

Skład dokumentów w systemie \LaTeX

Marcin Szyrka

Katedra Informatyki Stosowanej
AGH w Krakowie

2018/2019

Literatura

1. Tobias Oetiker et al: **Nie za krótkie wprowadzeni do systemu \LaTeX 2e**. Tłumaczenie na język polski Tomasz Przechlewski, Ryszard Kubiak, 2007
2. Diller A.: **\LaTeX Wiersz po wierszu**, Wydawnictwo Helion, Gliwice, 2000
3. Manuale do poszczególnych pakietów wskazane na stronie www wykładu.

- L^AT_EX jest systemem składu umożliwiającym tworzenie dowolnego typu dokumentów (w szczególności naukowych i technicznych) o wysokiej jakości typograficznej.
- Wysoka jakość składu jest niezależna od rozmiaru dokumentu – zaczynając od krótkich listów do bardzo grubych książek.
- L^AT_EX automatyzuje wiele prac związanych ze składaniem dokumentów np.: referencje, cytowania, generowanie spisów (treści, rysunków, symboli itp.) itd.
- L^AT_EX jest zestawem instrukcji umożliwiających autorom skład i wydruk ich prac na najwyższym poziomie typograficznym. Do formatowania dokumentu L^AT_EX stosuje T_EXa (wymawiamy 'tech' – greckie litery τ, ε, χ).
- Dokumenty złożone w L^AT_EXu cechuje determinizm – uzyskamy ten sam efekt niezależnie od systemu operacyjnego pod którym odbywa się kompilacja, czy też drukarki użytej do drukowania dokumentów.
- Plik L^AT_EXowy jest plikiem tekstowym, który oprócz tekstu zawiera polecenia formatujące ten tekst (analogicznie do języka HTML).
- Wynikiem kompilacji może być plik DVI (*DeVice Independent*) lub plik pdf.

Struktura prostego dokumentu

```
1 \documentclass[a4paper,12pt]{article}           % preambuła
2 \usepackage[polish]{babel}
3 \usepackage[latin2]{inputenc}                 % utf8, cp1250
4 \usepackage[T1]{fontenc}
5 \usepackage{times}
6
7 \begin{document}                               % część główna
8
9 \section{Sztuczne życie}
10
11 W 1987 roku na konferencji naukowej w Nowym Meksyku w Stanach
12 Zjednoczonych narodziła się kolejna dziedzina sztucznej inteligencji
13 nazwana sztucznym życiem (artificial life). Zajmuje się ona
14 tworzeniem systemów, które mogą się samodzielnie rozwijać
15 i doskonalić.
16
17 Już w 1968 roku Aristin Lindenmaier podjął próbę stworzenia
18 uniwersalnego języka genetycznego używanego przez rośliny
19 i w rezultacie rozwinął algorytmy odtwarzające strukturę roślin.
20 W~wyniku tych prac powstał matematyczny opis wzrostu roślin,
21 na część naukowca nazwany L-systemem.
22 \end{document}
```

demo.tex, ptaki.tex, ptaki_amcs.tex, ptaki_fi.tex

latex

```
latex test.tex
dvips test.dvi -o test.ps
ps2pdf test.ps
```

pdflatex

```
pdflatex test.tex
```

- Przy pierwszej kompilacji po zmianie tekstu, dodaniu nowych etykiet itp., \LaTeX tworzy sobie spis rozdziałów, obrazków, tabel itp., a dopiero przy następnej kompilacji korzysta z tych informacji.
- W pierwszym przypadku rysunki powinny być przygotowane w formacie eps, a w drugim w formacie pdf. Ponadto, jeżeli używamy polecenia **pdflatex**, to do dokumentu można wstawiać grafikę bitową (np. w formacie jpg).

Podstawowe zasady przygotowania pliku źródłowego

- Poszczególne słowa oddzielamy spacjami, przy czym ilość spacji nie ma znaczenia. Po kompilacji wielokrotne spacje i tak będą wyglądały jak pojedyncza spacja.
- Aby uzyskać **twardą spację**, zamiast znaku spacji należy użyć znaku **tyldy**.
- Znakiem końca akapitu jest pusta linia (ilość pustych linii nie ma znaczenia), a nie znaki przejścia do nowej linii.
- Znaki \$ & % # _ { } ~ ^ \ mają specjalne znaczenie i nie można ich wstawić bezpośrednio do pliku – w ośmiu pierwszych przypadkach należy je poprzedzić znakiem \.
- \LaTeX sam formatuje tekst. **Nie starajmy się go poprawiać**, chyba, że naprawdę wiemy co robimy.
- Wszystkie kwestie dotyczące łamania linii tekstu, akapitów, stron, itp. są rozstrzygane na etapie kompilacji i później wygląd dokumentu nie ulega zmianie.
- Opcje decydujące o wyglądzie dokumentu po kompilacji grupowane są w tzw. **klasach dokumentów**. Większość wydawnictw naukowych na świecie dostarcza własne klasy dokumentów dla \LaTeX a, po użyciu których dokumenty są przygotowane do publikacji w tych wydawnictwach.

- **TeX Live** – dystrybucja TeXa dostępna we wszystkich popularnych wydaniach Linuksa.
- **TeXstudio** – zaawansowane zintegrowane środowisko do składu dokumentów w L^AT_EXu.
- <https://www.overleaf.com> – środowisko on-line, wystarczy przeglądarka www, nie trzeba instalować!
- **PGF/TikZ, QTikZ, dia, gimp** – narzędzia do przygotowania grafiki.
- **latexdiff** – porównywanie różnych wersji tego samego dokumentu.

Formatowanie tekstu

```
1 zwykły tekst, zwykły tekst, \emph{tekst wyróżniony},
2 zwykły tekst, \textbf{tekst pogrubiony}, zwykły tekst,
3 \texttt{czcionka maszynowa}, \textit{kursywa, kursywa,
4 {\em wyróżnienie w tekście pisanym kursywą},
5 kursywa, kursywa}, \textsc{kapitaliki}, \textsf{krój bezszeryfowy}
```

zwykły tekst, zwykły tekst, *tekst wyróżniony*, zwykły tekst, **tekst pogrubiony**, zwykły tekst, `czcionka maszynowa`, *kursywa, kursywa*, wyróżnienie w tekście pisanym kursywą, *kursywa, kursywa*, KAPITALIKI, krój bezszeryfowy

```
6 {\small mała czcionka}, {\large duża czcionka},
7 normalna wielkość czcionki, {\footnotesize rozmiar typowy
8 dla stopki}, {\Large bardzo duża czcionka},
9 \textit{\LARGE ,,jeszcze większa'' czcionka w połączeniu
10 z kursywą} {\huge i jeszcze większa} {\Huge i jeszcze większa}
```

mała czcionka, duża czcionka, normalna wielkość czcionki, rozmiar typowy dla stopki, bardzo duża czcionka, „*jeszcze większa*” *czcionka w połączeniu z kursywą* i jeszcze większa i jeszcze większa

Środowiska

Wiele instrukcji L^AT_EXa to **środowiska** mające postać:

```
1 \begin{nazwa-polecenia}
2 tekst
3 \end{nazwa-polecenia}

4 \begin{flushleft}
5 Akapit wyrównany do lewej strony.
6 \end{flushleft}
7
8 \begin{center}
9 Akapit wyśrodkowany.
10 \end{center}
11
12 \begin{flushright}
13 Akapit wyrównany do prawej strony.
14 \end{flushright}
```

Akapit wyrównany do lewej strony.

Akapit wyśrodkowany.

Akapit wyrównany do prawej strony.

Wypunktowanie

```
1 Do tej pory powstało wiele form sztucznego życia.
2 Zalicza się do nich np.:
3
4 \begin{itemize}
5 \item pętle -- rozmnażające się litery,
6 \item boidy -- istoty podobne do ptaków,
7 \item animki -- kwadraty poszukujące pokarmu,
8 \item bimorfy -- rozmnażające się kształty,
9 \item L-systemy -- sztuczne kwiaty.
10 \end{itemize}
```

Do tej pory powstało wiele form sztucznego życia. Zalicza się do nich np.:

- pętle – rozmnażające się litery,
- boidy – istoty podobne do ptaków,
- animki – kwadraty poszukujące pokarmu,
- bimorfy – rozmnażające się kształty,
- L-systemy – sztuczne kwiaty.

Numerowanie

```
1 Do tej pory powstało wiele form sztucznego życia.
2 Zalicza się do nich np.:
3
4 \begin{enumerate}
5 \item pętle -- rozmnażające się litery,
6 \item boidy -- istoty podobne do ptaków,
7 \item animki -- kwadraty poszukujące pokarmu,
8 \item bimorfy -- rozmnażające się kształty,
9 \item L-systemy -- sztuczne kwiaty.
10 \end{enumerate}
```

Do tej pory powstało wiele form sztucznego życia. Zalicza się do nich np.:

1. pętle – rozmnażające się litery,
2. boidy – istoty podobne do ptaków,
3. animki – kwadraty poszukujące pokarmu,
4. bimorfy – rozmnażające się kształty,
5. L-systemy – sztuczne kwiaty.

Lista pojęć

```
1 Do tej pory powstało wiele form sztucznego życia.
2 Zalicza się do nich np.:
3
4 \begin{description}
5 \item[pętle] -- rozmnażające się litery,
6 \item[boidy] -- istoty podobne do ptaków,
7 \item[animki] -- kwadraty poszukujące pokarmu,
8 \item[bimorfy] -- rozmnażające się kształty,
9 \item[L-systemy] -- sztuczne kwiaty.
10 \end{description}
```

Do tej pory powstało wiele form sztucznego życia. Zalicza się do nich np.:

- pętle** – rozmnażające się litery,
- boidy** – istoty podobne do ptaków,
- animki** – kwadraty poszukujące pokarmu,
- bimorfy** – rozmnażające się kształty,
- L-systemy** – sztuczne kwiaty.

- Środowiska `itemize`, `enumerate` i `description` można zagnieżdżać tworząc wielopoziomowe wypunktowania, wyliczenia itp.
- Można indywidualnie zmienić symbol dla danego punktu w środowiskach `itemize` i `enumerate` podając go w nawiasie kwadratowym po poleceniu `item`, np.:

```
1 \item[--] pętle -- rozmnażające się litery,  
2 \item[ $\spadesuit$ ] boidy -- istoty podobne do ptaków,
```

- Można sterować odstępem między punktami stosując polecenie `setlength`

```
3 \begin{enumerate}  
4 \setlength{\itemsep}{2mm}
```

- Redefiniując `labelitemi`, `labelitemii`, `labelitemiii` i `labelitemiv` można ustalić znaki używane na różnych poziomach wypunktowania, np.:

```
5 \renewcommand{\labelitemi}{ $\star$ }
```

- Zastosowanie pakietu `enumerate` pozwala w bardzo wygodny sposób ustalać formę numerowania, np.:

```
6 \begin{enumerate} [ (i) ]
```

wyliczenia.tex

Standardowe klasy dokumentów

- `article` – artykuły, krótkie opracowania, ...
- `report` – dłuższe opracowania, prace, dyplomowe, ...
- `book` – książki,
- `letter` – listy.

Opcje klas dokumentów:

`10pt`, `11pt`, `12pt` rozmiary podstawowego tekstu w dokumencie,

`a4paper`, `b5paper`, ... rozmiar papieru,

`fleqn` eksponowane wzory matematyczne dosuwane do lewej strony (zamiast centrowania),

`titlepage`, `notitlepage` umieszczanie tytułu (lub nie) na oddzielnej stronie,

`onecolumn`, `twocolumn` skład jedno- lub dwukolumnowy,

`oneside`, `twoside` wydruk jedno- lub dwustronny,

`openright`, `openany` tytuły rozdziałów umieszczane na stronie nieparzystej lub dowolnej.

Uwaga: Każda z klas ma zdefiniowane wartości domyślne dla podanych opcji.

```
\documentclass[a4paper,11pt,twoside,fleqn,openany]{book}
```

- `\`, `\newline` – łamanie linii bez rozpoczynania nowego akapitu,
- `*` – j.w., ale z zakazem łamania strony w miejscu złamania linii,
- `\newpage` – rozpoczęcie nowej strony,
- `\linebreak`_[n] – zachęta do łamania wiersza – parametr *n* przyjmuje wartości od 0 do 4 i określa stopień zachęty lub niezgody, domyślna wartość 4 to bezwarunkowy zakaz lub nakaz,
- `\nolinebreak`_[n] – niezgoda na łamanie wiersza,
- `\pagebreak`_[n] – zachęta do łamania strony,
- `\nopagebreak`_[n] – niezgoda na łamanie strony,
- `\clearpage` – rozpoczęcie składu nowej strony,
- `\cleardoublepage` – rozpoczęcie składu nowej strony od strony o numerze nieparzystym.
- `\sloppy` – obniżenie domyślnych standardów składu tekstu – linie nie będą wchodzić na marginesy, ale mogą być zwiększone odstępy między słowami,
- `\fussy` – przywrócenie domyślnych parametrów składu (przeciwnieństwo `\sloppy`).

Składanie tekstu – dodatki

- W przypadku nieprawidłowego dzielenia słowa, można wskazać miejsce podziału za pomocą sekwencji `\-`. Można również zastosować w preambule polecenie `\hyphenation`{sło\ -wo, dłu\ -giesło\ -wo, ...} Słowa z listy argumentów można dzielić wyłącznie we wskazanych miejscach.
- `\mbox`{tekst} – wskazany tekst nie będzie łamany między linie;
- ```angielskie''` – cudzysłowy “angielskie”;
- `,,polskie''` – cudzysłowy „polskie” (2 przecinki + 2 apostrofy),
- `<<francuskie>>` – cudzysłowy «francuskie» (przy użyciu cudzysłowu w tekście już objętym cudzysłowem);
- – **dywiz**, np. niebiesko-czarny – W przypadku dzielenia na łączniku, dywiz należy powtórzyć na początku nowej linii;
- -- **półpauza**, np. str. 11–13 (bez odstępów przed i po półpauzie);
- --- **pauza**, np. stosowana jako znak przestankowy (można też użyć w takim miejscu półpauzy jak na tym slajdzie);
- `W~domu` – **twarda spacja**;
- `\sim` – **tylda** do stosowania np. w adresach internetowych;
- Adresy internetowe można zapisywać użyciem polecenia `url` po dodaniu w preambule pakietu o takiej nazwie:
`\url{http://home.agh.edu.pl/~mszpyrka}`
`http://home.agh.edu.pl/~mszpyrka`

Klasa article (1)

```
1 \documentclass[a4paper,10pt]{article}
2 \usepackage[polish]{babel}
3 % ...
4
5 \title{Ptaki żyjące w Polsce}
6
7 \author{Ferdynand Wspaniały\\
8 \small Katedra Bocianoznawstwa\\
9 \small Uniwersytet Ornitologiczny w Krakowie\\
10 \small \texttt{ferdynand@uo.edu.pl}}
11
12 \date{02.03.2013}
13
14 \begin{document}
15 \maketitle
16
17 \begin{abstract}
18 Niniejszy artykuł o~ptakach ma charakter popularny. ...
19 \end{abstract}
20
21 \section{Wprowadzenie}
22 \label{sec:wprowadzenie}
23
24 % Tekst w pierwszej sekcji
```

Klasa article (2)

```
1 \section{Wprowadzenie}
2 \label{sec:wprowadzenie}
3
4 % Tekst w pierwszej sekcji
5
6 \section{Rodzina krukowatych}
7 \label{sec:rodzinakrukowatych}
8
9 % Tekst w drugiej sekcji
10
11 \subsection{Kruk}
12 \label{ssec:kruk}
13
14 % Tekst w podsekcji
15
16 \subsection{Gawron}
17 \label{ssec:gawron}
18
19 % Tekst w podsekcji
20
21 % ...
```

Klasy report i book

```
1 \documentclass[a4paper,10pt]{report}
2 % Dalej preambuła jak dla article
3
4 \chapter{Wprowadzenie}
5 \label{cha:wprowadzenie}
6
7 % Tekst pierwszego rozdziału
8
9 \chapter{Rodzina krukowatych}
10 \label{cha:rodzinakrukowatych}
11
12 % Tekst w drugiego rozdziału
13
14 \section{Kruk}
15 \label{sec:kruk}
16
17 % Tekst podrozdziału
18
19 \section{Gawron}
20 \label{sec:gawron}
21
22 % Tekst podrozdziału
```

Uwaga: Klasę `book` używamy podobnie jak `article`, ale dla klasy `book` nie jest dostępne środowisko `abstract`.

Duże dokumenty

Przy pracy nad dużym dokumentem (książka, praca dyplomowa) warto podzielić źródła na mniejsze części, np. każdy rozdział w oddzielnym pliku.

- `\include{plik}` – dołącza do dokumentu zawartość pliku wstawiając przed i po łamanie strony. Uwaga: **Pomijamy rozszerzenie `tex`**.
- `\input{plik}` – dołącza do dokumentu zawartość pliku (bez dodatkowego łamania strony).

Plik główny (klasa `report` lub `book`):

```
1 \begin{document}          %Preambuła
2 \maketitle
3 \tableofcontents
4 \clearpage
5
6 \include{rozdzial1}
7 \include{rozdzial2} % ...
8 \appendix
9 % Od tego miejsca rozdziały są traktowane jako dodatki, A, B, ...
10 \include{dodatekA}
11 \include{dodatekB} % ...
12
13 \bibliography{bibfile}
14 \end{document}
```

Tabele

Podstawowym środowiskiem do tworzenia tabel w \LaTeX u jest `tabular`. Jeżeli chcemy do tabeli dodać napis i numer, to należy umieścić środowisko `tabular` wewnątrz środowiska `table`.

```
1 \begin{table}
2 \caption{Stany sygnalizatora}
3 \label{tab:stanySygnalizatora}
4 \begin{center}
5 \begin{tabular}{|c||l|l|l|l|}
6 \hline Stan & S1 & S2 & S3 & S4 \\ \hline \hline
7 1 & zielone & zielone & czerwone & czerwone \\ \hline
8 2 & czerwone & zielone & zielone & czerwone \\ \hline
9 3 & czerwone & czerwone & czerwone & zielone \\ \hline
10 \end{tabular}
11 \end{center}
12 \end{table}
```

Tablica: Stany sygnalizatora

Stan	S1	S2	S3	S4
1	zielone	zielone	czerwone	czerwone
2	czerwone	zielone	zielone	czerwone
3	czerwone	czerwone	czerwone	zielone

Wstawki

Elementy takie jak tabele i rysunki są umieszczane w postaci tzw. **wstawek**. Wstawki są tworzone za pomocą środowisk `figure` (rysunki) i `table` (tabele). Jeżeli wstawka nie mieści się na danej stronie, \LaTeX automatycznie przenosi ją na kolejną stronę uzupełniając pozostałe miejsce tekstem. \LaTeX przechowuje wstawki w dwóch kolejkach FIFO (rysunki i tabele) i na nowej stronie próbuje umieszczać kolejne wstawki, jeśli takie są w kolejkach. Jeśli czegoś nie można zmieścić, jest to przesuwane na kolejną stronę. Może się zdarzyć, że pojedynczy rysunek, którego nie można z jakichś względów poprawnie wstawić przesunie całą kolejkę na koniec dokumentu.

Parametry: `\begin{table}` [. . .]

- `h` – bez przemieszczania (dokładnie w miejscu użycia),
- `t` – na górze strony,
- `b` – na dole strony,
- `p` – na stronie zawierającej wyłącznie wstawki,
- `!` – pominięcie większości parametrów sterujący, „próba wymuszenia”.

Instrukcje `\listoffigures` i `\listoftables` wstawiają do dokumentu odpowiednio spis rysunków i spis tabel.

`ptaki3.tex`

Zarządzanie odstępem między kolumnami

Odstęp między kolumnami w tabeli można zdefiniować za pomocą polecenia `@{...}`. Zastępuje ona odstęp międzykolumnowy treścią umieszczoną między klamrami.

```
1 \begin{center}
2 \begin{tabular}
3 { |@{}c|@{~}c|@{\hspace{4mm}}c|@{}c@{}|@{\ldots}c@{abcd}|c| }
4 \hline
5 X & X & X & X & X & X & X \\
6 X & X & X & X & X & X & X \\
7 X & X & X & X & X & X & X \\
8 X & X & X & X & X & X & X \\
9 \hline
10 \end{tabular}
11 \end{center}
```

X	X	X	X	...Xabcd	X
X	X	X	X	...Xabcd	X
X	X	X	X	...Xabcd	X
X	X	X	X	...Xabcd	X

Polecenia `multicolumn` i `cline`

Polecenie `multicolumn` służy do łączenia komórek w ramach danego wiersza. Polecenie `cline` rysuje poziomą linię podobnie jak `hline`, ale tylko w zakresie wskazanych kolumn.

```
1 \begin{table}[!htb]
2 \caption{Warunki terenowe dla utwierdzenia przebiegu}
3 \label{tab:warunkiTerenowe}
4 \begin{tabular}{|c|c|c|c|c|c|c|c|}
5 \cline{2-8}
6 \multicolumn{1}{|l|}{} & \multicolumn{7}{c|}{Zwrotnice} \\
7 Przebiegi & 3/4 & 5 & 6 & 7/8 & 15/16 & 17 & 18 \\
8 B1 & + & + & & & & & \\
9 B2 & - & & + & o+ & & & \\
10 \end{tabular}
11 \end{table}
```

Tablica: Warunki terenowe dla utwierdzenia przebiegu

Przebiegi	Zwrotnice						
	3/4	5	6	7/8	15/16	17	18
B1	+	+					
B2	-		+	o+			

Pakiet `makecell`

Pakiet `makecell` pozwala na modyfikowanie wyglądu komórek tabeli, oferując m.in.: łamanie linii w komórce, sterowanie wyrównaniem tekstu w pionie i poziomie, sterowanie odstępami wokół zawartości komórki, składanie komórek zajmujących wiele wierszy, składanie komórek dzielonych diagonalnie itp.

```
1 \begin{tabular}{|l|l|l|l|}
2 \hline
3 Rodzina & Gatunek & Wychów piskląt \\ \hline
4 Rodzina krukowatych & Gawron & Pisklęta są gniazdownikami \\ \hline
5 Rodzina jaskółkowatych & Jaskółka oknówka & \\
6 \makecell[l]{Pisklęta są rzekomymi} \\ \hline
7 \end{tabular}
```

wyrównanie w pionie (tcb) i/lub poziomie
(lcrp), np. `tr, {}{p{3cm}}`

możliwość łamania linii

`makecell_demo.tex`

Rodzina	Gatunek	Wychów piskląt
Rodzina krukowatych	Gawron	Pisklęta są gniazdownikami
Rodzina jaskółkowatych	Jaskółka oknówka	Pisklęta są rzekomymi gniazdownikami

Formatowanie nagłówka tabeli

- Polecenie `thead` działa analogicznie jak `makecell`, ale jest używane do definiowania komórek w nagłówku tabeli. Oba polecenia mają wersję `*`, której użycie powoduje zastosowanie dodatkowych odstępów wokół komórki.
- Pakiet pozwala na redefiniowanie domyślnego sposobu składania nagłówków tabel:

```
\renewcommand\theadalign{t1}
\renewcommand\theadfont{\bfseries\normalsize}
```

- Polecenie `rothead` definiuje komórki z tekstem obróconym o 90°.

```
1 \setlength\rotheadsize{15mm}
2 \begin{tabular}{|l|l|l|} \hline
3 \rothead{Rodzina} & \rothead{Gatunek} & \\
4 \rothead{Wychów} \\ \hline
5 ...
```

Rodzina	Gatunek	Wychów piskląt
Rodzina krukowatych	Kruk	Pisklęta są gniazdownikami
Rodzina krukowatych	Gawron	Pisklęta są gniazdownikami

Polecenie `multirowcell`

Polecenie `multirowcell` pozwala na definiowanie komórek obejmujących wiele wierszy. Przy składaniu nagłówka tabeli można wykorzystać analogicznie polecenie `multirowthead`.

```
1 \begin{tabular}{|l|l|l|} \hline
2 Rodzina & Gatunek & Wychów piskląt \\ \hline
3 \multirowcell{2}[0pt][l]{Rodzina krukowatych} & & \\
4 Kruk & Pisklęta są gniazdownikami \\ \cline{2-3}
5 & Gawron & Pisklęta są gniazdownikami \\ \hline
6 \multirowcell{2}[0pt][l]{Rodzina jaskółkowatych} & & \\
7 Jaskółka dymówka & \dots \\ \cline{2-3}
8 & Jaskółka oknówka & \dots \\ \hline
9 \end{tabular}
```

przesunięcie w pionie zawartości komórki (może być ujemne)

sposób wyrównania w poziomie

Rodzina	Gatunek	Wychów piskląt
Rodzina krukowatych	Kruk	Pisklęta są gniazdownikami
	Gawron	Pisklęta są gniazdownikami
Rodzina jaskółkowatych	Jaskółka dymówka	...
	Jaskółka oknówka	...

Diagonalny podział komórki

```
1 \begin{tabular}{|l|l|l|}
2 \hline
3 \diaghead(2,-1){xxxxxxxxxxxxxxxxxxxx}
4 {Krukowate} & Jaskółkowate & \\
5 \thead{Jaskółka dymówka} & & \\
6 \thead{Jaskółka oknówka} & & \\
7 \hline
8 \dots
9 \end{tabular}
```

proporcje komórki (od znaku zależy wybór przekątnej)

tekst na podstawie którego definiowana jest szerokość kolumny

Jaskółkowate	Jaskółka dymówka	Jaskółka oknówka
Krukowate		
Kruk		
Gawron		

- Zmiana wysokości wierszy (wszystkich):

```
\renewcommand{\arraystretch}{1.18}
```

- Zmiana wysokości wierszy (indywidualnie):

```
\Gape[1mm][4mm]{Rodzina krukowatych} & Kruk \\ \hline  
% pakiet makecell, najpierw góra, później dół
```

- Obracanie tabeli (wymagany pakiet `rotating`):

```
\begin{sidewaystable*}[!ht]  
\caption{Decision table}  
\label{tab:decisionTable}  
\begin{tabular}  
...  
\end{tabular}  
\end{sidewaystable*}
```

- Dostosowywanie wyglądu podpisów (pakiet `caption`):

```
\begin{table}[!hb]  
\captionsetup{labelsep=colon,belowskip=4mm,aboveskip=-9mm}  
\caption{Tablica decyzyjna}  
...
```

Wyrównywanie liczb w kolumnach

Pakiet `dcolumn` pozwala na definiowanie kolumn wyrównywanych według kropki dziesiętnej. Typ kolumny `D` przyjmuje trzy parametry: separator dziesiętny w kodzie źródłowym, separator dziesiętny w pliku wynikowym oraz maksymalną liczbę cyfr przed i po przecinku.

```
1 \begin{tabular}  
2 {l|l|c|r|D{.}{,}{3.4}|}  
3 \hline  
4 123.4 & 123.4 & 123.4 & 123.4\\  
5 12.34 & 12.34 & 12.34 & 12.34\\  
6 1.234 & 1.234 & 1.234 & 1.234\\  
7 0.1234 & 0.1234 & 0.1234 & 0.1234\\  
8 \hline  
9 \end{tabular}
```

123.4	123.4	123.4	123,4
12.34	12.34	12.34	12,34
1.234	1.234	1.234	1,234
0.1234	0.1234	0.1234	0,1234

Pakiet array (1)

Pakiet `array` wprowadza dodatkowo opcje `m{...}` i `b{...}`, które pozwalają na sterowanie sposobem rozmieszczenia zawartości komórek w pionie.

```
1 \begin{tabular}{|p{7mm}|p{7mm}|p{7mm}|}
2 \hline
3 x y z x y z x y z x y z x y z & x y z x y z x y z & 1 1 1 1 1\\\hline
4 \end{tabular}
5 \begin{tabular}{|m{7mm}|m{7mm}|m{7mm}|}
6 \hline
7 x y z x y z x y z x y z x y z & x y z x y z x y z & 1 1 1 1 1\\\hline
8 \end{tabular}
9 \begin{tabular}{|b{7mm}|b{7mm}|b{7mm}|}
10 \hline
11 x y z x y z x y z x y z x y z & x y z x y z x y z & 1 1 1 1 1\\\hline
12 \end{tabular}
```

x y z	x y z	1 1 1	x y z			x y z		
x y z	x y z	1 1	x y z	x y z	1 1 1	x y z		
x y z	x y z		x y z	x y z	1 1	x y z	x y z	
x y z			x y z	x y z		x y z	x y z	1 1 1
x y z			x y z			x y z	x y z	1 1

Pakiet array (2)

Pakiet `array` umożliwia również łatwe sterowanie formatowaniem kolumn. W preambule tabeli można umieścić polecenia `>{...}` i `<{...}`, które wstawiają odpowiednio na początku i końcu każdej komórki w danej kolumnie tekst podany jako ich argumenty.

```
1 \begin{tabular}{|>{\itshape}m{7mm}|m{7mm}|>{\bfseries}m{15mm}<{\simmm}|}
2 \hline
3 x y z x y z x y z x y z x y z & x y z x y z & 10 \\ \hline
4 x y z x y z x y z x y z x y z & x y z x y z & 12 \\ \hline
5 \end{tabular}
```

x y z		
x y z		
x y z	x y z	10 mm
x y z	x y z	
x y z		
x y z		
x y z		
x y z	x y z	12 mm
x y z	x y z	
x y z		

Pakiet longtable

Pakiet `longtable` pozwala składać tabele ciągnące się przez wiele kolejnych stron dokumentu (wymagana podwójna kompilacja).

```
1 \begin{longtable}{|c|c|c|c|c|c|c|c|c|c|}
2 \caption{Warunki terenowe}
3 \label{tab:warunkiTerenowe}\
4 \cline{2-10}
5 \multicolumn{1}{|l|}{} & \multicolumn{9}{|c|}{Zwrotnice} \\ \hline
6 {P} & 3/4 & 5 & 6 & 7/8 & 15/16 & 17 & 18 & 19/20 & 21/22 \\ \hline
7 \endfirsthead
8 \cline{2-10}
9 \multicolumn{1}{|l|}{} & \multicolumn{9}{|c|}{Zwrotnice} \\ \hline
10 {P} & 3/4 & 5 & 6 & 7/8 & 15/16 & 17 & 18 & 19/20 & 21/22 \\ \hline
11 \endhead
12 \multicolumn{10}{|c|}{Stopka tabeli} \\ \hline
13 \endfoot
14 \multicolumn{10}{|c|}{Stopka na ostatniej stronie} \\ \hline
15 \endlastfoot
16 B1 & + & + & & & & & & & \\ \hline
17 B2 & -- & & + & o+ & & & & & \\ \hline
18 ...
19 \end{longtable}
```

`longtable.tex`

Środowisko tabbing

Polecenie `\=` definiuje tabulator, a polecenie `\>` powoduje przesunięcie do kolejnego tabulatora.

```
1 \begin{tabbing}
2 Anna\hspace{7mm} \= Nowak\hspace{15mm} \= 111 222 111 \\
3 Ewa \> Nowakowska \> 222 222 222 \\
4 Konstatny \> Konstantykiewicz \> 123 123 123
5 \end{tabbing}
```

```
Anna      Nowak      111 222 111
Ewa      Nowakowska  222 222 222
Konstatny Konstantykiewicz 123 123 123
```

Bez użycia `\hspace`:

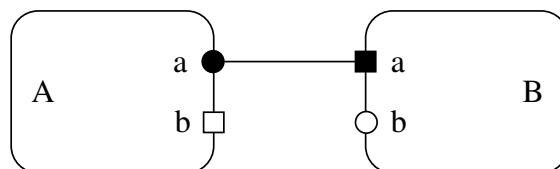
```
Anna Nowak 111 222 111
Ewa Nowak 222 222 222
Konstatny Konstantykiewicz 123 123 123
```

Możliwe jest zdefiniowanie kolumn o zadanej szerokości, np.:

```
\makebox[25mm][l]{Anna} \= \makebox[35mm][l]{Nowak} \= ... \\
% \makebox[szerokość][wyrównanie: l, c, r]{tekst}
```

Wstawianie grafiki

```
1 \begin{figure} [h]
2 \centerline{\includegraphics [scale=0.5] {xccs4}}
3 \caption{Diagram XCCS}
4 \label{fig:xccs}
5 \end{figure}
```



Rysunek: Diagram XCCS

UWAGA: Zarówno w środowisku `figure` jak i `table` polecenie `label` powinno się znaleźć bezpośrednio po poleceniu `caption`.

Jako parametry polecenia `includegraphics` można podać: `width` – skalowanie rysunku do podanej szerokości, `height` – skalowanie rysunku do podanej wysokości, `angle` – obrót o podany kąt przeciwnie do ruchu wskazówek zegara, `trim={0 20pt 60pt 120pt},clip` – obcięcie widocznej części rysunku (left, lower, right, upper).

W jednym środowisku `figure` można umieścić kilka grafik.

Etykiety

- L^AT_EX dostarcza bardzo wygodny mechanizm odsyłaczy (ang. cross-references), dzięki któremu bezbłędnie możemy odwoływać się do elementów zamieszczonych w dokumencie: rozdziałów, sekcji, równań rysunków, tabel, definicji itd.
- Etykieta jest tworzona z użyciem polecenia `label`, a odwołania z użyciem poleceń `ref` i `pageref`.
- Definiując etykiety warto nadawać im przedrostki określające typ etykietowanego obiektu, np. `eq`, `cha`, `sec`, `fig`, `def` itp. **W etykietach nie można używać spacji i znaków diakrytycznych!**

```
1 Liczbę wszystkich różnych  $\$k\$$ -wyrazowych wariacji bez powtórzeń
2 zbioru  $\$n\$$ -elementowego oznaczamy symbolem  $\$V_{n}^{k}\$$ 
3 i wyznaczamy zgodnie z wzorem~(\ref{eq:wariacje}).
4 \begin{equation}
5 \label{eq:wariacje}
6 V_{n}^{k} = \frac{\text{frac}{n!}{(n-k)!}}
7 \end{equation}
```

Liczbę wszystkich różnych k -wyrazowych wariacji bez powtórzeń zbioru n -elementowego oznaczamy symbolem V_n^k i wyznaczamy zgodnie z wzorem (1).

$$V_n^k = \frac{n!}{(n-k)!} \quad (1)$$

Przypisy

Przypisy składamy z użyciem polecenia `\footnote{tekst}`.

W Polsce przypisy umieszcza się przed znakiem przestankowym. W krajach anglosaskich przypisy umieszcza się po kropce lub przecinku.

LaTeX numeruje przypisy automatycznie. W klasie `article` numeracja jest ciągła, a w klasach `report` i `book` przypisy są numerowane w ramach rozdziałów.

Niektórych poleceń nie można używać wewnątrz argumentów innych poleceń (są to tzw. polecenia kruche), np. polecenie `\footnote` nie może być umieszczone w argumencie polecenia `\section` lub `\caption`.

Ograniczenie to można usunąć stosując polecenie `\protect`. Polecenie to odnosi się wyłącznie do instrukcji znajdującej się tuż za nim.

```
\caption{Topologia sieci w budynku C2}\protect\footnote{Zdjęcie
dzięki uprzejmości ...}
```

W przypadku konieczności zamieszczenia informacji np. o finansowaniu artykułu przy jego tytule, zamiast polecenia `\footnote` stosuje się `\thanks`

```
\title{Reachability graphs for Alvis models}\thanks{The paper
is supported by ...}}
```

Wzory matematyczne

```
1 Funkcją kwadratową nazywamy funkcję postaci
2  $f(x) = ax^2 + bx + c$ , gdzie  $a \neq 0$ .
3
4 Funkcją kwadratową nazywamy funkcję postaci
5  $f(x) = ax^2 + bx + c$ , gdzie  $a \neq 0$ .
6
7 Funkcją kwadratową nazywamy funkcję postaci
8 \begin{equation}
9  $f(x) = ax^2 + bx + c$ ,
10 \end{equation}
11 gdzie  $a \neq 0$ .
```

Funkcją kwadratową nazywamy funkcję postaci $f(x) = ax^2 + bx + c$, gdzie $a \neq 0$.

Funkcją kwadratową nazywamy funkcję postaci

$$f(x) = ax^2 + bx + c,$$

gdzie $a \neq 0$.

Funkcją kwadratową nazywamy funkcję postaci

$$f(x) = ax^2 + bx + c, \tag{2}$$

gdzie $a \neq 0$.

Wzory matematyczne – przykłady

```
1  $$A \cup B = \{x \in A \vee x \in B\}$$
2
3  $$\sum_{k=1}^{\infty} \frac{1}{k^2+1}$$
4
5  $$\lim_{n \to \infty} a_n = g \Leftrightarrow
6  \forall \varepsilon > 0 \exists N_{\varepsilon} \in \mathbb{N} \forall n > N_{\varepsilon}
7  \left| a_n - g \right| < \varepsilon$$
8
9
10 $$\int_c^d \left[ \int_{u(y)}^{v(y)} f(x,y) dx \right] dy$$
```

$$A \cup B = \{x: (x \in A) \vee (x \in B)\}$$

$$\sum_{k=1}^{\infty} \frac{1}{k^2 + 1}$$

$$\lim_{n \rightarrow \infty} a_n = g \Leftrightarrow \forall \varepsilon > 0 \exists N_{\varepsilon} \in \mathbb{N} \forall n > N_{\varepsilon} : |a_n - g| < \varepsilon$$

$$\int_c^d \left[\int_{u(y)}^{v(y)} f(x,y) dx \right] dy$$

rtcp sieci.tex

Tryb matematyczny i tekstowy

L^AT_EX inaczej składa wzory w trybie matematycznym i tekstowym!

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

L^AT_EX sam dobiera wielkość czcionek i odstępów w poszczególnych partiach wzorów:

$$\sqrt{2} \sqrt{x^2 + \sqrt{1 + \sqrt{\sqrt{2} - 1}}}$$
$$\frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \dots}}}$$

```
1  $$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} =
2  \frac{\pi^2}{6}$$
3
4  $$\sqrt{2} \sqrt{x^2 + \sqrt{1 + \sqrt{\sqrt{2}-1}}}$$
5
6  $$\frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \dots}}}$
7  $$
```

Odstępy we wzorach

We wzorach matematycznych \TeX w odmienny sposób traktuje symbole należące do różnych rodzajów, tj. cyfry, zmienne, nazwy funkcji, operatory itp. Wiele symboli używa się w różnych kontekstach w taki sposób, jakby należały do dwóch różnych rodzajów. \TeX odmiennie składa formuły w stylu wystawowym, tekstowym, indeksowym i podwójnego indeksu. W każdym przypadku istnieje wersja normalna i zacieśniona.

$f: X \rightarrow Y$
 $f : X \rightarrow Y$

```
$f\colon X \to Y$\ $f : X \to Y$
```

Użycie w drugim przypadku dwukropka jest nieprawidłowe, gdyż dwukropek jest traktowany jako operator dwuargumentowy i wstawiane są odstępy po obu stronach dwukropka (między literą f i dwukropkiem powinien być mały odstęp).

Inne mylone symbole:

- `\setminus` – odejmowanie zbiorów, `\backslash` – zwyczajny symbol;
- `\mid` – operator dwuargumentowy, `\vert` – pionowa kreska;
- `\parallel` – symbol relacji, `\Vert` – zwyczajny symbol;
- `\left<` – nawias ostrokątny, `<` – symbol relacji;
- `\right>` – nawias ostrokątny, `>` – symbol relacji.

Ograniczniki

Ograniczniki to symbole nawiasów i inne podobne symbole, które mogą rozszerzać się pionowo.

(x, y) , $\{x, y\}$, $[x, y]$, $\langle x, y \rangle$, $\lceil x, y \rceil$, $\lfloor x, y \rfloor$

```
$(x,y)$, $\{x,y\}$, $[x,y]$, $\langle x,y \rangle$,  
$\lceil x,y \rceil$, $\lfloor x,y \rfloor$
```

Rozszerzające się ograniczniki uzyskujemy za pomocą poleceń `\left` i `\right`, po których bezpośrednio umieszczamy odpowiedni symbol ogranicznika.

$$\left(\sum_{i=0}^{10} i^2\right) \quad \left\{\sum_{i=0}^{10} i^2\right\} \quad \left[\sum_{i=0}^{10} i^2\right] \quad \left|\sum_{i=0}^{10} i^2\right| \quad \left\|\sum_{i=0}^{10} i^2\right\|$$

```
1 $$  
2 \left(\sum_{i = 0}^{10} i^2 \right)\qquad  
3 \left\{\sum_{i = 0}^{10} i^2 \right\}\qquad  
4 \left[\sum_{i = 0}^{10} i^2 \right]\updownarrow\qquad  
5 \left\vert\sum_{i = 0}^{10} i^2 \right.\qquad  
6 \left.\sum_{i = 0}^{10} i^2 \right\Vert  
7 $$
```

Ograniczniki muszą zawsze występować parami!

Środowisko array (1)

Środowisko `array` funkcjonuje podobnie jak `tabular`, ale służy do tworzenia struktur tabelarycznych zawierających wyrażenia matematyczne.

$$U = (A \quad b) = \left(\begin{array}{ccccc} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{array} \right)$$

```
1  $$
2  U = \left( \begin{array}{cc} A & b \end{array} \right) =
3  \left(
4  \begin{array}{lllll}
5  a_{11} & a_{12} & \dots & a_{1n} & b_1 \\
6  a_{21} & a_{22} & \dots & a_{2n} & b_2 \\
7  \vdots & \vdots & \ddots & \vdots & \vdots \\
8  a_{m1} & a_{m2} & \dots & a_{mn} & b_m \\
9  \end{array}
10 \right)
11  $$
```

Środowisko array (2)

Ramka tworzona przez środowisko `array` posiada oś, tj. hipotetyczną linię na wysokości której umieszcza się znak minus. Linia ta przebiega mniej więcej w połowie wysokości ramki, ale można ją zmieniać za pomocą parametrów c , t i b .

$$\begin{array}{ccccccc} & & & & & 1 & 2 & 3 \\ & & & & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & = & 4 & 5 & 6 & = & 7 & 8 & 9 \\ 4 & 5 & 6 & & 7 & 8 & 9 & & & & \\ 7 & 8 & 9 & & & & & & & & \end{array}$$

```
1  $$
2  \begin{array}[t]{l}
3  1 & 2 & 3 \\
4  \end{array} =
5  \begin{array}[c]{l}
6  1 & 2 & 3 \\
7  \end{array} =
8  \begin{array}[b]{l}
9  1 & 2 & 3 \\
10 \end{array}
11  $$
```

Środowisko array (3)

$$f(x) = \begin{cases} -x^2 & \text{dla } x \leq 0, \\ \sqrt{x} + \sin x & \text{dla } x > 0. \end{cases} \quad (3)$$

$$\begin{cases} (t_1^+ - t_1^-) \circ C_{P'} = 0 \\ (t_2^+ - t_2^-) \circ C_{P'} = 0 \\ \dots \\ (t_m^+ - t_m^-) \circ C_{P'} = 0 \end{cases} \quad (4)$$

```
1 \begin{equation}
2 \label{eq:funkcjaf}
3 f(x) = \left\lbracket
4 \begin{array}{rcl}
5 -x^2 & \text{dla } x \leq 0, \\
6 \sqrt{x} + \sin x & \text{dla } x > 0.
7 \end{array} \right\rbracket.
8 \end{equation}
9
10 \begin{equation}
11 \left\{ \begin{array}{l}
12 (t_1^+ - t_1^-) \circ C_{P'} = 0 \\
13 (t_2^+ - t_2^-) \circ C_{P'} = 0 \\
14 \dots \\
15 (t_m^+ - t_m^-) \circ C_{P'} = 0
16 \end{array} \right.
17 \end{equation}
```

Pakiety AMS

W L^AT_EXu dostępnych jest kilka pakietów zawierających w nazwie skrót **ams** (American Mathematical Society). Pakiety te rozszerzają możliwości L^AT_EXa dotyczące składu wzorów matematycznych.

- amssymb** – Pakiet dostarcza dużej liczby poleceń, tworzących różnego rodzaju symbole matematyczne.
- amsfonts** – Pakiet udostępnia polecenia **mathbb** i **mathfrak**. Jest ładowany automatycznie przez **amssymb**.
- amsmath** – Pakiet stanowi główną część dystrybucji AMS-L^AT_EX. Definiuje on liczne otoczenia i polecenia, wspomagające skład wyrażen i formuł matematycznych.

Niekiedy kolejność dołączania pakietów może generować informacje o błędach. Wynikają one np. z faktu redefiniowania pewnych nazw. Zgłaszany błąd o redefiniowaniu symboli `\lll` i `\LLL` można usunąć zmieniając kolejność dołączania pakietów **amssymb** i **babel**.

Środowisko align

Każdy wiersz w środowisku `align` jest postaci

`wyr1 & wyr2 & & ... & wyrn,`

przy czym każde z wyrażeń jest przetwarzane w trybie matematycznym w stylu wystawowym. Wyrażenia o indeksach nieparzystych są wyrównane do prawej krawędzi, a o numerach parzystych do lewej.

Jeżeli separatory `&` ponumerujemy kolejno od lewej do prawej, to elementy z kolejnych wierszy będą wyrównywane w pionie wzdłuż separatorów o numerach nieparzystych.

Separatory o numerach parzystych służą do rozdzielania kolumn.

$$(\sin x)' = \cos x, \qquad (\cos x)' = -\sin x \qquad (5)$$

$$(\sin x)'' = -\sin x, \qquad (\cos x)''' = \sin x \qquad (6)$$

```
1 \begin{align}
2 (\sin x)' &= \cos x, & (\cos x)' &= -\sin x \\
3 (\sin x)'' &= -\sin x, & (\cos x)''' &= \sin x \\
4 \end{align}
```

Analogicznie funkcjonuje środowisko `align*`, ale formuły nie są automatycznie numerowane (etykietowane).

Środowisko align – przykłady (1)

$$|z| = 0 \iff z = 0, \qquad (7)$$

$$|z| \geq 0, \qquad (8)$$

$$\left| \frac{z_1}{z_2} \right| = \frac{|z_1|}{|z_2|}, \qquad (9)$$

```
1 \begin{align}
2 &\left| z \right| = 0 \iff z = 0, \label{eq:comp1} \\
3 &\left| z \right| \geq 0, \label{eq:comp2} \\
4 &\left| \frac{z_1}{z_2} \right| = \\
5 &\frac{\left| z_1 \right|}{\left| z_2 \right|}, \label{eq:comp3} \\
6 \end{align}
```

$$\sqrt{ax^2 + bx + c} = \pm x\sqrt{a} \pm t, \qquad a > 0$$

$$\sqrt{ax^2 + bx + c} = tx \pm \sqrt{c}, \qquad c > 0$$

$$\sqrt{ax^2 + bx + c} = (x - x_1)t, \qquad \Delta > 0.$$

```
1 \begin{align*}
2 \sqrt{ax^2 + bx + c} &= \pm x \sqrt{a} \pm t, && a > 0 \\
3 \sqrt{ax^2 + bx + c} &= tx \pm \sqrt{c}, && c > 0 \\
4 \sqrt{ax^2 + bx + c} &= (x - x_1)t, && \Delta > 0. \\
5 \end{align*}
```


Środowisko align – przykłady (2)

$$\begin{array}{ll} \mathbf{Act}: \frac{}{a.E \xrightarrow{a} E} & \mathbf{Com}_3: \frac{E \xrightarrow{a} E' \quad F \xrightarrow{\bar{a}} F'}{E|F \xrightarrow{\tau} E'|F'} \\ \mathbf{Sum}_j: \frac{E_j \xrightarrow{a} E'_j}{\sum_{i \in I} E_i \xrightarrow{a} E'_i}, \text{ gdzie } j \in I & \mathbf{Res}: \frac{E \xrightarrow{a} E'}{E \setminus L \xrightarrow{a} E' \setminus L}, \text{ gdzie } a, \bar{a} \notin L \end{array}$$

```
1 \begin{align*}
2 \text{\textbf{Act}}\colon & \frac{}{a.E \xrightarrow{a} E} & \text{\textbf{Com}}_3\colon & \frac{E \xrightarrow{a} E' \quad F \xrightarrow{\bar{a}} F'}{E|F \xrightarrow{\tau} E'|F'} \\
3 \text{\textbf{Sum}}_j\colon & \frac{E_j \xrightarrow{a} E'_j}{\sum_{i \in I} E_i \xrightarrow{a} E'_i}, \text{ gdzie } j \in I & \text{\textbf{Res}}\colon & \frac{E \xrightarrow{a} E'}{E \setminus L \xrightarrow{a} E' \setminus L}, \text{ gdzie } a, \bar{a} \notin L \\
4 \\
5 \\
6 \\
7 \\
8 \\
9 \\
10 \\
11 \\
12 \\
13 \\
14 \\
15 \\
16 \\
17 \end{align*}
```

Środowisko split

Środowisko `split` pozwala łatwo łamać długie wzory matematyczne. Symbol `&` wskazuje punkt wyrównania do lewej występujących po nim fragmentów linii.

$$\begin{aligned} \int x^2 e^x dx &= x^2 e^x - 2 \int x e^x dx = \\ &= x^2 e^x - 2 \left(x e^x - \int e^x dx \right) = \\ &= x^2 e^x - 2x e^x + 2 e^x + C \end{aligned} \tag{10}$$

```
1 \begin{equation}
2 \label{eq:calka1}
3 \begin{split}
4 \int x^2 e^x dx &= x^2 e^x - 2 \int x e^x dx = \\
5 &= x^2 e^x - 2 \left( x e^x - \int e^x dx \right) = \\
6 &= x^2 e^x - 2x e^x + 2 e^x + C \\
7 \end{split}
8 \end{equation}
```

Środowisko `cases` służy do definiowania przez przypadki. Automatycznie wstawia lewy nawias klamrowy.

$$\lambda(2^\alpha) = 2^{\beta-2} \begin{cases} \beta = \alpha, & \text{dla } \alpha \geq 3 \\ \beta = 3, & \text{dla } \alpha = 2 \\ \beta = 2, & \text{dla } \alpha = 1 \end{cases}$$

```

1  $$
2  \lambda(2^{\alpha}) = 2^{\beta - 2}
3  \begin{cases}
4  \beta = \alpha, & \text{dla } \alpha \geq 3 \\
5  \beta = 3, & \text{dla } \alpha = 2 \\
6  \beta = 2, & \text{dla } \alpha = 1
7  \end{cases}
8  $$

```

Notki wyjaśniające

$$\begin{aligned} x \in (U \cup V) \cap W &\iff (x \in U \cup V) \wedge x \in W, \\ &\iff (x \in U \vee x \in V) \wedge x \in W, \\ &\iff (x \in U \wedge x \in W) \vee (x \in V \wedge x \in W), \end{aligned}$$

co wynika z zastosowania tożsamości ...,

$$\begin{aligned} &\iff (x \in U \cap W) \vee (x \in V \cap W), \\ &\iff x \in (U \cap W) \cup (V \cap W). \end{aligned}$$

```

1  \begin{align*}
2  x \in (U \cup V) \cap W
3  &\iff (x \in U \cup V) \wedge x \in W, \\
4  &\iff (x \in U \vee x \in V) \wedge x \in W, \\
5  &\iff (x \in U \wedge x \in W) \vee (x \in V \wedge x \in W), \\
6  \intertext{co wynika z zastosowania tożsamości ...,}
7  &\iff (x \in U \cap W) \vee (x \in V \cap W), \\
8  &\iff x \in (U \cap W) \cup (V \cap W).
9  \end{align*}

```

Środowisko subequations

Środowisko `subequations` pozwala składać serie wzorów matematycznych, z możliwością odwoływania się do nich jako do całości lub do indywidualnych wzorów.

$$A = a.b.c.A \tag{11a}$$

$$B = A \setminus \{c\} \tag{11b}$$

Wzory (11) definiują dynamikę agentów w prezentowanym przykładzie. Agent A (11a) cyklicznie wykonuje akcje a , b i c . Agent B (11b) zdefiniowany jest ...

```
1 \begin{subequations}
2 \label{eq:resExample}
3 \begin{align}
4 A &= a.b.c.A \label{eq:rese1} \\
5 B &= A \backslash \{c\} \label{eq:rese2}
6 \end{align}
7 \end{subequations}
```

```
1 Wzory (\ref{eq:resExample}) definiują dynamikę agentów
2 w prezentowanym przykładzie. Agent  $A$  (\ref{eq:rese1})
3 cyklicznie wykonuje akcje  $a$ ,  $b$  i  $c$ .
4 Agent  $B$  (\ref{eq:rese2}) zdefiniowany jest ...
```

Litery w trybie matematycznym

- $\mathcal{A}, \mathcal{B}, \dots$ – `\mathcal{A}`, `\mathcal{B}`
- $\mathbb{N}, \mathbb{R}, \dots$ – `\mathbb{N}`, `\mathbb{R}`
- $\mathcal{A}, \mathcal{B}, \dots$ – `\mathfrak{A}`, `\mathfrak{B}`
- ϕ, φ – `\phi`, `\varphi`
- i, j – `\imath`, `\jmath` – polecenia te pozwalają uzyskać litery i i j bez kropek, co jest przydatne, gdy chcemy np. umieścić nad i symbol wektora, np. $\vec{i} \cdot \sqrt[3]{2}$ – `\vec{\imath} \cdot \sqrt[3]{2}`
- większość standardowych funkcji matematycznych ma zdefiniowane swoje nazwy i należy je stosować, np. $\sin x$, $\sin x$ – `\sin x`, `\sin x` – drugi zapis nie jest poprawny.

`toffi` = `toffi` = `toffi` = `toffi` = `toffi`

```
1 $$
2 \mathbf{toffi} = \mathrm{toffi} = \mathsf{toffi}
3 = \mathtt{toffi} = \mathit{toffi}
4 $$
```

Jeżeli w trybie matematycznym stosujemy nazwę wieloliterową np. *off* i pojawiają się niepotrzebne odstępy między literami – *off* – to należy zastosować polecenie `mathit` – `\mathit{off}`.

$$\overline{m + nx + y}$$

$$\underbrace{a + a + \dots + a}_n = n \cdot a$$

$$A = \{ \overbrace{n \in \mathbb{Z}}^{\text{opis}} : \overbrace{n \neq 0 \wedge n \bmod 2 = 0}^{\text{warunek}} \}$$

$$k = \prod_{\substack{0 < i \leq n \\ 0 \leq j < n \\ i \neq j}} (i - j)$$

```

1 \begin{align*}
2 & \& \overline{m + n} \underline{x + y} \\
3 & \& \underbrace{a + a + \ldots + a}_{n} = n \cdot a \\
4 & \& A = \{ \overbrace{n \in \mathbb{Z}}^{\text{opis}} : \overbrace{n \neq 0 \wedge n \bmod 2 = 0}^{\text{warunek}} \} \\
5 & \& \overbrace{n \neq 0 \wedge n \bmod 2 = 0}^{\text{warunek}} \\
6 & \& k = \prod_{\substack{0 < i \leq n \\ 0 \leq j < n \\ i \neq j}} (i - j) \\
7 & \& \\
8 \end{align*}

```

Polecenie newtheorem

Polecenie `newtheorem` służy do definiowania środowisk typu definicja, twierdzenie itp. oraz powiązanych z nimi liczników.

```

1 \theoremstyle{definition}
2 \newtheorem{df}{Definicja}
3 \newtheorem{tw}{Twierdzenie} [chapter]
4 \theoremstyle{remark}
5 \newtheorem{lm}[tw]{Lemat} % w preambule
6
7 \begin{df}
8 \label{def:grafEtykietowany}
9 Graf skierowany  $\mathcal{G} = (V, A, \gamma)$  nazywamy \em grafem
10 etykietowanym nad zbiorem etykiet \L, jeżeli łuki grafu
11  $\mathcal{G}$  mają przypisane etykiety ze zbioru \L.
12 \end{df}

```

Definicja 1 Graf skierowany $\mathcal{G} = (V, A, \gamma)$ nazywamy grafem etykietowanym nad zbiorem etykiet L , jeżeli łuki grafu \mathcal{G} mają przypisane etykiety ze zbioru L .

Uwaga: W przykładzie zdefiniowano trzy środowiska: `df` (Definicja), `tw` (Twierdzenie) i `lm` (Lemat). W przypadku definicji użyto numeracji ciągłej (w całej książce), a w przypadku twierdzeń numeracja jest prowadzona w ramach rozdziałów. Lematy współdziela licznik z twierdzeniami. Definicje i twierdzenia będą formatowane z użyciem stylu `definition`, a lematy z użyciem stylu `remark`.

W pakiecie `amsthm` dostępne jest również środowisko `proof`.

`srodowiska.tex`

Środowisko thebibliography

Środowisko `thebibliography` służy do samodzielnego przygotowania bibliografii.

```
1 Czasowe CP-sieci mogą one być wykorzystywane do modelowanie systemów
2 czasu rzeczywistego, zarówno do przedstawienia specyfikacji
3 wymagań~\cite{MSzTSz03} jak i mniej lub bardziej szczegółowego
4 projektu takiego systemu.
5
6 \begin{thebibliography}{9}
7
8 \bibitem{CSh99}
9 CERONE, A., MAGGIOLLO-SHETTINI, A.: \textit{Time-Based Expressivity
10 of Time Petri Nets for System Specification}, Theoretical Computer
11 Science, Vol. 216, 1999, pp. 1-53.
12
13 \bibitem{HM}
14 HEITMEYER, C., MANDRIOLI, D. (Eds.): \textit{Formal Methods for
15 Real-Time Computing}, John Wiley & Sons, Chichester, 1996.
16
17 \bibitem{MSzTSz03}
18 SZPYRKA M., SZMUC T.: \textit{Specification of external system
19 behaviour based on D-nets: theoretical aspects and computer tools},
20 Automatyka, AGH, Tom 7, Zeszyt 1-2, Kraków, 2003, str. 275 - 282.
21
22 \end{thebibliography}
```

Środowisko thebibliography – uwagi

- L^AT_EX umieszcza w spisie bibliografii wszystkie pozycje zdefiniowane w ramach środowiska `thebibliography`, nawet jeśli nie są cytowane – zazwyczaj brak powołania się w treści dokumentu na pozycję wykazaną w spisie literatury jest błędem edytorskim.
- Bibliografia jest wyświetlana dokładnie w takiej kolejności jak wpisano ją w środowisku `thebibliography`. Zachowane zostaje również formatowanie.
- Polecenie `cite` można użyć z opcjonalnym parametrem np. `\cite[str. 24]{CSh99}`. W efekcie w tekście zobaczymy odwołanie postaci [*7*, *str. 24*].
- Parametr środowiska `thebibliography` służy do rezerwacji odpowiedniej ilości miejsca na etykiety i powinien zawierać tyle znaków ile najdłuższa etykieta, np. *99*, gdy mamy 34 pozycje w bibliografii, albo *aaa*, gdy pozycji jest 102 (zakładamy, że są numerowane).
- W celu zmiany nazwy dla sekcji (rozdziału) z bibliografią trzeba redefiniować polecenie `\refname` (dla klasy *article*) lub `\bibname` (dla klas *report* i *book*), np. `\renewcommand{\bibname}{Literatura}`

BIB_TE_X

BIB_TE_X jest systemem przeznaczonym do tworzenia bibliografii we współpracy z **L_AT_EX**em. Baza danych **BIB_TE_X**a składa się z plików o rozszerzeniu **bib** zawierających opis poszczególnych pozycji bibliograficznych, z których chcemy korzystać.

```
@Book{Wil98,  
author = {Wilson, R. J.},  
title = {Wprowadzenie do teorii grafów},  
publisher = {PWN},  
year = {1998},  
address = {Warszawa},  
}
```

```
@Article{AlDi94,  
author = {Alur, R. and Dill, D.},  
title = {A theory of timed automata},  
journal = {Theoretical Computer Science},  
year = {1994},  
volume = {126},  
number = {2},  
pages = {183-235},  
}
```

zrodla.bib

Wsparcie BIB_TE_Xa w TeXstudio

```
@book{ID,  
author = {author},  
title = {title},  
date = {date},  
OPTeditor = {editor},  
OPTeditora = {editora},  
OPTeditorb = {editorb},  
OPTeditorc = {editorc},  
OPTtranslator = {translator},  
OPTannotator = {annotator},  
OPTcommentator = {commentator},  
OPTintroduction = {introduction},  
OPTforeword = {foreword},  
OPTafterword = {afterword},  
OPTsubtitle = {subtitle},  
OPTtitleaddon = {titleaddon},  
OPTmaintitle = {maintitle},  
OPTmainsubtitle = {mainsubtitle},  
OPTmaintitleaddon = {maintitleaddon},  
OPTlanguage = {language},  
OPToriglanguage = {origlanguage},  
OPTvolume = {volume},  
OPTpart = {part},  
OPTedition = {edition},  
OPTvolumes = {volumes},
```

Środowisko generuje rekord o odpowiedniej strukturze. Pola z przedrostkiem **OPT** są opcjonalne. Jeżeli je wypełniamy, to przedrostek należy usunąć.

Korzystanie z plików BIB_TE_XA

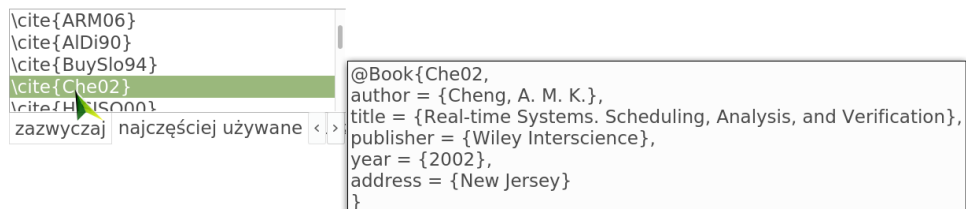
- W celu wygenerowania spisu literatury należy w dokumencie umieścić polecenia **bibliographystyle** i **bibliography**.
- Polecenie **bibliographystyle** (umieszczamy przed pierwszym użyciem **cite**) określa styl składania bibliografii, który zostanie użyty, np. *abbrv*, *alpha*, *plain*. Formatowanie i kolejność umieszczenia pozycji w bibliografii są zależne od wybranego stylu.
www.cs.stir.ac.uk/~kjt/software/latex/showbst.html
- Polecenie **bibliography** umieszcza się zwykle na końcu dokumentu, w miejscu gdzie ma zostać zamieszczony spis literatury. Jego argumentem jest lista plików (bez rozszerzeń) z bazami danych BIB_TE_XA.
- BIB_TE_X dodaje do bibliografii tylko te pozycje, które zacytowano. Niecytowaną pozycję można dodać poleceniem `\nocite{etykieta}`. Polecenie `\nocite{*}` doda wszystkie rekordy.

```
1 \begin{document}
2 \bibliographystyle{abbrv}
3 ...
4 Bardziej szczegółowe wprowadzenie do teorii grafów
5 można znaleźć w~\cite{Wil98}.
6 ...
7 \bibliography{expertsystems,mathematics,formalmethods}
```

[25] R.J. Wilson. *Wprowadzenie do teorii grafów*. PWN, Warszawa, 1998.

Korzystanie z plików BIB_TE_XA w TeXstudio

w~monografii~\cite|



Przy stosowaniu BIB_TE_XA, TeXstudio pozwala łatwo przeglądać bazy danych w plikach *bib* i ułatwia znalezienie odpowiedniej pozycji.

bibtex_demo.tex

- Autorów (redaktorów) podajemy wpisując najpierw nazwisko (może składać się z więcej niż jednego słowa, a potem **po przecinku** imiona lub inicjały.
- Jeżeli potrzebujemy podać więcej niż jednego autora, to separatorem pomiędzy autorami jest słowo **and**.
- Przy długiej liście nazwisk można po wymienieniu kilku z nich dodać **and others** co wygeneruje w bibliografii tekst **et al**.
- W zależności od użytego stylu **BIBTEX** może zmienić wielkość liter np. w tytułach. Jeżeli chcemy, aby jakiś tekst nie podlegał tego typu modyfikacjom (np. skrót), to umieszczamy go dodatkowo w klamrach, np.
title = {Sieci {P}etriego},
- Przy tworzeniu bibliografii konieczna jest dwukrotna kompilacja pliku. Za pierwszym razem zbierane są informacje o cytowanych pozycjach, a za drugim razem budowana jest właściwa bibliografia.
- Wygenerowana bibliografia znajduje się w pliku o rozszerzeniu **bbl** (stara wersja jest zawsze nadpisywana przy nowej kompilacji). Jeżeli konieczne są ręczne poprawki, to można je wykonać w tym pliku, a później dołączyć ten plik za pomocą polecenia **input** użytego w miejsce **thebibliography**. Robi się to np. przy finalnej kompilacji książki, gdy wiadomo, że nie będziemy już zmieniać listy cytowanych źródeł.

Środowisko verbatim

```
1 Rozkaz umieszczony w pętli \verb!while! jest
2 powtarzany do momentu, gdy wartość wyrażenia będzie równa 0.
3
4 \begin{verbatim}
5 while (wyrażenie) instrukcja;
6 \end{verbatim}
7
8 Pętla \verb+do while+ jest podobna do pętli \verb!while!,
9 z tą różnicą, że warunek kontynuacji (wyrażenie)
10 jest sprawdzany po wykonaniu instrukcji
11
12 \begin{verbatim}
13 do instrukcja while (wyrażenie);
14 \end{verbatim}
```

Rozkaz umieszczony w pętli **while** jest powtarzany do momentu, gdy wartość wyrażenia będzie równa 0.

```
while (wyrażenie) instrukcja;
```

Pętla **do while** jest podobna do pętli **while**, z tą różnicą, że warunek kontynuacji (wyrażenie) jest sprawdzany po wykonaniu instrukcji

```
do instrukcja while (wyrażenie);
```


- Pakiet `color` dostarcza makroinstrukcje umożliwiające kolorowanie tekstu, tła i definiowanie własnych kolorów.

- **Tekst czerwony, niebieski, zielony i jeszcze raz czerwony.**

```
{\color{red} Tekst czerwony}, {\color{blue}niebieski},  
\textcolor{green}{zielony}  
\textcolor{red}{\bf i jeszcze raz czerwony}.
```

- **czerwone tło, żółte tło i zielony tekst,**

```
\colorbox{red}{czerwone tło},  
\colorbox{yellow}{\color{green} żółte tło i zielony tekst},
```

- **tekst w ramce,** `\fcolorbox{green}{yellow}{tekst w ramce}`

- definiowanie własnych kolorów:

```
\definecolor{darkred}{rgb}{0.9,0,0}  
\definecolor{grey}{rgb}{0.4,0.4,0.4}  
\definecolor{orange}{rgb}{1,0.6,0.05}
```

```
przykład, \fcolorbox{darkred}{grey}{\color{orange} przykład}
```

- **mieszanie kolorów,** `{\color{blue!50!white} mieszanie kolorów}`

Pakiet listings

Pakiet `listings`, podobnie jak środowisko `verbatim`, pozwala na umieszczanie w dokumencie kodu źródłowego, ale pozwala kontrolować jego wygląd. Na podstawie załadowanych opcji, umożliwia m.in. kolorowanie składni.

```
1 \usepackage{listings}  
2 \lstloadlanguages{Ada,C++}  
3  
4 \definecolor{darkred}{rgb}{0.9,0,0}  
5 \definecolor{grey}{rgb}{0.4,0.4,0.4}  
6 \definecolor{darkgreen}{rgb}{0.2,0.5,0.05}  
7 \lstset{language=C++,  
8 basicstyle=\ttfamily\small,  
9 keywordstyle=\color{darkgreen}\ttfamily\bfseries\small,  
10 stringstyle=\color{red}\ttfamily\small,  
11 commentstyle=\color{grey}\ttfamily\small,  
12 numbers=left,  
13 numberstyle=\color{darkred}\ttfamily\scriptsize,  
14 identifierstyle=\ttfamily\small,  
15 showstringspaces=false,  
16 morekeywords={}}
```

Polecenie `lstset` można użyć w dokumencie wielokrotnie zmieniając bieżące ustalenia – nadpisywane są tylko te ustawienia, które użyjemy w poleceniu.

Pakiet listings

Rozkaz umieszczony w pętli `while` jest powtarzany do momentu, gdy wartość wyrażenia będzie równa 0.

Listing 1: Pętla while

`while` (wyrażenie) instrukcja;

Pętla `do while` jest podobna do pętli `while`, z tą różnicą, że warunek kontynuacji (wyrażenie) jest sprawdzany po wykonaniu instrukcji

`do` instrukcja `while` (wyrażenie);

```
1 Rozkaz umieszczony w pętli \lstinline!while! jest
2 powtarzany do momentu, gdy wartość wyrażenia będzie równa 0.
3
4 \begin{lstlisting}[caption=Pętla while]
5 while (wyrażenie) instrukcja;
6 \end{lstlisting}
7
8 Pętla \lstinline!do while! jest podobna do pętli \lstinline!while!,
9 z tą różnicą, że warunek kontynuacji (wyrażenie)
10 jest sprawdzany po wykonaniu instrukcji
11
12 \begin{lstlisting}
13 do instrukcja while (wyrażenie);
14 \end{lstlisting}
```

Sterowanie opcjami środowiska lstlisting (1)

- Ogranicznikami dla polecenia `lstlisting` może być dowolny znak, który nie występuje w kodzie.
- Opcje dla danego listingu ustalamy jako listę wartości typu `key=value` oddzielonych przecinkami.
- `firstline=3` – numer pierwszej wyświetlanej linii kodu,
- `lastline=7` – numer ostatniej wyświetlanej linii kodu,
- `numbers=left` – miejsce umieszczenia numerów linii kodu,
- `float=[th]` – potraktowanie listingu jako wstawki,
- `caption=Nagłówek listingu` – nagłówek (listing numerowany),
- `title=Nagłówek listingu` – nagłówek (listing nienumerowany),
- `tabsize=2` – rozmiar tabulatora,
- `showspaces=true` – włączenie/wyłączenie wyświetlania spacji,
- `showtabs=true` – włączenie/wyłączenie wyświetlania tabulatorów,
- `tab=\rightarrowfill` – zdefiniowanie sposobu wyświetlania tabulatorów,
- `extendedchars=true` – włączenie obsługi znaków diakrytycznych,

Sterowanie opcjami środowiska `lstlisting` (2)

- `aboveskip=2pt` – odstęp przed listingiem,
- `belowskip=2pt` – odstęp po listingu,
- `frame=trBL` – definicja ramki (mała litera – linia pojedyncza, wielka litera – linia podwójna),
- `frameround=fttt` – *t* wskazuje zaokrąglone narożniki (kolejność: top, right, bottom, left),
- `label=etykieta` – definicja etykiety dla listingu (odwołania z użyciem `ref` i `pageref`),
- `backgroundcolor=\color{yellow}` – definicja koloru tła,
- `emph={słowo1, słowo2}` – lista słów do dodatkowego wyróżniania,
- `emph={ [2] słowo1, słowo2 }` – lista słów do dodatkowego wyróżniania (klasa nr 2),
- `emphstyle=\underbar` – zdefiniowanie sposobu wyróżniania słów,
- `emphstyle={ [2] \color{red} }` – zdefiniowanie sposobu wyróżniania słów (klasa nr 2),
- `captionpos=b` – miejsce umieszczenia nagłówka.

Środowisko `lstlisting` – przykład

```
1 \begin{lstlisting}[caption=Obliczanie pierwiastka kwadratowego,  
2 captionpos=t, label=src:sqrt, frame=LBtr, frameround=tfttf]  
3 int main()  
4 {  
5     const float EPS = 0.0001;  
6     float x1, x2, a;  
7  
8     cout << "Podaj liczbę rzeczywistą: ";  
9     cin >> a;  
10    x1 = a;  
11    x2 = 0.5 * (x1 + a / x1);  
12  
13    while(fabs(x2 - x1) > EPS)  
14    {  
15        x1 = x2;  
16        x2 = 0.5 * (x1 + a / x1);  
17    }  
18  
19    cout << "Pierwiastek: " << x2 << endl;  
20 }  
21 \end{lstlisting}
```

listingi.tex

```
1 \lstdefinlanguage{Alvis}
2 {
3 keywords={agent,in,out,delay,jump,exec,alt,data,type,
4 critical,start,exit,far,loop,if,else,elseif,select,
5 cli,sti,proc,elseif,every,environment,null},
6 ndkeywords={Char,Bool,Int,Double,String,rem,sqrt,
7 head,tail,signal,durable,queue},
8 sensitive=true,
9 morecomment=[l]{--},
10 morecomment=[s]{/*}{*/},
11 morestring=[b]",
12 }
```

Dia Diagram Editor

<https://live.gnome.org/Dia>

Dia jest programem rozpowszechnianym na licencji GPL. Umożliwia tworzenie grafiki prezentacyjnej różnorodnego rodzaju – schematów blokowych, diagramów i wykresów. Zawiera bibliotekę gotowych obiektów (symboli).

Sposób użycia:

1. Przygotować diagram – można używać polskie znaki diakrytyczne.
2. Wyeksportować do formatu **eps** – *Encapsulated Postscript (używający czcionek Pango) (*.eps)*.
3. Jeżeli potrzebujemy rysunek w formacie **pdf**, należy wykonać konwersję stosując polecenie **epstopdf**, np.: `epstopdf plik.eps`

- **Adobe Reader** – bezpłatna przeglądarka dokumentów zapisanych w formacie **pdf**;
- **Gimp** – popularne narzędzie do przetwarzania grafiki bitowej (licencja GPL).

Skopiowanie grafiki z pliku pdf – sposób użycia:

1. Otworzyć dokument korzystając z Adobe Reader, zastosować maksymalne możliwe powiększenie.
2. Korzystając z polecenia **Tools** → **Select&Zoom** → **Snapshot tools** zaznaczyć interesujący fragment pliku (np. grafikę) – zostanie automatycznie skopiowany do schowka.
3. W Gimpie utworzyć nowy plik **Plik** → **Utwórz** → **Ze Schowka**, dokonać wymaganej obróbki grafiki.
4. Wyeksportować grafikę do formatu obsługiwanego przez **pdflatex**, np. **png**, **jpg**.

Xfig

<http://www.xfig.org/>

- **Xfig** jest programem do tworzenia grafiki wektorowej (open source). Umożliwia eksport do bardzo wielu formatów graficznych (zarówno grafika wektorowa jak i bitowa). Zawiera bibliotekę gotowych obiektów (symboli).
- **Xfig** stosuje własny dobrze udokumentowany format **fig**. Do formatu **fig** można wyeksportować np. wykresy 2D i 3D generowane w Octave.
- Przygotowany rysunek można m.in. wyeksportować do formatu **eps** i **pdf**.
- **Xfig** umożliwia przygotowanie rysunków łączonych z wzorami zapisanymi w \LaTeX u.

- **GNU Octave** jest środowiskiem oraz interaktywnym językiem programowania przystosowanym do realizacji wektoryzowalnych obliczeń numerycznych.
 - Składnia Octave odpowiada składni programu Matlab, chociaż w wielu aspektach (np. grafiki czy animacji) jest uboższa. Programy napisane w Octave z zachowaniem pewnych reguł, można bez problemu uruchamiać w Matlabie.
 - Octave jest oprogramowaniem typu open source. Octave dostępny jest na większości systemów uniksowych. Rozprowadzany jest na zasadach licencji GNU GPL.
 - Strona domowa Octave: <http://www.gnu.org/software/octave>
1. John W. Eaton, David Bateman, Søren Hauberg: **GNU Octave A high-level interactive language for numerical computations**. Edition 3 for Octave version 3.0.5, July 2007 (pdf + Reference Card)
 2. Alfio Quarteroni, Fausto Saleri: **Scientific Computing with MATLAB and Octave**. Second Edition, Springer 2006
 3. P.J.G. Long: **Introduction to Octave**. Department of Engineering, University of Cambridge 2005 (pdf)

Rysowanie wykresów w 2D

Funkcja **plot** w najprostszej wersji przyjmuje jako swoje argumenty wartości odciętych i rzędnych punktów należących do wykresu. Punkty te są następnie traktowane jako węzły łamanej.

```
plot([2 5 6 4 4])

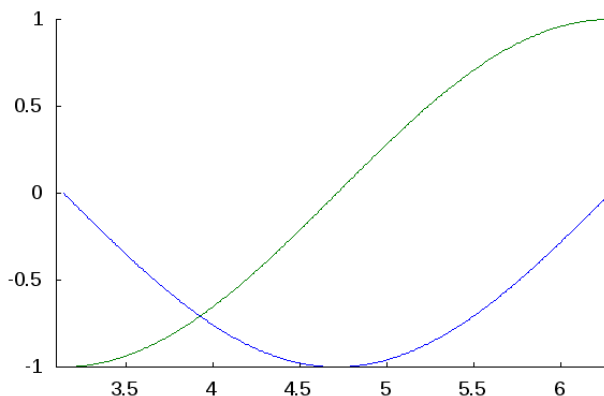
x = [3:10];
y = [2 3 6 4 4 1 7 5];
plot(x, y)

x = [0 : 0.05 : 4];
plot(x, sqrt(x))

x = [1:5];
y = [2 4 5 3 2; 1 1 2 5 1];
plot(x, y)

x = [pi : 0.01 : 2*pi];
plot(x, [sin(x); cos(x)])

# lub
plot(x, sin(x), x, cos(x))
```



Formatowanie wykresu – kolory i style linii

- Kolor linii wykresu można zmienić podając jednoliterowe oznaczenie jednego z predefiniowanych kolorów: **y** – żółty, **m** – purpurowy, **c** – zielononiebieski (cyan), **r** – czerwony, **g** – zielony, **b** – niebieski, **w** – biały i **k** – czarny.

```
plot(x, cos(x), 'r')
plot(x, cos(x), 'g')
plot(x, cos(x), 'r', x, sin(x), 'g')
```

- Kolor linii wykresu można ustalić również podając trzy wartości z przedziału $[0, 1]$, określające kolor w schemacie RGB:

```
plot(x, cos(x), 'color', [0.4 0.5 0.2])
```

- Zamiast linii ciągłej można wybrać jeden z dostępnych sposobów oznaczania wyliczonych punktów wykresu: **+** – symbol plusa, **o** – kółko, ***** – gwiazdka, **x** – znak x, **.** – kropka, **^** – znak potęgi, **s** – kwadrat, **d** – romb.

```
x = [0 : 0.1 : 2*pi];
plot(x, sin(x) + cos(x), '*')
```

Formatowanie wykresu – osie układu współrzędnych

Bezpośrednio po narysowaniu wykresu można zmodyfikować parametry dotyczące wyświetlania osi wykresu stosując polecenie **axis**.

- Ustawienie zakresu dla osi x : **axis**([-2 2])
- Ustawienie zakresu dla osi x i y : **axis**([-5 5 -2 2])
- Wymuszenie takiej samej skali na obu osiach: **axis**('equal')
- Włączenie/wyłączenie osi: **axis**('on'), **axis**('off')
- Włączenie znaczników osi dla wybranej/ych osi i wyłączenie dla pozostałych: **axis**('tic[x]')
- Odwrócenie osi y (mniejsze wartości na górze): **axis**('ij')
- Odwrócenie osi y (większe wartości na górze): **axis**('xy')

Opcje można łączyć w ramach polecenia **axis**:

```
axis([-2 8 -2 2], 'square', 'tic[x]')
```

Rysowanie wykresów

- `fplot` – użycie wskaźnika do funkcji: `fplot(@sin, [-10 10])`
- `fplot` – użycie nazwy funkcji: `fplot('sin', [-10 10])`

- Wykres słupkowy:

```
y = [2 3 3 4 3 5 6 2 1];  
bar(y)  
bar(y, 0.2)
```

- Wykres słupkowy (ułożenie poziome): `barh(y)`
- Wykres schodkowy: `stairs(y)`
- Wykres kołowy: `pie(y)`
- Wypełniony wykres powierzchniowy:

```
x = linspace(0, 10, 200);  
y = sin(x);  
Y = y .* x;  
area(y)
```

Formatowanie wykresu

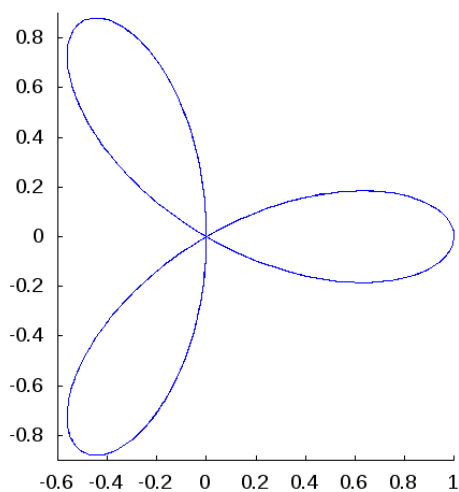
- Włączenie/wyłączenie ramki: `box('off')`, `box('on')`, `box, box on`, `box off`
- Wstawienie oznaczeń dla osi układu współrzędnych:

```
xlabel('X')  
ylabel('Y')  
xlabel 'X'  
ylabel 'Y'
```

- Dodanie tytułu do wykresu: `title('f(x) = x*sin(x)')`
- Dodanie etykiety do wykresu: `text(50, 50, 'f(x)')`
- Włączenie/wyłączenie siatki: `grid('off')`, `grid('on')`, `grid, grid on`, `grid off`
- Włączenie/wyłączenie zamrażania wykresu (kolejny wykres będzie nakładany na obecny): `hold('off')`, `hold('on')`, `hold, hold on`, `hold off`

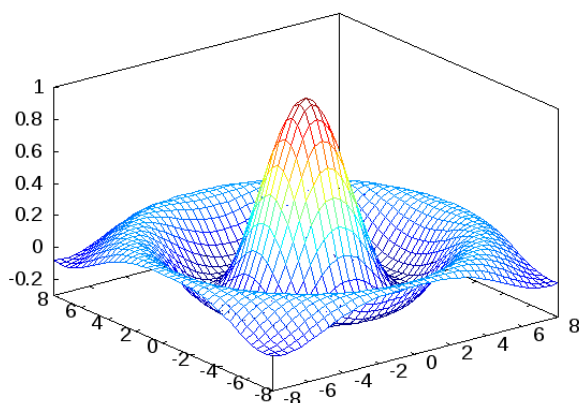
Krzywe parametryczne

```
t = [0 : 0.01 : 2*pi];  
r = cos(3 * t);  
x = r .* cos(t);  
y = r .* sin(t);  
plot(x, y)  
axis('equal')
```



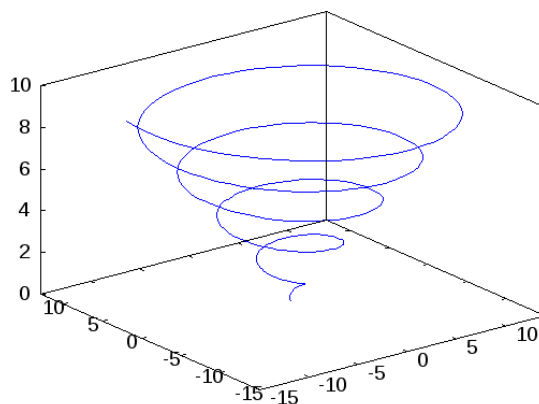
Wykresy 3D

```
x = y = linspace(-8, 8, 41)';  
[xx, yy] = meshgrid(x, y);  
z = sqrt(xx.^2 + yy.^2) + eps;  
z = sin(z) ./ z;  
mesh(x, y, z)
```



Funkcja `meshgrid` przygotowuje macierze z odpowiednimi wartościami z dziedziny funkcji $f(x, y)$. Funkcja `mesh` rysuje wykres powierzchniowy.

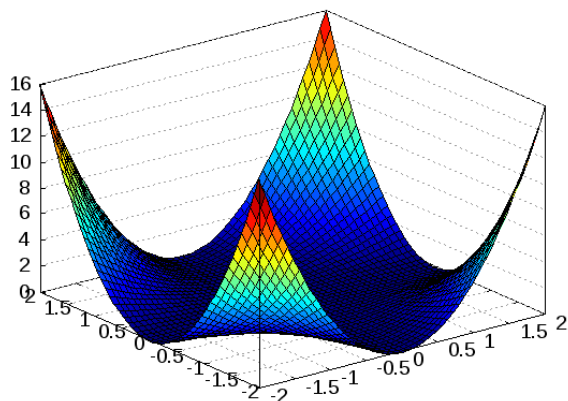
```
1 t = linspace(0, 30, 301);
2 x = sin(t) .* t ./ 2;
3 y = cos(t) .* t ./ 2;
4 z = t ./ 3;
5 plot3(x, y, z)
```



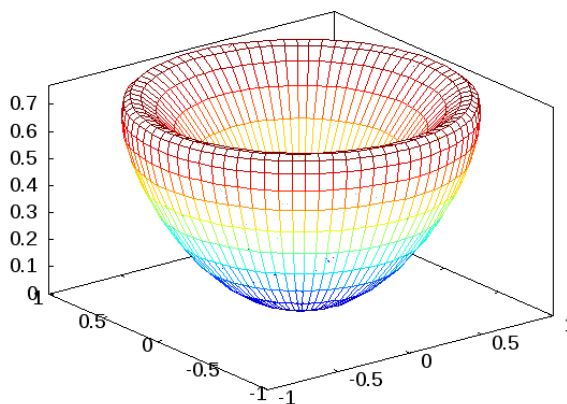
Rysowanie wykresów 3D

- Użycie funkcji `surf` zamiast `mesh` pozwala uzyskać wykres powierzchniowy, zamiast siatkowego.

```
1 x = y = linspace(-2, 2, 51)';
2 [xx, yy] = meshgrid(x, y);
3 z = (xx.^ 2) .* (yy.^ 2);
4 surf(x, y, z)
```



```
1 t = linspace(0, 2*pi, 81);
2 p = linspace(0, pi/2, 21);
3 [tt,pp] = meshgrid(t, p);
4 r = 2 * cos(pp);
5 x = r .* sin(pp) .* cos(tt);
6 y = r .* sin(pp) .* sin(tt);
7 z = r .* sin(pp) .* cos(pp);
8 mesh(x,y,z)
```



Zapisywanie wykresu do pliku

Zapis utworzonego wykresu do pliku jest realizowany przez funkcję **print**:

```
print -d... nazwa-pliku
```

W miejsce ... należy wpisać format graficzny, np.: eps, fig, png, jpg, gif, svg.

```
print -dfig wykres.fig
```

```
print -dpng wykres.png
```

Przed nazwą pliku można dodać opcje wydruku, np. **-mono** zamienia rysunek na czarno-biały,

Po wyeksportowaniu wykresu 2D lub 3D do formatu **fig** można je otworzyć w Xfigu i wykorzystać jako element przygotowywanego rysunku. Wykresy wyeksportowane do formatu **fig** są w formacie wektorowym.

- **PGF (Portable Graphics Format)** – wewnętrzny silnik dostarczający kolekcję niskopoziomowych elementów do konstruowania grafiki.
- **TikZ (TikZ ist kein Zeichenprogramm)** – wysokopoziomowy interfejs użytkownika.

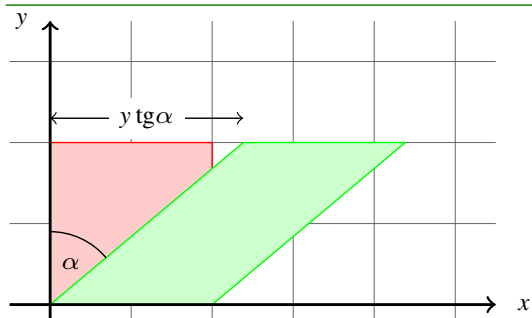
Oba pakiety są zintegrowane z \LaTeX em pozwalając na tworzenie wysokiej jakości grafiki wektorowej. Rysunki są definiowane jako seria poleceń \LaTeX a.

```

1 \usepackage{tikz}
2 \usetikzlibrary{calc,through,backgrounds,positioning}
3 % ewentualnie inne potrzebne biblioteki
4
5 \begin{tikzpicture}
6 % polecenia tworzące rysunek
7 \end{tikzpicture}
8
9 \begin{figure}[!ht] % lub jako wstawka
10 \begin{center}
11 \begin{tikzpicture}
12 % polecenia tworzące rysunek
13 \end{tikzpicture}
14 \end{center}
15 \caption{Podpis...}
16 \label{fig:etykieta}
17 \end{figure}

```

TikZ – pierwszy przykład



tikz_demo.tex

```

1 \begin{tikzpicture}
2 \draw[step=1cm,draw=grey,very thin] (-0.5,-0.2) grid (5.5,3.5);
3 \filldraw [draw=red,fill=red!20!white] (0,0) -- (0,2) -- (2,2)
4 -- (2,0) -- cycle;
5 \filldraw [draw=green,fill=green!20!white] (0,0)
6 -- (intersection of 0,0--40:3 and 0,2--4,2) coordinate (t)
7 -- ++(2,0) -- (2,0) -- cycle;
8 \draw[thick] [->] (-0.5,0) -- (5.5,0) node [right=3pt] {$x$};
9 \draw[thick] [->] (0,-0.2) -- (0,3.5) node [left=3pt] {$y$};
10 \node at (0,0.5) [right] {$\alpha$};
11 \draw (40:0.9) arc (40:90:0.9);
12 \draw[<->] (0,2.3) -- node [fill=white] {$y\,\{\rm tg\}\,\alpha$}
13 ( $ (t) + (0,0.3) $ );
14 \end{tikzpicture}

```

Rysowanie ścieżek

Ścieżka jest ciągiem linii prostych i krzywych, które są połączone.

- Rysowanie łamanej:
`\draw (0,0) -- (0,2) -- (1,3.25);`
- Rysowanie łamanej zamkniętej:
`\draw (2,0) -- (3,0) -- (3,1) -- cycle;`
- Aby narysować krzywą podajemy: punkt początkowy, 2 punkty kontrolne i punkt końcowy. Można podać jeden punkt kontrolny, który będzie wówczas użyty 2 razy:
`\draw (0,0) .. controls (1,1) and (2,1) .. (2,0);!`
- Linia złożona z dwóch odcinków i krzywej:
`\draw (0,0) -- (1,0) -- (2,2) .. controls (1,1) and (3,1) .. (2,0);`
- Dwa niepołączone odcinki:
`\draw (-1,-1) -- (2,1) (2,2) -- (3,2);`

Polecenie `draw` może być uzupełnione o opcje umieszczone w nawiasie kwadratowym (dotyczą one całej ścieżki), np.: `\draw[thick,rounded corners=8pt] ...`,

`\draw[blue,very thick] ...`

Dostępne grubości linii: *very thin*, *thin*, *semithick*, *thick*, *very thick*, *ultra thick*.

Dostępne style linii: *dashed*, *dotted*, *loosely dashed/dotted*, *densely dashed/dotted*.

Wypełnianie wnętrza i cieniowanie

Użycie polecenia `fill` zamiast `draw` powoduje zamalowanie wnętrza obszaru

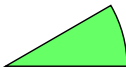
ograniczonego ścieżką:



```
\tikz \fill[red!60!white] (0,0) -- (1.2,0) arc (0:30:1.2) -- cycle;
```

Zastosowanie `filldraw` powoduje jednoczesne narysowanie krawędzi i wypełnienie

wnętrza:



```
\tikz \filldraw[green!60!white, draw=black] (0,0) -- (1.2,0) arc (0:30:1.2) -- cycle;
```

Domyślnie do cieniowania używane są kolory szary i biały. Użycie polecenia `shadedraw` powoduje jednoczesne rysowanie kształtu i cieniowanie.



```
1 \tikz \shade (0,0) rectangle (1.8,1);
2 \tikz \shadedraw (0,0) rectangle (1.8,1);
3 \tikz \shade[top color=yellow,bottom color=black] (0,0)
4   rectangle (1.8,1);
5 \tikz \shadedraw[inner color=yellow,outer color=black,
6   draw=yellow] (0,0) rectangle (1.8,1);
7 \tikz \shade[ball color=green] (0,.5) circle (.5cm);
```

Określanie współrzędnych punktów

Początek układu współrzędnych jest domyślnie w lewym dolnym rogu rysunku (sytuacja ulega zmianie, gdy użyjemy ujemnych współrzędnych).

Współrzędne kartezjańskie: $(1, 2)$

Współrzędne biegunowe: $(30:1\text{cm})$

Przy rysowaniu ścieżki można wyróżnić punkt, który jest punktem bieżącym. Kolejne punkty można podawać względem punktu bieżącego:

- $+(1, 0)$ – bez zmiany bieżącego punktu,
- $++(0, 2)$ – ze zmianą punktu bieżącego (staje się nim nowy wyliczony punkt).


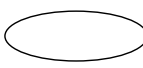





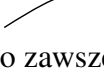
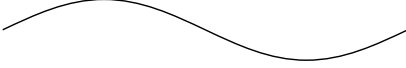
Współrzędne punktu można podać jako punkt przecięcia dwóch linii:

- $(p | - q)$ – przecięcie pionowej linii przechodzącej przez p i poziomej przechodzącej przez q , np.: $(30:1\text{cm} | - 0, 0)$
- **intersection of** $1, 0--1, 1$ **and** $0, 0--30:1\text{cm}$ – przecięcie dwóch odcinków

Punktom można nadawać nazwy:


```
\coordinate (A) at (0,0);
\coordinate [label=left:\textcolor{blue}{\$A\$}] (A) at (0,0);
\draw (-1,-1) -- (2,1) coordinate (B);
\draw (B) circle (1.2cm);
```

Rysowanie – wybrane kształty

- Okrąg:  `\draw (0,0) circle (7pt);`
- Elipsa:  `\draw (0,0) ellipse (20pt and 7pt);`
- Fragment okręgu – podajemy punkt początkowy łuku, kąt początkowy, kąt końcowy i promień:  `\draw (3mm,0mm) arc (0:110:3mm);`
- Fragment elipsy:  `\draw (0,0) arc (0:100:0.75cm and 0.5cm);`
- Prostokąt – podajemy lewy dolny i prawy górny narożnik:  `\draw (0,0) rectangle (0.5,0.5);`
- Parabola – o wierzchołku $(0,0)$ i przechodząca przez punkt $(0.7,0.5)$:  `\draw (0,0) parabola (0.7,0.5);`
- Krzywa bazująca na paraboli, ale z punktem przegięcia/wierzchołkiem w punkcie $(0.4,0.2)$:  `\draw (0,0) parabola bend (0.4,0.2) (0.8,0.4);`
- Sinusoida:  `\draw (0,0) sin (1,0.4);`
Polecenie to zawsze rysuje fragment sinusoidy z przedziału $(0, \pi/2)$ odpowiednio przeskalowanego.

`\draw (0,0) sin (1,0.3) cos (2,0) sin (3,-0.3) cos (4,0);`

Węzły

Polecenie `node` pozwala na dodanie węzła we wskazanym miejscu. W opcjach można

ustalić kształt dodawanego węzła, np.: 

```
\node (A) at (0,2) [shape=circle,draw] {$x_1$};
```

Ustawienie opcji `inner sep` pozwala zmniejszyć wewnętrzne marginesy, a przez to rozmiar węzła. Dostępne są również opcje `minimum width` i `minimum height`.

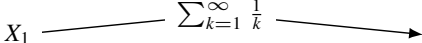
Każdy węzeł dostarcza szereg kotwic do mocowania połączeń zapisywanych jako `nazwa-węzła.nazwa-kotwicy`, np.: `A.center`, `A.west` itd. Definiując połączenie można wskazać konkretne kotwice: `\draw [->] (A.east) -- (B.west);`

Wskazywanie kotwic nie jest wymagane, system sam próbuje je dobrać.

Do rysowania połączeń można użyć polecenia `to`. W najprostszej wersji rysowana jest prosta. Wśród opcji można m.in. wskazać kąt wyjścia (`out`) i kąt wejścia (`in`) połączenia:

```
\draw [->] (A) to [out=135,in=45] (B);
```

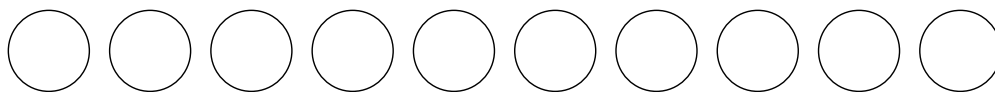
Węzeł z tekstem można dodać również jako element ścieżki:

x_1 

```
\draw[-latex] (0,0) node[fill=white] {$X_1$} -- (2,0.2)
node[fill=white] {$\sum_{k=1}^{\infty}\frac{1}{k}$} -- (4,0);
```

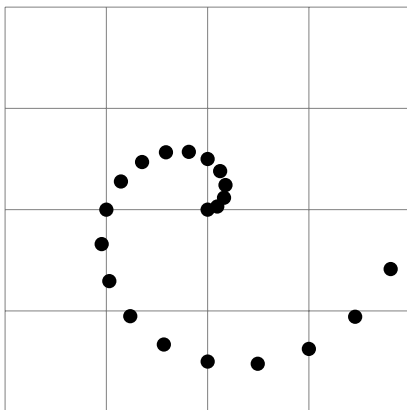
Pętle i obliczenia

Pętla `foreach`:



```
\tikz \foreach \x in {1,...,10} \draw (\x,0) circle (0.4cm);!
```

Po załadowaniu biblioteki `calc` można wykonywać obliczenia. Wyrażenia umieszczają się między znakami `$`:



```
\draw [help lines] (0,0) grid (4,4);
\foreach \i in {0,0.1,...,2}
  \fill ($ (2,2) !\i! \i*180:(3,2) $)
  circle (2pt);
```

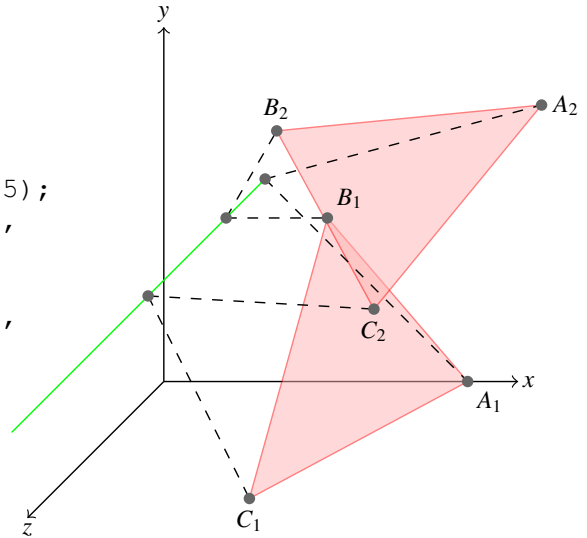
Uwaga: Zapis `(1,2)!.75!(3,4)` oznacza punkt 3/4 odległości między punktami `(1,2)` i `(3,4)`. Zapis `(1,2)!.75!60:(3,4)` oznacza, że przed wyliczeniem punktu w odległości 3/4 od `(1,2)`, najpierw obracamy odcinek względem punktu `(1,2)` o 60° .

Rysowanie w \mathbb{R}^3

```

1 \draw [->] (0,0,0) -- (3.5,0,0)
2   node [right] {$x$};
3 \draw [->] (0,0,0) -- (0,3.5,0)
4   node [above] {$y$};
5 \draw [->] (0,0,0) -- (0,0,3.5)
6   node [below] {$z$};
7 \draw [draw=green] (1,2,0) -- (1,2,6.5);
8 \filldraw [draw=red,fill=red!30!white,
9   opacity=0.5] (3,0,0) -- (2,2,1) --
10  (2,0,3) -- cycle;
11 \filldraw [draw=red,fill=red!30!white,
12   opacity=0.5] (3.73,2.73,0) --
13  (1.5,2.86,1) -- (3.23,1.87,3) --
14  cycle;
15 \draw[dashed] (1,2,3) -- (2,0,3)
16  (1,2,3) -- (3.23,1.87,3)
17  (1,2,0) -- (3,0,0)
18  (1,2,0) -- (3.73,2.73,0)
19  (1,2,1) -- (2,2,1)
20  (1,2,1) -- (1.5,2.86,1);

```

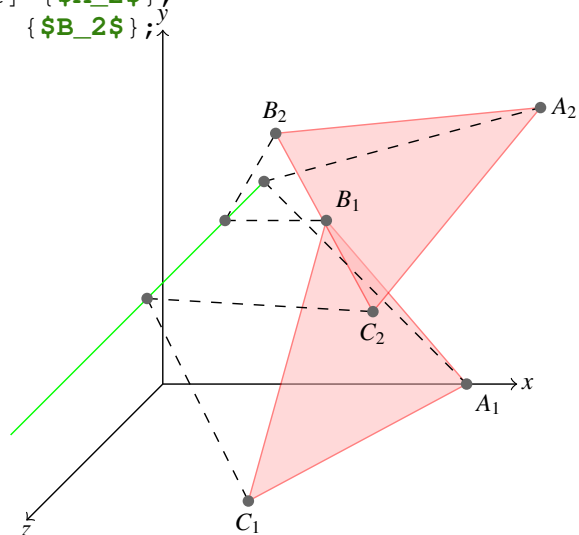


Rysowanie w \mathbb{R}^3

```

12 \node at (1,2,3) [circle,fill=grey] {};
13 \node at (1,2,0) [circle,fill=grey] {};
14 ...
15 \node at (2,0,3) [below=2pt] {$C_1$};
16 \node at (3,0,0) [below right=2pt] {$A_1$};
17 \node at (2,2,1) [above right=2pt] {$B_1$};
18 \node at (3.23,1.87,3) [below=2pt] {$C_2$};
19 \node at (3.73,2.73,0) [right=2pt] {$A_2$};
20 \node at (1.5,2.86,1) [above=2pt] {$B_2$};

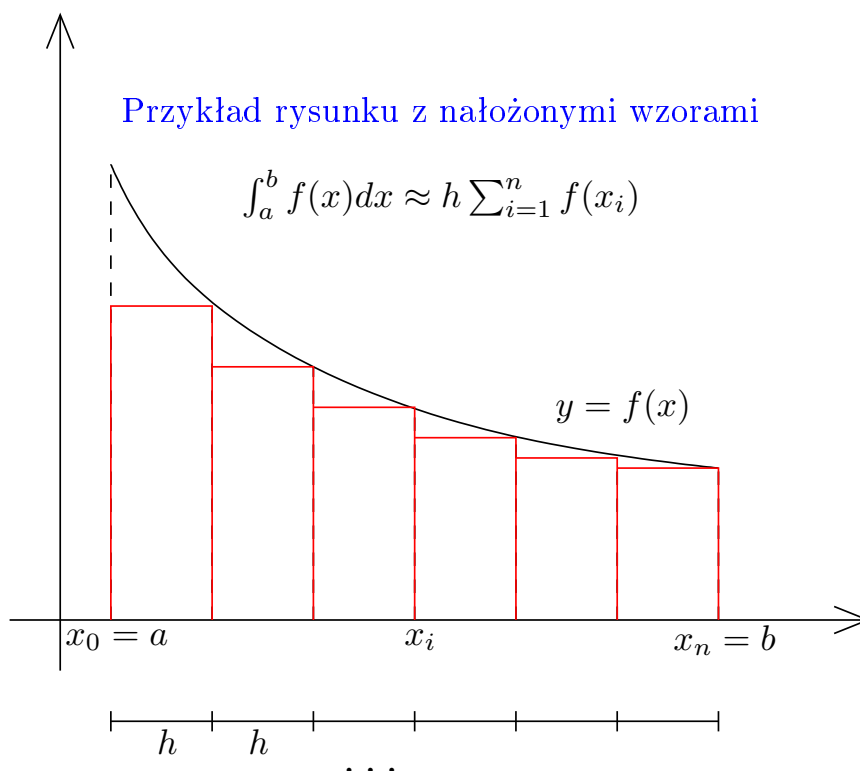
```

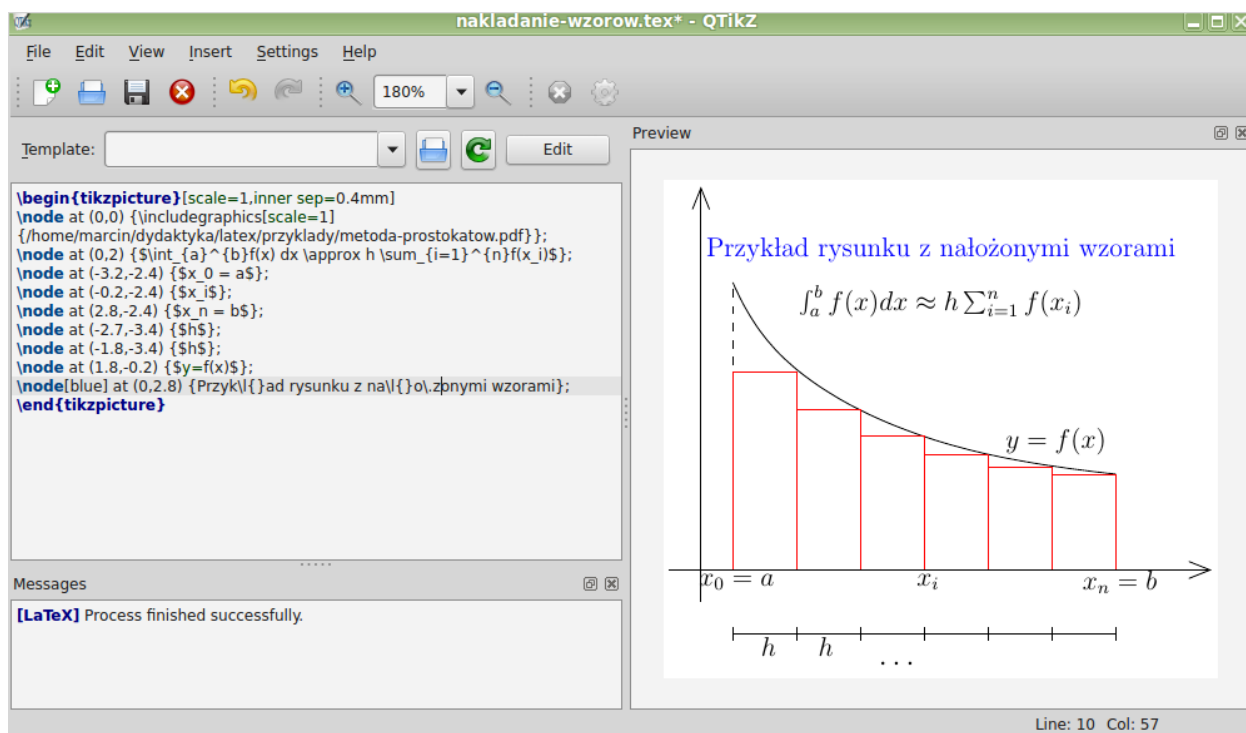


Nakładanie wzorów na rysunek

```
1 \documentclass{standalone}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \usepackage[polish]{babel}
5 \usepackage{tikz}
6
7 \begin{document}
8
9 \begin{tikzpicture}[scale=1,inner sep=0.4mm]
10 \node at (0,0) {\includegraphics[scale=1]{metoda-prostokatow}};
11 \node at (0,2) {\int_a^b f(x) dx \approx h \sum_{i=1}^n f(x_i)};
12 \node at (-3.2,-2.4) {\mathbf{x}_0 = a};
13 \node at (-0.2,-2.4) {\mathbf{x}_i};
14 \node at (2.8,-2.4) {\mathbf{x}_n = b};
15 \node at (-2.7,-3.4) {\mathbf{h}};
16 \node at (-1.8,-3.4) {\mathbf{h}};
17 \node at (1.8,-0.1) {\mathbf{y=f(x)}};
18 \node[blue] at (0,2.8) {Przykład rysunku z nałożonymi wzorami};
19 \end{tikzpicture}
20
21 \end{document}
```

Nakładanie wzorów na rysunek – wynik kompilacji





Style strony

\LaTeX pozwala wybrać za pomocą polecenia `\pagestyle{opcja}` jeden z trzech sposobów składania pagin:

- **plain** – pagina górna (nagłówek) pusta, dolna (stopka) z wycelowanym numerem,
- **headings** – pagina dolna pusta, górna z numerem strony i tytułem,
- **empty** – pagina górna i dolna puste.

Styl pojedynczej strony można ustawić poleceniem `\thispagestyle{opcja}`.

Pakiet **fancy** pozwala w łatwy sposób dostosować wygląd pagin. Zarówno nagłówek jak i stopka składają się w tym podejściu z trzech definiowalnych części. Można dodawać linie ozdobne, definiować paginy w kilkoma liniami tekstu, niezależnie definiować nagłówki i stopki dla stron parzystych i nieparzystych.

```

1 \usepackage{fancyhdr} % wszystko w preambule
2 \pagestyle{fancy}
3 \fancyhead{} %wyczyść wszystkie pola % R ight
4 \fancyhead[RO,LE]{\rightmark} % L eft
5 \fancyhead[LO,RE]{\leftmark} % C enter
6 \fancyfoot{} % E ven
7 \fancyfoot[LE,RO]{Strona \thepage} % O dd
8 \fancyfoot[CO,CE]{***}
9 \renewcommand{\headrulewidth}{0.4pt}
10 \renewcommand{\footrulewidth}{0.4pt}

```

`\leftmark` i `\rightmark` zawierają nazwę bieżącego rozdziału i sekcji.

ptaki4.tex

Polecenie `newcommand`

Polecenie `newcommand` pozwala na definiowanie własnych poleceń użytkownika.

```
\newcommand{nazwa} [liczba-argumentów] {tekst}
```

Przy definiowaniu polecenia bezargumentowego należy pominąć nawias kwadratowy. Liczba obowiązkowych argumentów musi być z przedziału 1–9. W części *tekst* wolno używać zarówno standardowych instrukcji L^AT_EXa, jak też zdefiniowanych przez użytkownika. Nie wolno używać poleceń `newcommand` i `newenvironment`, stosować w nazwach polskich liter diakrytycznych i rekursji.

```
1 \newcounter{zad}
2 \newcommand{\zadanie}{\addtocounter{zad}{1}
3   \medskip \noindent \textbf{\thezad. }}
4
5 \zadanie (2 pkt.) Dane są wektory  $(0, 1, 3)$ ,  $(1, -1, 2)$ ,
6  $(0, 0, 1)$ . Sprawdź czy tworzą one bazę przestrzeni  $\mathbb{R}^3$ .
7
8 \zadanie (2 pkt.) Podaj definicję przestrzeni afinicznej.
```

1. (2 pkt.) Dane są wektory $(0, 1, 3)$, $(1, -1, 2)$, $(0, 0, 1)$. Sprawdź czy tworzą one bazę przestrzeni \mathbb{R}^3 .

2. (2 pkt.) Podaj definicję przestrzeni afinicznej.

`newcommand_demo.tex, protokol.tex`

Skorowidz

Do wygenerowania skorowidza konieczne jest użycie pakietu `makeidx` oraz umieszczenie w preambule instrukcji `\makeindex`. Hasła do indeksu umieszcza się bezpośrednio w tekście poleceniem `\index{hasło}`. Skorowidz jest umieszczany w miejscu wstawienia instrukcji `\printindex`.

Najczęściej spotykaną w literaturze klasą sieci Petriego są `{\em sieci miejsc i przejść}`. `\index{siecx@sieć!miejsc i przejść}` `\index{PTsiecx@PT-sieć}` Są one podstawowym językiem modelowania współbieżności i~synchronizacji procesów dyskretnych ...

- Hasło indeksu powinno być umieszczane bezpośrednio po poprzedzającym je słowie, bez wstawiania dodatkowych spacji.
- `\index{miejsce}` – hasło pierwszego stopnia,
- `\index{siecx@sieć}` – hasło z poprawionym sortowaniem,
- `\index{miejsce!bezpieczne}` – hasło drugiego stopnia
- `\index{siecx@sieć!miejsc i przejść}` – hasło drugiego stopnia (poprawione sortowanie).

Wygenerowany indeks znajduje się w pliku o rozszerzeniu `ind`. Plik ten można w razie potrzeby *ręcznie* poprawić przed ostatnią kompilacją gotowego dokumentu.

`skorowidz.tex`

Klasa `aghdpl` – plik z główny

```
1 \documentclass[pdflatex]{aghdpl}
2 \usepackage[polish]{babel}
3 \usepackage{enumerate} % dodatkowe pakiety
4 \usepackage{listings}
5 \lstloadlanguages{TeX}
6
7 \author{Marcin Szpyrka}
8 \shortauthor{M. Szpyrka}
9 \titlePL{Przygotowanie pracy dyplomowej w systemie \LaTeX}
10 \titleEN{Thesis in \LaTeX}
11 % ...
12 \begin{document}
13 \titlepages
14 \tableofcontents
15 \clearpage
16
17 \include{rozdzial1}
18 \include{rozdzial2}
19 % ...
20 \appendix
21 \include{dodatekA}
22 \include{dodatekB}
23 \bibliography{bibfile}
24 \end{document}
```

Klasa `beamer`

- Klasa `beamer` służy do tworzenia prezentacji w \LaTeX u. Slajdy do tego wykładu zostały złożone z użyciem klasy `beamer`! Różnica między wersją do druku, a wersją wyświetlaną podczas wykładu to wyłącznie wynik włączenia kilku dodatkowych opcji na etapie kompilacji.
- Skompilowana prezentacja jest w formacie `pdf` co oznacza, że będzie wyglądała identycznie niezależnie od systemu użytego do jej wyświetlenia.
- Zastosowanie \LaTeX a do tworzenia prezentacji jest bardziej czasochłonne niż użycie narzędzi typu WYSIWYG, ale produkt jest zazwyczaj lepszej jakości. Możliwe jest stosowanie wszystkich poznanych możliwości \LaTeX a i łatwo jest przenieść treść np. z pracy dyplomowej do prezentacji na obronę.

```
1 \documentclass{beamer}
2 ...
3 \usetheme{Warsaw} % wybór stylu decydującego o wyglądzie
4 \title[Slajdy w beamerze]{Slajdy w beamerze -- przykłady}
5 \author[M. Szpyrka.]{Marcin Szpyrka}
6 \date[2013]{19.02.2013}
7 \institute[AGH-UST]
8 {Wydział EAIiIB\ Katedra Informatyki Stosowanej}
9
10 \begin{document}
11 ...
```

Slajdy

Pojedyncze slajdy budujemy korzystając ze środowiska `frame`.

```
1 \begin{frame}
2   \titlepage
3 \end{frame}
4
5 %-----
6
7 \begin{frame}[fragile]
8 \frametitle{...}
9
10  % zawartość slajdu
11 \end{frame}
```

Opcja `fragile` musi być stosowana, jeżeli slajd zawiera środowiska `verbatim` lub `lstlisting`. Slajdy w dokumencie mogą być grupowane w sekcje, podsekcje itd. Wyświetlanie informacji o aktualnej sekcji (podsekcji itp.) jest zależne od wybranego stylu. Jeśli jest taka potrzeba za pomocą polecenia `tableofcontents` można wstawić spis treści. Każde umieszczenie na slajdzie polecenia `\pause` powoduje podział slajdu na kolejne sceny. Wyświetlanie slajdu zatrzymuje się na poleceniu `\pause`. Po wybraniu kolejnego slajdu (PgDn) następuje wyświetlenie kolejnego fragmentu (do następnego polecenia `\pause` lub do końca slajdu, gdy brak już `\pause`).

Nakładki

Poza stosowaniem polecenia `\pause`, beamer dostarcza bardzo elastyczny system nakładek (`overlays`), które decydują o sposobie wyświetlania slajdu. W rezultacie slajd jest wyświetlany jako seria scen.

```
1 \begin{frame}
2 \frametitle{...}
3
4 \begin{itemize}
5 \item<1-3>\alert<2>{W RTCP-sieciach ...}
6 \item<2,4>\color<2>{blue}{W RTCP-sieciach ...}
7 \item<3-> W RTCP-sieciach ...
8 \item<-2> W RTCP-sieciach, każdy ...
9 \end{itemize}
10
11 \only<-2>{\alert{\textbf{RTCP-sieci}}}
12 \uncover<3->{CP-sieci vs. RTCP-sieci}
13
14 \end{frame}
```

Zastosowanie polecenia `\uncover` powoduje, że tekst zajmuje miejsce nawet, gdy nie jest widoczny. W przypadku użycia polecenia `\only`, niewyświetlany tekst nie zajmuje miejsca – należy uważać, aby zawartość slajdów *nie skakała*.

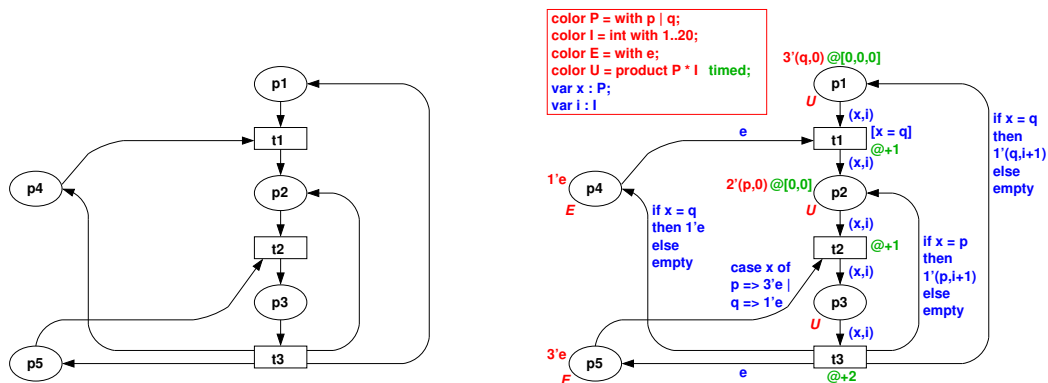
Jeżeli chcemy wyświetlać kolejne punkty jeden po drugim, to można zastosować konstrukcję `\begin{itemize}[<+>]` lub `\begin{itemize}[<+| alert@+>]`

Nakładki – grafika

```

1 \begin{frame}
2 \frametitle{Kolorowane sieci Petriego}
3
4 \centerline{
5 \only<1>{\includegraphics[width=7cm,angle=270]{cp2}}%
6 \only<2>{\includegraphics[width=7cm,angle=270]{cp3}}%
7 \only<3>{\includegraphics[width=7cm,angle=270]{cp4}}%
8 \only<4>{\includegraphics[width=7cm,angle=270]{cp5}}%
9 \end{frame}

```



Poprawne wyświetlanie tego typu slajdów wymaga, aby wszystkie obrazki miały ten sam rozmiar. Można to uzyskać np. stosując białą (niewidoczną) ramkę.

Bloki

Znakowanie

Znakowaniem sieci \mathcal{N} nazywamy dowolną funkcję M określoną na zbiorze miejsc sieci taką, że:

$$\forall p \in P: M(p) \in 2^{C(p)*}. \quad (12)$$

Rozkładem pieczętek czasowych sieci \mathcal{N} nazywamy dowolną funkcję S określoną na zbiorze miejsc sieci taką, że $\forall p \in P: S(p) \in \mathbb{R}$.

```

1 \begin{block}{Znakowanie}
2 {\em Znakowaniem} sieci  $\mathcal{N}$  nazywamy dowolną
3 funkcję  $M$  określoną na zbiorze miejsc sieci taką, że:
4 \begin{equation}
5 \label{eq:znakowanie}
6 \forall p \in P: M(p) \in 2^{C(p)*}.
7 \end{equation}
8 \end{block}
9 \begin{block}{}
10 {\em Rozkładem pieczętek czasowych} sieci  $\mathcal{N}$ 
11 nazywamy dowolną funkcję  $S$  określoną na zbiorze miejsc
12 sieci taką, że  $\forall p \in P: S(p) \in \mathbb{R}$ .
13 \end{block}

```

Alvis Language

- Communication Diagrams (AlvisCD)
- Alvis Code Language (AlvisCL)

Alvis Toolkit

- Alvis Editor
- Alvis Translator
- Alvis VM

```

1 \begin{columns}
2 \column{0.6\textwidth}
3 \begin{block}{Alvis Language}
4 \begin{itemize}
5 \item Communication Diagrams (AlvisCD)
6 \item Alvis Code Language (AlvisCL)
7 \end{itemize}
8 \end{block}
9 \column{0.2\textwidth}
10 \begin{block}{Alvis Toolkit}
11 \begin{itemize}
12 \item Alvis Editor
13 \item Alvis Translator
14 \item Alvis VM
15 \end{itemize}
16 \end{block}
17 \end{columns}

```

beamer_demo.tex, beamer_agh.tex

Pakiet `textpos`

```

1 \begin{textblock}{2}(11,1.5)
2 \textcolor{red}{Haskell, Python, C, C++, Java}
3 \end{textblock}
4
5 \begin{textblock}{3}(8,3)
6 \textcolor{blue}{Haskell, Python, C, C++, Java}
7 \end{textblock}
8
9 \begin{textblock}{1}(10,8)
10 \textcolor{darkgreen}{Haskell, Python, C, C++, Java}
11 \end{textblock}
12
13 \begin{textblock}{7}(1,14)
14 \textcolor{orange}{Haskell, Python, C, C++, Java}
15 \end{textblock}

```

Haskell,
Python, C,
C++, Java

Haskell, Python,
C, C++, Java

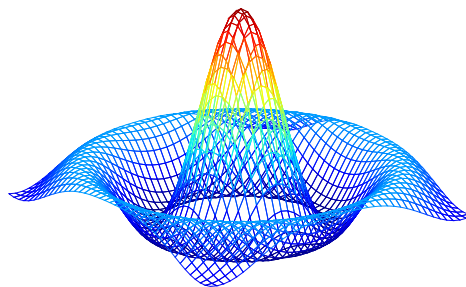
Haskell,
Py-
thon,
C,
C++,
Java

Polecenie `textblock` wykorzystuje swoje jednostki miary (podajemy tylko liczby). Pakiet dostarcza polecenia pozwalające je redefiniować. Pierwszy argument definiuje szerokość bloku, a dwa kolejne określają współrzędne lewego górnego narożnika.

Haskell, Python, C, C++, Java

Powiększanie fragmentu slajdu

```
x = y = linspace(-8, 8, 41)';  
[xx, yy] = meshgrid(x, y);  
z = sqrt(xx.^2 + yy.^2) + eps;  
z = sin(z) ./ z;  
mesh(x, y, z)
```

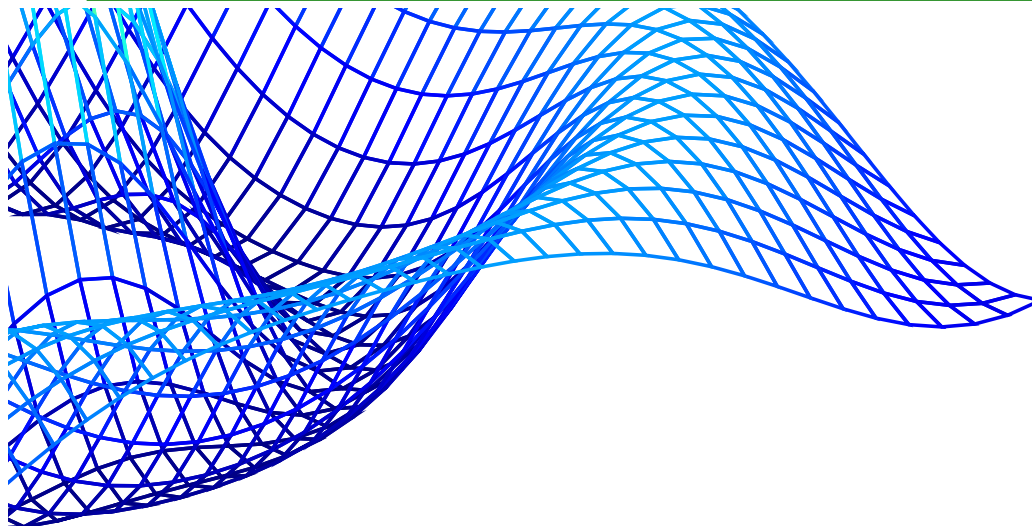


```
\framezoom<1><2>[border] (-0.3cm,0.6cm) (5.8cm,1.7cm)  
\framezoom<1><3>[border] (9cm,1.5cm) (2cm,1.5cm)
```

Pierwsza para argumentów wskazuje lewy górny narożnik ramki, a druga jej szerokość i wysokość. Punkt o współrzędnych (0,0) oznacza początek normalnego tekstu zawartości slajdu (nie bierzemy pod uwagę tytułu).

Powiększanie fragmentu slajdu

```
x = y = linspace(-8, 8, 41)';  
[xx, yy] = meshgrid(x, y);  
z = sqrt(xx.^2 + yy.^2) + eps;  
z = sin(z) ./ z;  
mesh(x, y, z)
```

Ramki i odległości

Ramka jest częścią dokumentu, która jest traktowana jako całość i nie można jej dzielić na wiersze lub strony.

- `\framebox`[długość][pozycja]{tekst} – parametr *długość* można pominąć (rozmiar ramki będzie do pasowany do tekstu), parametr *pozycja* może przyjąć jedną z wartości *c*, *l*, *r*, *s*. Wartość *s* powoduje rozciągnięcie tekstu (jeśli się da) do zadanej szerokości.

Ala ma kota.

- `\makebox`[długość][pozycja]{tekst} – j.w., ale bez rysowania krawędzi.
- `\fbox`{tekst}, `\mbox`{tekst} – wersje bez parametrów opcjonalnych.
- `\hspace`{długość} – wstawia poziomy odstęp o zadanej długości, w wersji `\hspace*`{długość} może być użyty na końcu wiersza.
- `\vspace`{długość} – wstawia pionowy odstęp o zadanej długości, w wersji `\vspace*`{długość} może być użyty na granicy stron.
- `\dotfill` – wypełnienie przestrzeni kropkami,

Ala ma kota.

`\framebox`[0.6\textwidth][1]{Ala ma kota. `\dotfill`}
- `\textwidth` – szerokość strony (tekstu).
- `\fill` – elastyczne polecenie długości; `\hfill` jest równoważne `\hspace`{\fill}, zaś `\vfill` jest równoważne `\par\vspace`{\fill}.
- `\hrulefill` – wypełnienie przestrzeni linią ciągłą.

Środowisko `minipage`

Środowisko `minipage` pozwala na zdefiniowanie ramki, która jest przetwarzana w trybie akapitowym.

```
\begin{minipage} [pozycja1] [wysokość] [pozycja2] {szerokość}
...
\end{minipage}
```

- `pozycja1` – sposób wyrównania ramki względem środowiska, w którym jest umieszczona; `t` – linia bazowa pierwszego wiersza ramki wyrównana z linią bazową wiersza, w którym umieszczono ramkę; `b` – linia bazowa ostatniego wiersza ramki wyrównana z linią bazową wiersza, w którym umieszczono ramkę; `c` (domyślna wartość) – pionowe centrowanie ramki.
- `wysokość` – wysokość tworzonej ramki akapitowej.
- `pozycja2` – pozycja tekstu w ramce; `t` – dosunięcie do góry, `b` – dosunięcie do dołu, `c` – wycentrowanie, `s` – rozciągnięcie w pionie. Jeżeli wartość ta nie jest określona, to używana jest wartość `pozycja1`.
- `szerokość` – szerokość tworzonej ramki akapitowej (jedyne obowiązkowy parametr).

`minipage_demo.tex`

Minipage – przykład

Rozkładem pieczętek czasowych sieci \mathcal{N} nazywamy dowolną funkcję S określoną na zbiorze miejsc sieci taką, że $\forall p \in P: S(p) \in \mathbb{R}$.

Rozkładem pieczętek czasowych sieci \mathcal{N} nazywamy dowolną funkcję S określoną na zbiorze miejsc sieci taką, że $\forall p \in P: S(p) \in \mathbb{R}$.

```
1 \hfill
2 \begin{minipage} {5cm}
3 \begin{block} {}
4 {\em Rozkładem pieczętek czasowych} sieci  $\mathcal{N}$ 
5 nazywamy dowolną funkcję  $S$  określoną na zbiorze miejsc
6 sieci taką, że  $\forall p \in P: S(p) \in \mathbb{R}$ .
7 \end{block}
8 \end{minipage}
9
10 \centerline{
11 \begin{minipage} {5cm}
12 ... % to samo co wyżej
13 \end{minipage}}
```

\LaTeX dostarcza dwa środowiska do cytowania: `quote` – do krótkich cytowań i `quotation` – do dłuższych.

```
1 Poniższy przykładowy cytat zaczerpnięto z
2 pracy Karla Poppera (Popper 1992).
3 \begin{quote}
4 Ilekroć dana teoria wydaje ci się jedyna możliwa,
5 bierz to za znak, że nie zrozumiałeś ani teorii,
6 ani problemu, który miała rozwiązać.
7 \end{quote}
```

Poniższy przykładowy cytat zaczerpnięto z pracy Karla Poppera (Popper 1992).

Ilekroć dana teoria wydaje ci się jedyna możliwa, bierz to za znak, że nie zrozumiałeś ani teorii, ani problemu, który miała rozwiązać.

Środowisko `quotation` stosuje się analogicznie.

Pakiet `algorithmicx`

Pakiet `algorithmicx` jest jednym z kilku pakietów umożliwiających składanie w \LaTeX u pseudokodu. Umieszczenie środowiska `algorithmic` wewnątrz środowiska `algorithm` pozwala uzyskać wstawkę.

Algorithm 1 Euclid's algorithm

...

▷ Miejsce na pseudokod

```
1 \usepackage{algorithm}
2 \usepackage{algpseudocode}
3
4 \begin{algorithm}[ht] % [H] w przypadku beamera
5 \caption{Euclid's algorithm}
6 \label{alg:euclid}
7 \begin{algorithmic}
8 \State ... \Comment{Miejsce na pseudokod}
9 \end{algorithmic}
10 \end{algorithm}
```

algorithmicx – przykład

```
1: procedure EUCLID( $a, b$ ) ▷ The g.c.d. of  $a$  and  $b$ 
2:    $r \leftarrow a \bmod b$ 
3:   while  $r \neq 0$  do ▷ We have the answer if  $r$  is 0
4:      $a \leftarrow b$ 
5:      $b \leftarrow r$ 
6:      $r \leftarrow a \bmod b$ 
7:   end while
8:   return  $b$  ▷ The gcd is  $b$ 
9: end procedure
```

```
1 \begin{algorithmic}[1]
2 \Procedure{Euclid}{ $a, b$ }\Comment{The g.c.d. of  $a$  and  $b$ }
3   \State  $r$ \gets  $a \bmod b$ 
4   \While{ $r \neq 0$ }\Comment{We have the answer if  $r$  is 0}
5     \State  $a$ \gets  $b$ 
6     \State  $b$ \gets  $r$ 
7     \State  $r$ \gets  $a \bmod b$ 
8   \EndWhile
9   \State \textbf{return}  $b$ \Comment{The gcd is  $b$ }
10 \EndProcedure
11 \end{algorithmic}
```

Pakiet sudoku

L^AT_EX oferuje mnóstwo nawet bardzo nietypowych pakietów :-)
Wystarczy poszukać!

```
1 \setlength\sudokusize{5cm}
2
3 \begin{sudoku}
4 |2|5| | |3| |9| |1|.
5 | |1| | |4| | | |.
6 |4| |7| | |2| |8|.
7 | | |5|2| | | | |.
8 | | | |9|8|1| | |.
9 | |4| | | |3| | | |.
10 | | | |3|6| | |7|2|.
11 | |7| | | | | |3|.
12 |9| |3| | | |6| |4|.
13 \end{sudoku}
```

2	5			3		9		1
	1				4			
4		7				2		8
		5	2					
				9	8	1		
	4				3			
			3	6			7	2
	7							3
9		3				6		4