
```

```
\end{array}\right],
```

otrzymujemy

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix},$$

Macierze (cd)

Polecenie `\\` powoduje przejście do nowego wiersza macierzy, zaś `{cc}` oznacza centrowanie wyrażeń w kolumnach macierzy. Każda kolumna tak utworzonej macierzy może być:

- wyrównana do lewej – parametr `l`;
- wyrównana do prawej – parametr `r`;
- wycentrowana – parametr `c`

Tyle ile kolumn posiada macierz należy tyle razy wpisać odpowiednią literkę w nawiasach klamrowych, czyli macierz o pięciu kolumnach wymaga np. wyrażenia `{ccccc}` itd.

Macierze (cd)

Można oczywiście stosować inne nawiasy, albo utworzyć macierz bez nawiasów. W szczególności, wypisując kod:

```
\left|\begin{array}{cc}
```

```
1 & 2 \\
```

```
3 & 4
```

```
\end{array}\right|,
```

otrzymujemy

$$\left| \begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right|,$$

Macierze w $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX-u

Łatwiej macierze tworzy cię za pomocą środowiska `matrix` w którym jak w innych środowiskach tablicowych, wpisy specyfikowane są wiersz po wierszu, gdzie kolumny rozdzielane są ampersandem (&), a wiersze macierzy znakiem końca wiersza (\\). Mianowicie, pisząc kod:

```
\begin{matrix}
```

```
1 & 2 \\
```

```
3 & 4
```

```
\end{matrix},
```

otrzymujemy

$$\begin{matrix} 1 & 2 \\ 3 & 4' \end{matrix}$$

Macierze w $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX-u (cd)

Macierze można również zapisywać z nawiasami (separatorami) i do tego służą odpowiednie środowiska:

Środowisko	Nawias	Środowisko	Nawias
<code>pmatrix</code>	$()$	<code>pmatrix*</code>	$()$
<code>bmatrix</code>	$[]$	<code>bmatrix*</code>	$[]$
<code>Bmatrix</code>	$\{ \}$	<code>Bmatrix*</code>	$\{ \}$
<code>vmatrix</code>	$ $	<code>vmatrix*</code>	$ $
<code>Vmatrix</code>	$ $	<code>Vmatrix*</code>	$ $

Środowiska bez gwiazdki domyślnie centrują kolumny, natomiast środowiska z gwiazdą pozwalają na zdefiniowanie wyrównania kolumn poprzez opcjonalny parametr.

Macierze w $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX-u

Przykładowo pisząc kod

```
\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad
\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad
\begin{Bmatrix} 0 & 1 \\ 1 & 0 \end{Bmatrix} \quad
\begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} \quad
\begin{Vmatrix} 0 & 1 \\ 1 & 0 \end{Vmatrix} \quad
```

otrzymujemy

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{Bmatrix} 0 & 1 \\ 1 & 0 \end{Bmatrix} \quad \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} \quad \begin{Vmatrix} 0 & 1 \\ 1 & 0 \end{Vmatrix}$$

Macierze w $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX-u (cd)

Ciekawym poleceniem z `amsmath` jest `smallmatrix`, dzięki któremu miniaturowe macierze można umieszczać w tekście.

Przykładowo, pisząc kod:

Ala ma macierz $\left(\begin{smallmatrix}$

`a & b\\`

`c & d`

`\end{smallmatrix}\right) i kota.`

otrzymujemy

Ala ma macierz $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ i kota.

Przegląd środowisk wyświetlających równania

Pakiet `amsmath` oferuje rozszerzony w stosunku do podstawowego \LaTeX -a zbiór struktur wyświetlających równania. W jego skład wchodzi:

- `equation` i `equation*`
- `align` i `align*`
- `gather` i `gather*`
- `flalign` i `flalign*`
- `multline` i `multline*`
- `alignat` i `alignat*`
- `split`

Przegląd środowisk wyświetlających równania (cd)

Oprócz `split` każde środowisko ma dwie wersje – z gwiazdka i bez, przy czym te bez gwiazdki posiadają automatyczne numerowanie wykorzystujące *\LaTeX equation counter*.
`Split` to podrzędne środowisko wykorzystywane jedynie w innych wspomnianych wcześniej (za wyjątkiem `multline`).

Środowiska equation i equation*

Środowiska equation i equation* służą do prezentacji pojedynczych wyrażeń (odpowiednio z autonumeracją i bez). Środowiska equation* jest równoważne użyciu `\[...\]`.

Pisząc

```
\begin{equation}
\eta \leq C(\delta(\eta) + \Lambda M(0, \delta))
\end{equation}
```

otrzymujemy

$$\eta \leq C(\delta(\eta) + \Lambda M(0, \delta)) \quad (1)$$

Środowisko `multline` i `multline*`

Środowisko `multline` i `multline*` jest odmiana `equation` wykorzystywana w przypadku, gdy całe wyrażenie nie mieści się w pojedynczej linii. Pierwsza linia `multline` jest wyrównana do lewej strony, ostatnia do prawej. Podobnie do `equation`, `multline` jest numerowane (całemu wyrażeniu kilku liniom – oczywiście jeden numer). W zależności od wybranej opcji pakietu `amsmath` numer wyrażenia może być przy pierwszej linii (opcja `leqno`) lub przy ostatniej (opcja `reqno`).

Środowisko `multline` i `multline*` (cd)

Pisząc

```
\begin{multline}  
\text{Pierwsza linia} \\  
\text{Jeszcze jedna} \\  
\text{I jeszcze jedna linia} \\  
\text{A ta jest ostatnia}  
\end{multline}
```

otrzymujemy

Pierwsza linia

Jeszcze jedna

I jeszcze jedna linia

A ta jest ostatnia (2)



Środowisko `multline` i `multline*` (cd)

Możliwe jest wymuszenie wyrównania każdej z linii pomiędzy pierwszą a ostatnią za pomocą komend `\shoveleft{< jedna z środkowych linii >}` i `\shoveright{< analogicznie >}`.

Środowisko `multline` i `multline*` (cd)

Pisząc

```
\begin{multline}  
\text{Pierwsza linia} \\\br/>\text{Jeszcze jedna} \\\br/>\shoveright{\text{I jeszcze jedna linia}} \\\br/>\text{A ta jest ostatnia}  
\end{multline}
```

otrzymujemy

Pierwsza linia

Jeszcze jedna

I jeszcze jedna linia

A ta jest ostatnia (3)



Środowisko split

Podobnie jak `multline`, `split` używany jest, równanie nie mieści się w jednej linii. Jednak w przypadku `split` dodatkowo mamy wyrównanie pomiędzy liniami, a punkty względem których ma nastąpić wyrównanie oznaczane jest za pomocą `&`. W odróżnieniu od innych środowisk, `split` nie prowadzi numeracji wzorów – jak już wspomnieliśmy `split` używany jest jako podśrodowisko, najczęściej w `equation`, `align`, czy też `gather`, w których to już autonumeracja występuje.

Środowisko split (cd)

Pisząc

```
\begin{equation}
\begin{split}
1024 &= 2 \cdot 512 \\
&= 2^2 \cdot 256 \\
&= 2^4 \cdot 128
\end{split}
\end{equation}
```

otrzymujemy

$$\begin{aligned}
 1024 &= 2 \cdot 512 \\
 &= 2^2 \cdot 256 \\
 &= 2^4 \cdot 128
 \end{aligned} \tag{4}$$

Środowisko gather i gather*

Środowiska gather i gather* wykorzystywane są dla zgrupowanych, następujących po sobie wzorów, każdy z nich jest wyrównywany do środka względem szerokości tekstu. Kolejne wzory oddzielane są za pomocą „\\”. O ile zachodzi taka potrzeba w gather można użyć split. Pisząc kod

```
\begin{gather}
(a+b)^{2}=a^{2}+2ab+b^{2}\\
(a+b)\cdot (a-b)=a^{2}-b^{2}
\end{gather}
```

otrzymujemy

$$(a + b)^2 = a^2 + 2ab + b^2 \quad (5)$$

$$(a + b) \cdot (a - b) = a^2 - b^2 \quad (6)$$

Środowisko gather i gather* (cd)

Pisząc kod

```
\begin{gather}
H_{1}=\frac{1}{2n} \\\
\begin{split}
H_{2}=\frac{1}{2n}\sum_{n=1}^{\infty}(-1)^{n-1}&\\
(n-1)^{p-2}\sum_{n=1}^{\infty}&\\
\prod_{i=1}^p\binom{n-i}{1-i}&\\
&\quad\cdot[(n-1)-(n-i-1-i)]^{n-i-1-i}\\
\end{split} \\\
H_{3}=(n-1)^2-\sum_{j=1}^p(n-i-1-i)^2\\
\end{gather}
```

Środowisko gather i gather* (cd)

otrzymujemy

$$H_1 = \frac{1}{2n} \tag{7}$$

$$H_2 = \frac{1}{2n} \sum_{l=0}^n (-1)^l (n-l)^{p-2} \sum_{l_1=l}^n \prod_{i=1}^p \binom{n_i}{l_i} \cdot [(n-l) - (n_i - l_i)]^{n_i - l_i} \tag{8}$$

$$H_3 = (n-l)^2 - \sum_{j=1}^p (n_i - l_i)^2 \tag{9}$$

Środowisko align i align*

W przypadku, gdy więcej niż jedno wyrażenie chcemy umieścić w jednej lub kilku liniach i potrzebne jest do tego wyrównanie w pionie, można skorzystać z środowisk `align` i `align*`. Kolejne linie oddziela się od siebie za pomocą „\\”, punkty względem których ma nastąpić wyrównanie zaznacza się za pomocą `&`.

Środowisko align i align* (cd)

Pisząc

```
\begin{align}
f_{1}(x) &= ax^2 + bx + c & f_{2}(x) &= 3x + 2 \\
f_{111}(x) &= \log 2x & f_{2222}(x) &= \pi x^2 \\
f_{11}(x) &= f_{1}(x) & f_{22}(x) &= f_{2}'(x)
\end{align}
```

otrzymujemy

$$f_1(x) = ax^2 + bx + c \qquad f_2(x) = 3x + 2 \qquad (10)$$

$$f_{111}(x) = \log 2x \qquad f_{2222}(x) = \pi x^2 \qquad (11)$$

$$f_{11}(x) = f_1(x) \qquad f_{22}(x) = f_2'(x) \qquad (12)$$

Środowisko align i align* (cd)

Oczywiście, otocznia align i align* można stosować również w przypadku, gdy w każdej linii umieszczamy po jednym wyrażeniu które chcemy wyrównać w pionie. Pisząc

```
\begin{align}
a_{11} &= b_{11} \\
a_{21} &= b_{21} + c_{11}
\end{align}
```

otrzymujemy

$$a_{11} = b_{11} \tag{13}$$

$$a_{21} = b_{21} + c_{11} \tag{14}$$

Środowisko `flalign` i `flalign*`

Przypominają one środowiska `align` i `align*`, ale tworzone przez nie kolumny wyrażeń rozmieszczone są równomiernie na całej szerokości strony, z tym, że pierwsza kolumna jest przysunięta maksymalnie do lewego marginesu, a ostatnia maksymalnie do prawego marginesu. W przypadku jednej kolumny środowiska te zachowują się tak samo jak `align` i `align*`.

Środowisko flalign i flalign* (cd)

Pisząc kod

```
\begin{flalign}
f_{1}(x) &= ax^2 + bx + c && f_{2}(x) = 3x + 2 \\
f_{111}(x) &= \log 2x && f_{2222}(x) = \pi x^2 \\
f_{11} &= f_{1}(x) && f_{22}(x) = f_{2}'(x)
\end{flalign}
```

otrzymujemy

$$f_1(x) = ax^2 + bx + c$$

$$f_{111}(x) = \log 2x$$

$$f_{11} = f_1(x)$$

$$f_2(x) = 3x + 2 \quad (15)$$

$$f_{2222}(x) = \pi x^2 \quad (16)$$

$$f_{22}(x) = f_2'(x) \quad (17)$$

Środowisko flalign i flalign* (cd)

Pisząc kod

```
\begin{flalign}
a_{11} &= b_{11} \\
a_{21} &= b_{21} + c_{11}
\end{flalign}
```

otrzymujemy

$$a_{11} = b_{11} \tag{18}$$

$$a_{21} = b_{21} + c_{11} \tag{19}$$

Środowisko cases

Środowisko `cases` jest podrzędnym środowiskiem wykorzystywane w innych wspomnianych wcześniej (za wyjątkiem `multline`). Służy ono do definiowania przez przypadki. Można go używać zamiast polecenia `array` z \LaTeX -a.

Środowisko cases

Pisząc kod

```
\{equation}
P_{i}=
\begin{cases}
0 & \text{gdy } i=2k \\
n! & \text{gdy } i=2k+1
\end{cases}
\{equation}
```

otrzymujemy

$$P_i = \begin{cases} 0 & \text{gdy } i = 2k \\ n! & \text{gdy } i = 2k + 1 \end{cases} \quad (20)$$

Odnosniki do równań

Również w czystym L^AT_EX-u do numerów równań, jak również rysunków i tabel można się w tekście odwoływać za pomocą odnośników.

Pisząc

```
\begin{equation} \label{refex}
```

$$e^{i\pi} + 1 = 0$$

```
\end{equation}
```

Równanie `\ref{refex}` zawiera pięć najważniejszych liczb w matematyce.

otrzymujemy

$$e^{i\pi} + 1 = 0 \tag{21}$$

Równanie 21 zawiera pięć najważniejszych liczb w matematyce.

Numerowanie wyrażeń

Aby wyłączyć numerację linii w środowisku posiadającym automatyczną numerację (środowisko bez gwiazdki) wystarczy umieścić w niej polecenie `\notag`. Do definiowania własnego symbolu (dowolny tekst, w tym liczba) służy polecenie `\tag{etykieta}`. Gdy zastosujemy polecenie `\tag*{etykieta}`, to otrzymamy własny symbol bez nawiasów wokół niego.

Numerowanie wyrażeń (cd)

Pisząc kod

```
\begin{equation}\tag{etykieta}
\eta \leq C(\delta(\eta)+\Lambda M(0,\delta))
\end{equation}
```

otrzymujemy

$$\eta \leq C(\delta(\eta) + \Lambda M(0, \delta)) \quad (\text{etykieta})$$

Numerowanie wyrażeń (cd)

Pisząc kod:

```
\begin{gather}
2+2=4 \\
3+3=6 \notag \\
5+5=10 \tag*{$\bigstar$}
2+2=5 \tag{sic!} \\
\end{gather}
```

otrzymujemy

$$2 + 2 = 4 \tag{22}$$

$$3 + 3 = 6$$

$$5 + 5 = 10 \tag{\bigstar}$$

$$2 + 2 = 5 \tag{sic!}$$

Numerowanie wyrażeń (cd)

L^AT_EX dostarcza narzędzia, dzięki któremu każde z wyrażeń może być numerowane osobno w swoim paragrafie (w pierwszym 1.1, 1.2, w drugim 2.1 itd.) - wystarczy zastosować polecenie:

```
\renewcommand{\theequation}
{\thesection.\arabic{equation}}.
```

Należy jednak wtedy pamiętać, aby na początku nowego rozdziału wyzerować licznik (`\setcounter`). Dzięki `amsmath` problem ten znika – należy użyć tylko polecenia

```
\numberwithin{equation}{section}.
```

Polecenie `\numberwithin` nie musi się odnosić jedynie do numerowania wyrażeń – może do każdego licznika).

Numerowanie wyrażeń. Środowisko subequations

Dodatkowo pakiet `amsmath` udostępnia dodatkową numerację wyrażeń: środowisko `subequations`, które pozwala składać serie wzorów matematycznych, z możliwością odwoływania się do nich jako do całości lub do indywidualnych wzorów.

Wyrażenia znajdujące się w środowisku `subequations` są numerowane z indeksem wyższym od poprzedniego o 1 i kolejnymi literkami (a, b, c itd). Ma przykład, jeśli poprzednie równanie miało numer (6.1), to użycie następnie środowiska `subequations` dla trzech kolejnych równań nada im numery (6.2a), (6.2b), (6.2c).

Środowisko subequations (cd)

Pisząc kod

```
\begin{subequations}\tag{ltxeq1}
\begin{align}
f(ax)&=af(x)\label{ltxeq1a}\\
f(x+y)&=f(x)+f(y)\label{ltxeq1b}
\end{align}
\end{subequations}
```

otrzymujemy

$$f(ax) = af(x) \tag{23a}$$

$$f(x + y) = f(x) + f(y) \tag{23b}$$

Środowisko subequations (cd)

Cytujemy wzory ze środowiska `textttsubequations` następująco.

Pisząc

Ze wzorów `\ref{ltxeq1}` wynika, że funkcja f jest liniowa. Własność jednorodności funkcji f podana jest we wzorze `\ref{ltxeq1a}`, a addytywność we wzorze `\ref{ltxeq1b}`.

otrzymujemy

Ze wzorów (23) wynika, że funkcja f jest liniowa. Własność jednorodności funkcji f podana jest we wzorze (23a), a addytywność we wzorze (23b).

Środowisko `eqnarray` i `eqnarray*`

Środowiska `eqnarray` i `eqnarray*` pozwalają składać serie wzorów matematycznych. Symbole `&` dzielą linię wzoru na części wyrównane odpowiednio: do prawej, do środka i do lewej.

Środowisko eqnarray i eqnarray*

Pisząc kod:

`$f\colon Act \to Act$` jest `\emph{funkcją przeetykietowującą}` spełniającą warunki (`\ref{eq:tau1}`) i (`\ref{eq:tau2}`):

```
\begin{eqnarray}
f(\tau) \&= \& \tau, \label{eq:tau1} \\
f(\bar{a}) \&= \& \overline{f(a)} \label{eq:tau2} \\
\text{ dla dowolnej etykiety } a \text{ in } Act.
\end{eqnarray}
```


Środowisko eqnarray i eqnarray* (cd)

otrzymujemy

$f: Act \rightarrow Act$ jest *funkcją przeetykietowującą* spełniającą warunki (24) i (25):

$$f(\tau) = \tau, \quad (24)$$

$$f(\bar{a}) = \overline{f(a)} \text{ dla dowolnej etykiety } a \in Act. \quad (25)$$