

Automatyczne publikowanie skryptów w programie MATLAB®

Damian Kowalów

Spis treści

1. Wprowadzenie	2
2. Przygotowanie kodu do publikacji	3
2.1. Dzielenie kodu na komórki	3
2.2. Wykonywanie kodu podzielonego na komórki	3
2.3. Korzystanie ze znaczników	5
2.4. Przykłady użycia znaczników	7
2.5. Wymuszone pobranie danych z wyjścia	12
3. Konfiguracja publikacji	14
3.1. Dostępne opcje w konfiguracji	15
4. Przykład w generowaniu plików publikacji	18
4.1. Przykład skryptu bez dodania znaczników	18
4.2. Przykład skryptu po dodaniu znaczników	20
5. Publikowanie z linii poleceń	22
6. Przydatne linki	22

1. Wprowadzenie

Tekst ten został oparty o pomoc programu MATLAB, do której referencja znajduje się w rozdziale 6.

Środowisko MATLAB dostarcza użytkownikowi możliwość publikowania utworzonego kodu programu, skryptu bądź funkcji do dokumentu mogącego posłużyć za prezentację wyników pracy. Zaimplementowane zostały najbardziej popularne obecnie formaty plików:

- \LaTeX ,
- PDF,
- pliki dokumentów Microsoft Word (.doc),
- pliki prezentacji Microsoft PowerPoint (.ppt),
- HTML,
- XML.

Opcje publikacji mają możliwość załączenia samego kodu programu uzupełnionego o komentarze, jak również wyników jego działania np. rysunków czy informacji pojawiających się w oknie poleceń. Zaletą automatycznej generacji dokumentów jest to, że mogą one zostać odpowiednio sformatowane oraz uzupełnione o tytuł, spis treści czy równania w notacji \LaTeX już na poziomie pisania skryptu. Dzięki takiemu zabiegowi użytkownik, który poprawnie sformatuje kod nie musi dodatkowo poświęcać czasu na edycję innych dokumentów, wklejanie rysunków czy wpisywanie komentarzy do kolejnych części kodu.

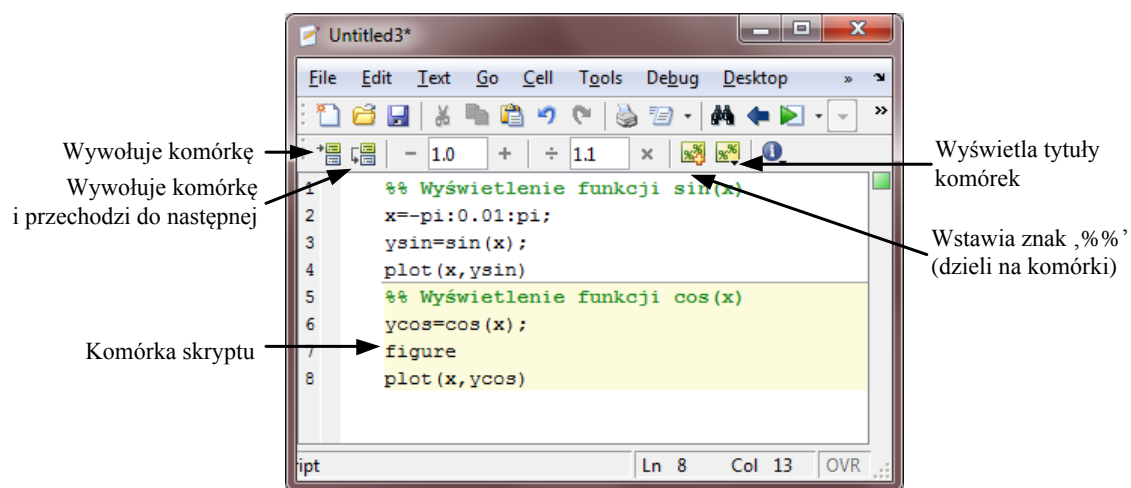
W celu uruchomienia procesu publikowania kodu pliku napisanego w programie MATLAB należy użyć przycisku znajdującego się w oknie edytora, użyć polecenia **File->Publish** z menu lub wykonać go poprzez funkcję `publish` w oknie poleceń uzupełniając ją o odpowiednie argumenty.

2. Przygotowanie kodu do publikacji

2.1. Dzielenie kodu na komórki

Kod skryptu można podzielić na części (Code Cells) a następnie wykonywać je w dowolnej kolejności, np. gdy nie ma potrzeby wykonywania całego pliku a tylko jego określoną część (np. wyświetlenie wykresu). Dzielenie kodu na części uzyskuje się poprzez zastosowanie podwójnego znaku procenta '%%' (rys. 2.1) na początku linii, która ma oddzielać kolejne części kodu.

Treść skryptu można podzielić poprzez kliknięcie w Menu->Cell->Insert Cell Break, kliknięcie prawym przyciskiem myszy w pole edytora a następnie wybranie opcji Insert Cell Break lub korzystając z przycisku dostępnego w edytorze.



Rys. 2.1. Przykład dzielenia kodu na sekcje

2.2. Wykonywanie kodu podzielonego na komórki

Fragment kodu podzielonego na komórki można wykonać poprzez umieszczenie kursora w polu danej komórki - będzie to potwierdzone zmianą koloru pola z białego na pomarańczowe, a następnie skorzystanie z dwóch dostępnych opcji w menu:

- Cell->Evaluate Current Cell (Ctrl+Enter) - wykonuje kod zawarty w aktywnej komórce,
- Cell->Evaluate Current Cell and Advance (Ctrl+Shift+Enter) - wykonuje kod zawarty w aktywnej komórce oraz dodatkowo przechodzi automatycznie do następnej w ko-

lejności.

Dodatkowo w pasku narzędzi dla trybu dzielenia na komórki znajdują się kolejno znaki $-$, $+$, \div oraz \mathbf{x} . Służą one do zmiany wartości zmiennej, która będzie zaznaczona w kodzie programu. Należy zauważyć, że w obrębie aktywnej komórki po każdej zmianie wartości, kod wykonuje się automatycznie. Efekt działania tej funkcji zaprezentowano na rys. 2.2. W szybki sposób zmieniono maksymalną wartość dla wektora \mathbf{x} z 2π na 3π , natomiast rysunek prezentujący wynik pojawił się automatycznie. Jeżeli kod miałby za zadanie tylko obliczyć wartość sinus należałoby dodać dodatkową komórkę i umieścić znak `%%` w linii poprzedzającej funkcję `plot`.

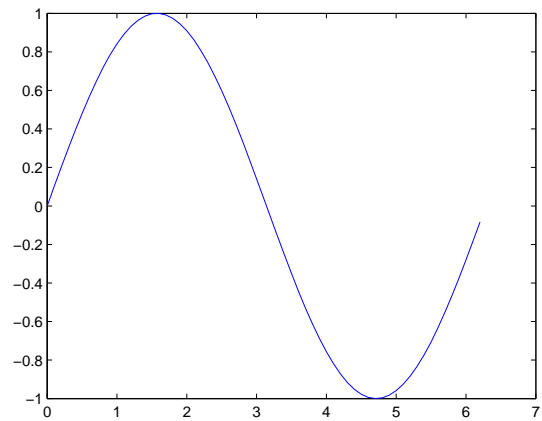
Kolejną funkcjonalnością edytora jest możliwość wykonywania tylko zaznaczonego przez użytkownika fragmentu kodu. Jeżeli istotne jest np. tylko wyświetlenie ponownie wykresu wystarczy zaznaczyć linię lub obszar w skrypcie, który jest za to odpowiedzialny a następnie skorzystać z polecenia `Evaluate Selection` klikając prawym przyciskiem myszy lub używając klawisza F9.

```

1 %% Wyświetlanie wykresu sin
2 x=0:0.1:2*pi;
3 y=sin(x);
4 plot(x,y);
5

```

(a) Kod skryptu z zaznaczoną zmienną odpowiedzialną za zakres dla wartości x



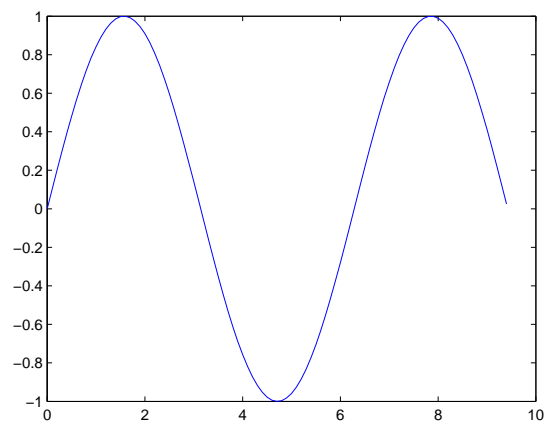
(b) Efekt działania

```

1 %% Wyświetlanie wykresu sin
2 x=0:0.1:3*pi;
3 y=sin(x);
4 plot(x,y);
5

```

(c) Kod skryptu po naciśnięciu przycisku '+'



(d) Efekt działania

Rys. 2.2. Przykład zastosowania inkrementacji zmiennej przy wykonywaniu skryptu

2.3. Korzystanie ze znaczników

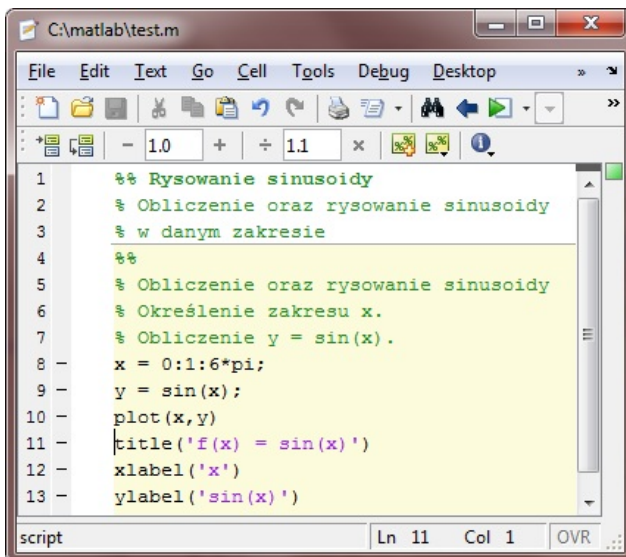
MATLAB umożliwia użytkownikowi dodanie w formie komentarzy różnego rodzaju znaczników, które w znacznym stopniu ułatwiają interpretację utworzonego dokumentu. Podstawowe opcje edycji tekstu wymienione są w tabeli 1, można wprowadzić je ręcznie, poprzez kliknięcie prawym przyciskiem myszy w oknie edytora a następnie wybranie opcji **Insert Text Markup** lub poprzez menu: **Cell->Insert Text Markup**.

Tab. 1. Znaczniki używane w edytorze

Opis znacznika	Przykład znacznika
Tytuł dokumentu i jego opis	%% TYTUŁ DOKUMENTU % Opis dokumentu
Tytuł sekcji i opis	%% TYTUŁ SEKCJI % Opis sekcji
Tytuł sekcji bez podziału na komórki	%%% TYTUŁ SEKCJI % Opis sekcji
Czcionka pogrubiona	% *pogrubiony tekst*
Czcionka pochyła	% _pochylony tekst_
Czcionka o stałej szerokości	% tekst
Hiperłącze	% <http://www.mathworks.com MathWorks>
Symbol TM	% MathWorks(TM)
Symbol [®]	% MathWorks(R)
Rysunek	% % << plik.png >> %
Lista punktowa	% * element % * element %
Lista numerowana	% # element % # element %
Znacznik HTML	% <html> % <table border=1><tr> % <td>one</td> % <td>two</td></tr></table> % </html>
Równanie w formacie L ^A T _E X	% \$\$ \frac{\sin(k)}{k} \$\$
Preformatowalny tekst	% To jest blok tekstu komentarza, % który jest widoczny w publikowanym % dokumencie. Rozpoczyna się od podwójnej % spacji
Składnia kodu z komentarza	%% % Należy rozpocząć od podwójnego % znaku procenta a kod po 3 spacjach % for i = 1:10 % disp x % end

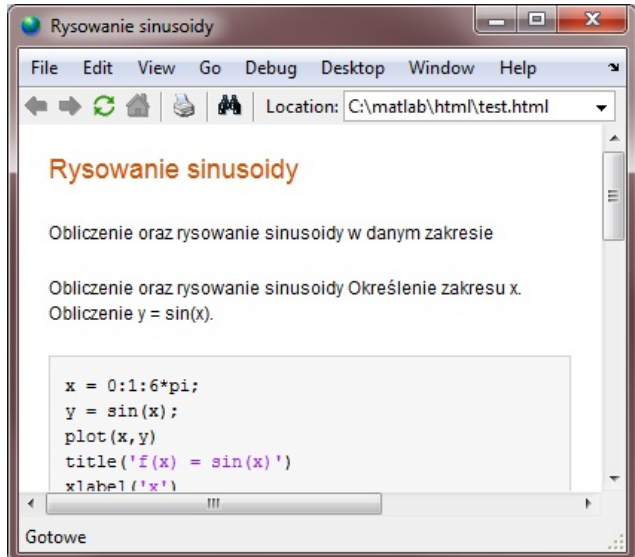
2.4. Przykłady użycia znaczników

Przykład użycia znaczników pokazano w formie rysunków prezentujących kod programu oraz jego efekt po włączeniu opcji publikowania. W celu poprawnego działania znacznika należy pamiętać o pozostawianiu pustych linii, stosowaniu znaku podwójnego procenta lub odpowiednim rozpoczęciu pisania w wierszu (ilość spacji).



```
1 %% Rysowanie sinusoidy
2 % Obliczenie oraz rysowanie sinusoidy
3 % w danym zakresie
4 %%
5 % Obliczenie oraz rysowanie sinusoidy
6 % Określenie zakresu x.
7 % Obliczenie y = sin(x).
8 -
9 x = 0:1:6*pi;
10 y = sin(x);
11 plot(x,y)
12 title('f(x) = sin(x)')
13 xlabel('x')
14 ylabel('sin(x)')
```

(a) Kod skryptu



Rysowanie sinusoidy

Obliczenie oraz rysowanie sinusoidy w danym zakresie

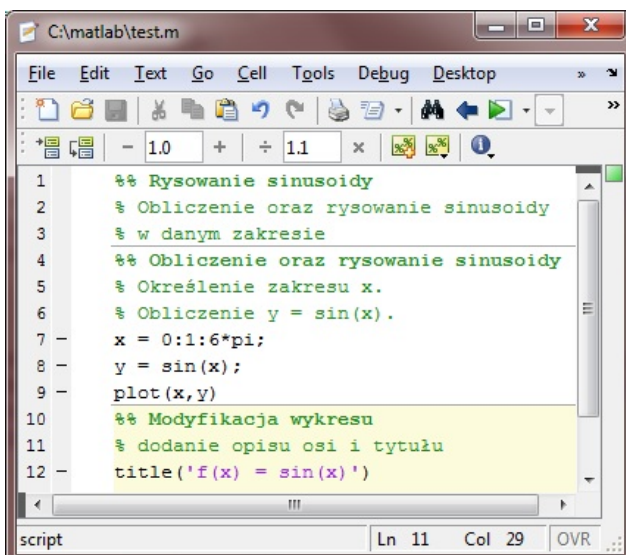
Obliczenie oraz rysowanie sinusoidy Określenie zakresu x.
Obliczenie y = sin(x).

```
x = 0:1:6*pi;
y = sin(x);
plot(x,y)
title('f(x) = sin(x)')
xlabel('x')
```

Gotowe

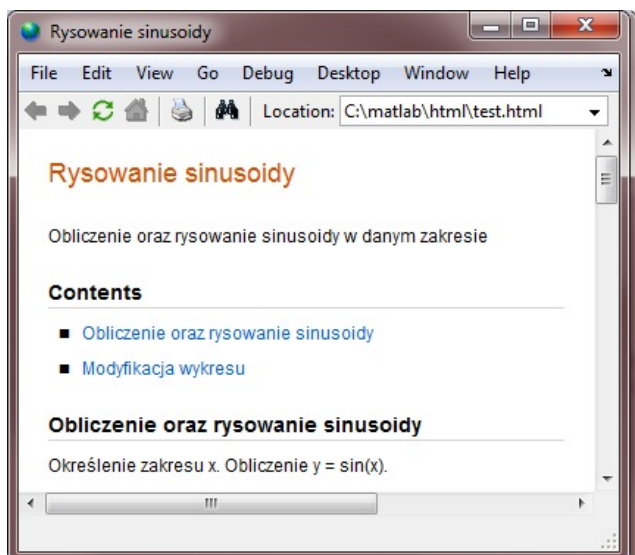
(b) Efekt publikacji

Rys. 2.3. Tytuł i opis dokumentu



```
1 %% Rysowanie sinusoidy
2 % Obliczenie oraz rysowanie sinusoidy
3 % w danym zakresie
4 %% Obliczenie oraz rysowanie sinusoidy
5 % Określenie zakresu x.
6 % Obliczenie y = sin(x).
7 -
8 x = 0:1:6*pi;
9 y = sin(x);
10 plot(x,y)
11 %% Modyfikacja wykresu
12 % dodanie opisu osi i tytułu
13 title('f(x) = sin(x)')
```

(a) Kod skryptu



Rysowanie sinusoidy

Obliczenie oraz rysowanie sinusoidy w danym zakresie

Contents

- [Obliczenie oraz rysowanie sinusoidy](#)
- [Modyfikacja wykresu](#)

Obliczenie oraz rysowanie sinusoidy

Określenie zakresu x. Obliczenie y = sin(x).

(b) Efekt publikacji

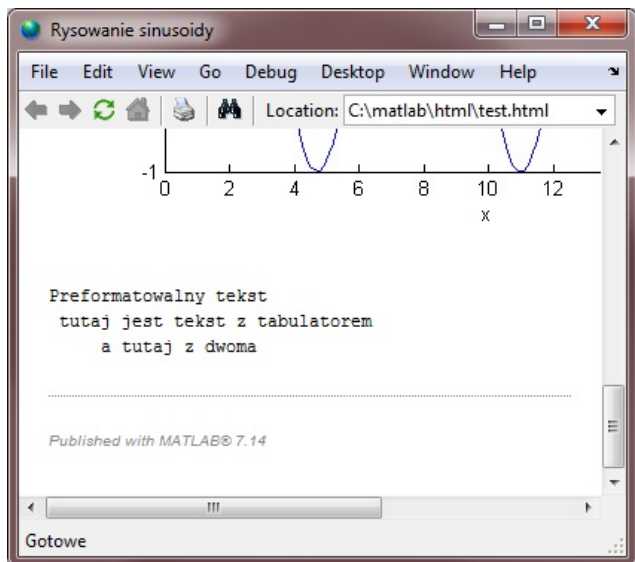
Rys. 2.4. Podrozdział i automatyczny spis treści

```

8 - x = 0:0.1:6*pi;
9 - y = sin(x);
10 - plot(x,y)
11 - title('f(x) = sin(x)')
12 - xlabel('x')
13 - ylabel('sin(x)')
14
15 %%
16 %
17 % Preformatowalny tekst
18 % tutaj jest tekst z tabulatorem
19 % |a tutaj z dwoma
20 %

```

(a) Kod skryptu



(b) Efekt publikacji

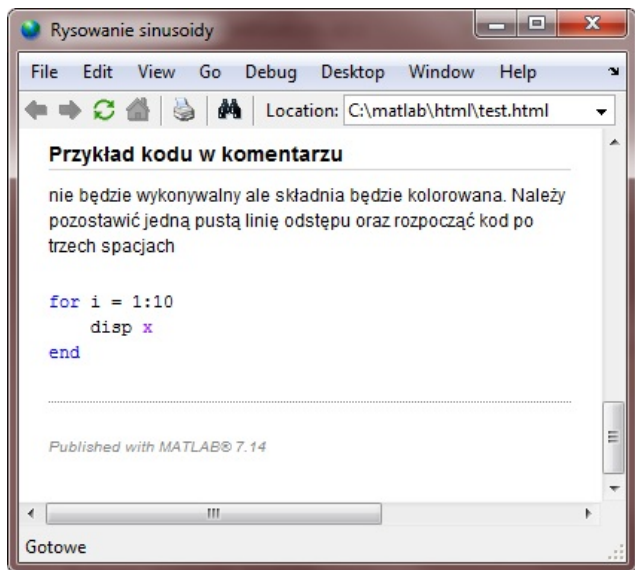
Rys. 2.5. Preformatowalny tekst

```

13 - ylabel('sin(x)')
14 %% Przykład kodu w komentarzu
15 % nie będzie wykonywalny ale składnia
16 % będzie kolorowana. Należy pozostawić
17 % jedną pustą linię odstępu oraz
18 % rozpocząć kod po trzech spacjach
19 %
20 % |for i = 1:10
21 % |disp x
22 % |end
23 %
24
25

```

(a) Kod skryptu



(b) Efekt publikacji

Rys. 2.6. Kolorowanie składni kodu z komentarza, który nie jest wykonywany


```
29 %% Lista nienumerowana
30 %%
31 %% * A
32 %% * B
33 %% * C
34 %% nowa linia
35 %%
36 %% nowa linia, która nie jest częścią
37 %% listy
38
39
40
41
```

(a) Kod skryptu

Rysowanie sinusoidy

Location: C:\matlab\html\test.html

Lista nienumerowana

- A
- B
- C nowa linia

nowa linia, która nie jest częścią listy

Published with MATLAB® 7.14

Gotowe

(b) Efekt publikacji

Rys. 2.7. Lista nienumerowana

```
29 %% Lista numerowana
30 %%
31 %% # A
32 %% # B
33 %% # C
34 %% |
35
36
37
38
39
40
41
```

(a) Kod skryptu

Rysowanie sinusoidy

Location: C:\matlab\html\test.html

```
for i = 1:10
    disp x
end
```

Lista numerowana

1. A
2. B
3. C

Published with MATLAB® 7.14

Gotowe

(b) Efekt publikacji

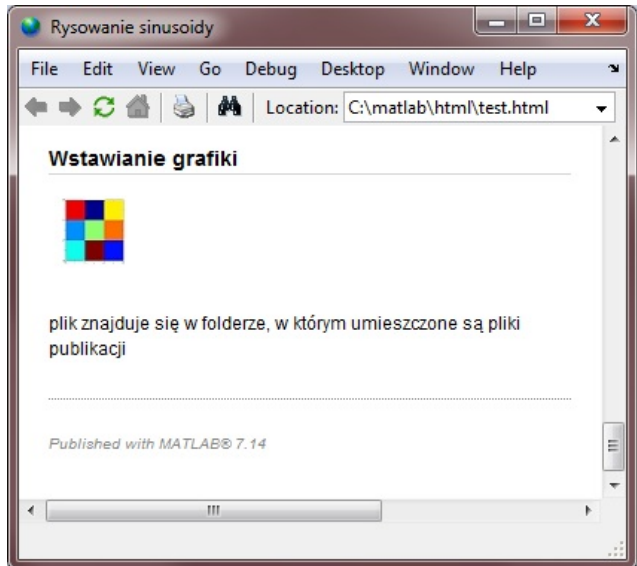
Rys. 2.8. Lista numerowana

```

C:\matlab\test.m
File Edit Text Go Cell Tools Debug Desktop
40
41 %% Wstawianie grafiki
42 %
43 % <<test1.jpg>>
44 %
45 % plik znajduje się w folderze,
46 % w którym umieszczone są
47 % pliki publikacji|
48
49
50
51
52
script Ln 47 Col 19 OVR

```

(a) Kod skryptu



(b) Efekt publikacji

Rys. 2.9. Wstawienie grafiki zewnętrznej

```

C:\matlab\test.m
File Edit Text Go Cell Tools Debug Desktop
59
60 %% Znacznik LaTeX
61 %
62 % <latex>
63 % \begin{tabular}{|c|c|} \hline
64 % $n$ & $n!$ \\ \hline
65 % 1 & 1 \\
66 % 2 & 2 \\
67 % 3 & 6 \\ \hline
68 % \end{tabular}
69 % </latex>
70 %
script Ln 60 Col 19 OVR

```

(a) Kod skryptu

Znacznik \LaTeX

n	$n!$
1	1
2	2
3	6

(b) Efekt publikacji

Rys. 2.10. Wstawienie znacznika \LaTeX (efekt widoczny tylko dla dokumentu stworzonego w środowisku \LaTeX)

```

78 %
79 %% Przykłady wzorów matematycznych
80 % Wzór który jest częścią tekstu
81 % i znajduje się w lini tekstu
82 % $f(x)=\frac{\sin(x)}{x}$ natomiast,
83 % to jest przykład wzoru, który
84 % nie występuje w nowej linii:
85 %
86 % $$e^{\pi i} + 1 = 0$$ |
87 %
88 % tu zaczyna się kolejna linia
89 %

```

(a) Kod skryptu

Rysowanie sinusoidy

Location: /C:/matlab/html/test.html#12

Przykłady wzorów matematycznych

Wzór który jest częścią tekstu i znajduje się w lini tekstu $f(x) = \frac{\sin x}{x}$ natomiast, to jest przykład wzoru, który nie występuje w nowej linii:

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

tu zaczyna się kolejna linia

Published with MATLAB® 7.14

(b) Efekt publikacji

Rys. 2.11. Wstawienie wzoru w notacji L^AT_EX

```

82 %
83 %% Przykład urzycia czcionek
84 %%
85 % *Czcionka pogrubiona*
86 % *_czcionka pochyla_
87 % |Czcionka o stałej szerokości|
88 % *_łączenie czcionek_*
89 % *_inny przykład_|
90 %
91 %
92 %
93 %

```

(a) Kod skryptu

Rysowanie sinusoidy

Location: /C:/matlab/html/test.html#10

tu zaczyna się kolejna linia

Przykład urzycia czcionek

- Czcionka pogrubiona
- czcionka pochyla
- Czcionka o stałej szerokości
- łączenie czcionek
- inny przykład

Published with MATLAB® 7.14

(b) Efekt publikacji

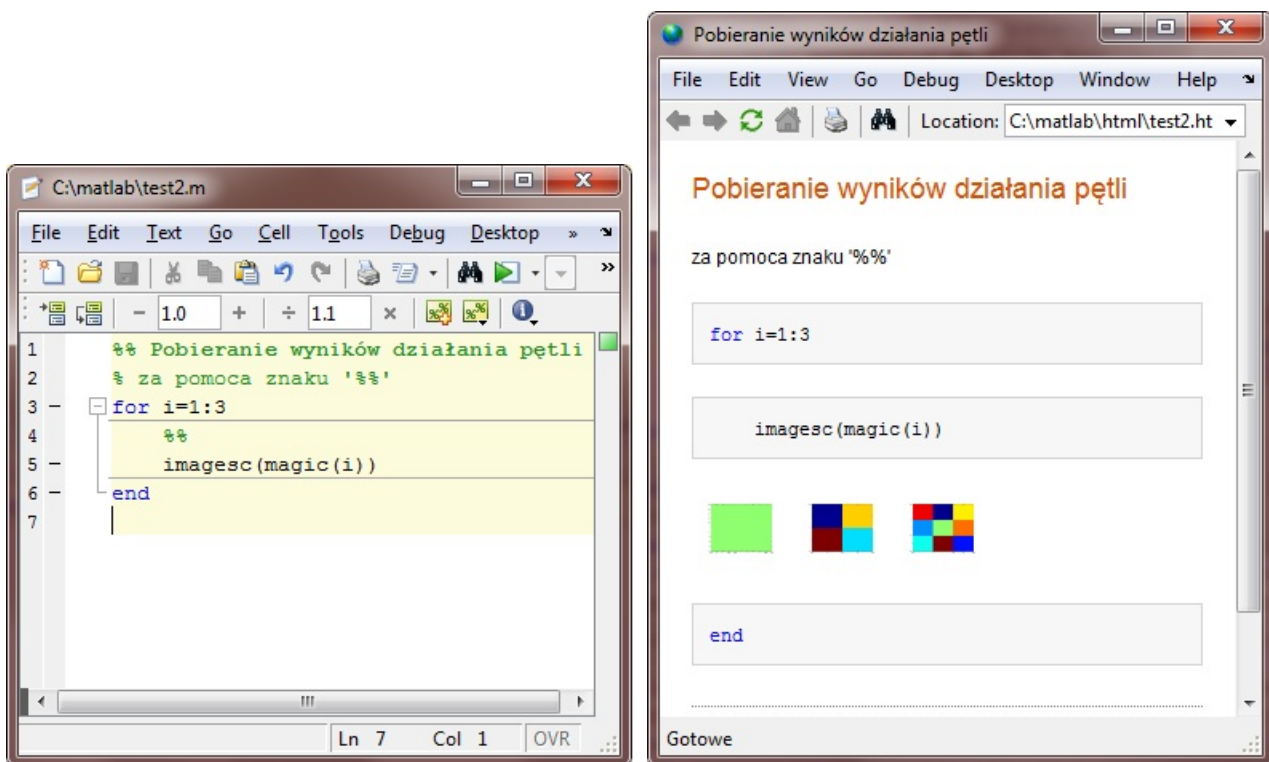
Rys. 2.12. Przykład użycia czcionek

2.5. Wymuszone pobranie danych z wyjścia

Podczas wykonywania się kodu iteracyjnie, w pętli lub nadpisywaniu wykresu w programie, w wyniku otrzymywana jest tylko jego ostatnia wersja, dlatego aby umożliwić użytkownikowi załączenie do pliku publikacji wszystkich wcześniej wykonywanych poleceń można skorzystać z dwóch sposobów:

- (a) użycie podwójnego znaku procenta,
- (b) użycie polecenia `snapshot`.

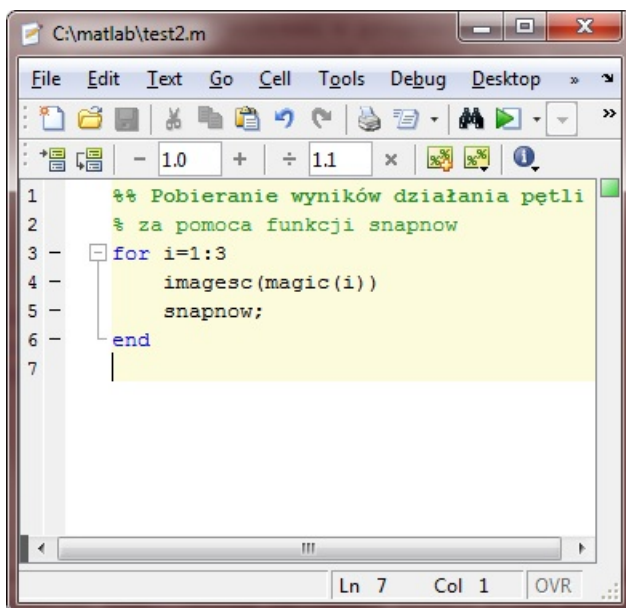
Przykład użycia przedstawiono na rys. 2.13 oraz 2.14.



(a) Kod skryptu

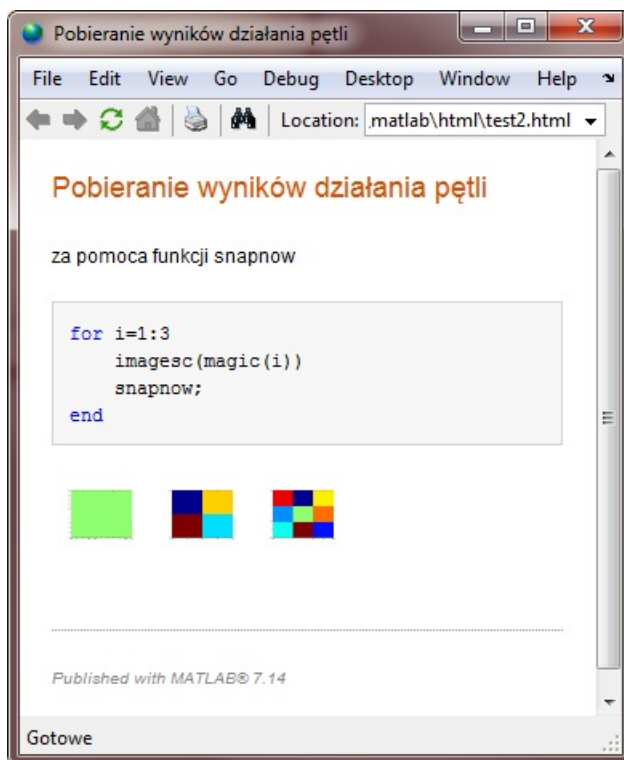
(b) Efekt publikacji

Rys. 2.13. Wymuszenie pobrania danych z wyjścia za pomocą znaku '%%'



```
1 %% Pobieranie wyników działania pętli
2 % za pomocą funkcji snapnow
3 for i=1:3
4     imagesc(magic(i))
5     snapnow;
6 end
7
```

(a) Kod skryptu



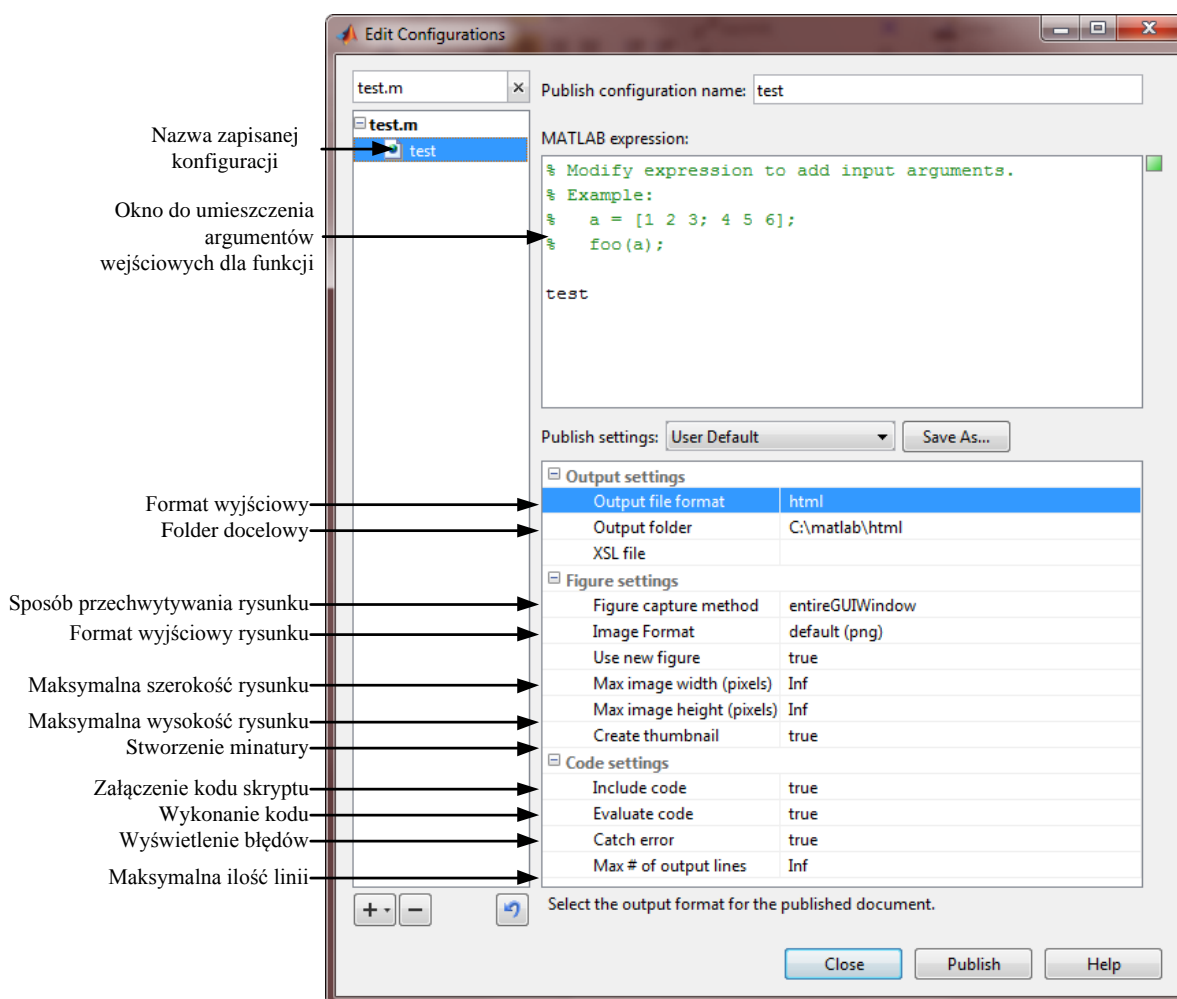
(b) Efekt publikacji

Rys. 2.14. Wymuszenie pobrania danych z wyjścia za pomocą funkcji snapnow

Główna różnica polega na tym, że podczas użycia znaku procenta kod programu jest rozdzielony podczas publikacji natomiast przy użyciu funkcji snapnow zachowuje spójną całość.

3. Konfiguracja publikacji

Po przygotowaniu kodu programu poprzez dodanie komentarzy oraz znaczników jest on gotowy do publikacji. Najszybszym sposobem jest wygenerowanie raportu z ustawieniami domyślnymi, jednak metoda ta sprawdza się tylko w przypadku gdy nie ma potrzeby podawania argumentów wejściowych dla funkcji oraz dokumentem wynikowym ma być plik HTML. W innych okolicznościach, gdy należy zdefiniować format wyjściowy, folder w którym ma znajdować się plik raportu, formaty obrazków itp. należy skorzystać z dostępnych opcji w narzędziu do publikowania. Aby uruchomić opcję publikacji należy wybrać z menu **File->Publish Configuration for...** a następnie **Edit Publish Configuration for...** Włączenie spowoduje pojawienie się okna konfiguracji (rys. 3.1).



Rys. 3.1. Okno konfiguracji narzędzia Publish

3.1. Dostępne opcje w konfiguracji

- (a) `Publish configuration name` (ang. nazwa konfiguracji) - istnieje możliwość utworzenia kilku różnych konfiguracji publikacji dokumentu oraz zapisania ich pod dowolną nazwą, jest to użyteczne szczególnie wtedy gdy plikiem wynikowym mają być różne dokumenty o różnych ustawieniach
- (b) `MATLAB expresion` (ang. okno poleceń) - w celu opublikowania funkcji dla konkretnych atrybutów wejściowych możliwe jest wpisanie ich w tym oknie a następnie uruchomienie procesu publikacji, np:
- ```
x=0:0.1:2*pi;
f_cos_sin(x)
```
- (c) `Output file format` (ang. format pliku wyjściowego) - określa w jakim formacie ma być plik wynikowy publikacji, należy zwrócić uwagę, że wybranie formatu wynikowego wiąże się z wybraniem formatu obrazków pod względem ich wspierania przez określony rodzaj dokumentów (np. HTML nie jest w stanie standardowo wyświetlić obrazków w formacie .pdf).
- (d) `Output folder` (ang. folder wyjściowy) - określa nazwę oraz miejsce na dysku twardym gdzie mają znajdować się opublikowane pliki.
- (e) `Figure capture method` (ang. metoda przechwytywania rysunków) - dekoracja okna określa czy takie elementy jak menu oka, tytuł okna, pasek narzędzi oraz obramowanie okna będą widoczne w wynikowym pliku publikacji. Poszczególne opcje przedstawia tabela 2

**Tab. 2.** Opcje przechwytywania rysunków

| Opcja                           | Dekoracja okna |              | Kolor tła    |              |
|---------------------------------|----------------|--------------|--------------|--------------|
|                                 | GUI            | Rysunek      | GUI          | Rysunek      |
| <code>entireGUIWindow</code>    | uwzględniona   | wykluczona   | wg. ustawień | białe        |
| <code>print</code>              | wykluczona     | wykluczona   | białe        | białe        |
| <code>getframe</code>           | wykluczona     | wykluczona   | wg. ustawień | wg. ustawień |
| <code>entireFigureWindow</code> | uwzględniona   | uwzględniona | wg. ustawień | wg. ustawień |

(f) `Image format` (ang. format rysunku) - opcje przedstawione są w tabeli 3

**Tab. 3.** Dostępne formaty plików rysunków

| Format                                   | Obsługiwane formaty rysunków                                                                                                                                                                                                                                                                       | Domyślny format rysunków                                                                                                                                   |
|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>doc</code>                         | każdy, który Microsoft Office może obsługiwać, czyli: <code>png</code> , <code>jpg</code> , <code>bmp</code> , oraz <code>tiff</code> . Przy opcji <code>'print'</code> również <code>eps</code> , <code>epsc</code> , <code>eps2</code> , <code>ill</code> , <code>meta</code> , <code>pdf</code> | <code>png</code>                                                                                                                                           |
| <code>html</code>                        | każdy format jest publikowany poprawnie, jednak należy sprawdzić czy narzędzie do otwierania pliku wynikowego jest w stanie wyświetlić dany format pliku z rysunkiem                                                                                                                               | <code>png</code>                                                                                                                                           |
| <code>L<sup>A</sup>T<sub>E</sub>X</code> | każdy format jest publikowany poprawnie                                                                                                                                                                                                                                                            | <code>epsc2</code> chyba, że wybrano opcję <code>getframe</code> , <code>entireFigureWindow</code> lub <code>entireGuiWindow</code> wtedy <code>png</code> |
| <code>pdf</code>                         | <code>bmp</code> lub <code>jpg</code>                                                                                                                                                                                                                                                              | <code>bmp</code>                                                                                                                                           |
| <code>ppt</code>                         | każdy, który Microsoft Office może obsługiwać                                                                                                                                                                                                                                                      |                                                                                                                                                            |
| <code>xml</code>                         | każdy format jest publikowany poprawnie, jednak należy sprawdzić czy narzędzie do otwierania pliku wynikowego jest w stanie wyświetlić dany format pliku z rysunkiem                                                                                                                               | <code>png</code>                                                                                                                                           |

(g) `Max image width (pixels)` (ang. maksymalna szerokość rysunku) - określa maksymalną szerokość rysunku (w pikselach), który zostanie załączony do pliku wynikowego publikacji, im mniejsza szerokość tym więcej rysunków może zmieścić się w jednym wierszu.

(h) `Max image height (pixels)` (ang. maksymalna wysokość rysunku) - analogicznie jak dla szerokości rysunku

(i) `Include code` (ang. załączenie kodu) - określenie czy kod skryptu ma znaleźć się w pliku wynikowym publikacji. Wyłączenie tej opcji spowoduje wyświetlenie samych znaczników określonych w pliku oraz elementów wyjściowych (figur, wyników prezentowanych w oknie poleceń).



- (j) **Evaluate code** (ang. wykonanie kodu) - określenie czy kod skryptu ma być wykonany do publikacji. Wyłączenie tej opcji spowoduje wyświetlenie samych znaczników określonych w pliku oraz kodu programu.
- (k) **Catch error** (ang. wyświetlenie błędów) - włączenie tej opcji spowoduje wyświetlenie błędów i ostrzeżeń pojawiających się w oknie poleceń.

## 4. Przykład w generowaniu plików publikacji

W tej części opisano przykładowe sposoby tworzenia dokumentacji na podstawie skryptów programu MATLAB. Ze względu na fakt, że widok dokumentów jest zbliżony w każdym z wynikowych formatów, w przykładach posłużono się formatem HTML, jednak najlepsze efekty pracy oraz manipulacji na pliku wynikowym można uzyskać stosując generację do plików  $\text{\LaTeX}$ .

### 4.1. Przykład skryptu bez dodania znaczników

Skrypt 1 przedstawia przykład kodu programu bez zastosowania znaczników. Pomimo, iż w kodzie występują komentarze dla poszczególnych części, wynikowy plik raportu (rys. 4.1) nie jest przejrzysty oraz nie przedstawia wyników kolejnych iteracji pętli `for`.

---

#### Skrypt 1 Skrypt bez zastosowania znaczników

---

```
function fourier_demoo
 t = 0:.1:pi*4; % przedział czasu
 y = sin(t); % wyliczenie funkcji sin
 updatePlot(t,y); % funkcja dodatkowa rysująca wykres

 % Przy każdej iteracji pętli określ nieparzystą harmoniczną k,
 % która będzie zwiększana w podanym zakresie
 for k = 3:2:9
 % Wykonaj obliczenie podane poniżej dla każdej z iteracji
 y = y + sin(k*t)/k;
 disp(['Gdy k = ', num2str(k)]);
 disp('Wykres wygląda następująco')
 % Odśwież wykres po każdej zmianie harmoniczej
 updatePlot (t,y)
 end

end

% Efekt Gibbsa objawia się tym, że w niektórych punktach wartość odtworzonego
% sygnału może przekroczyć wartość sygnału oryginalnego,
% powodując niepożądane, a nawet groźne zniekształcenia

function updatePlot(t,x)
% Funkcja dodatkowa do aktualizowania wykresu
 cla
 plot(t,x)
end
```

---



Rys. 4.1. Raport wygenerowany na podstawie skryptu 1

## 4.2. Przykład skryptu po dodaniu znaczników

Skrypt 1 uzupełniono o tytuł, podzielono go na sekcje oraz dodano formatowanie tekstu na podstawie wzorców zawartych w tabeli 1. Uzyskano dzięki temu przejrzysty dokument prezentujący wszystkie wykresy uzyskane podczas działania pętli. Należy zwrócić uwagę na formatowanie kodu: spacje, odstępy, puste linie itp.

---

### Skrypt 2 Skrypt po zastosowaniu znaczników

---

```
%% Uzyskiwanie przebiegu kwadratowego z przebiegu sinusoidalnego
% Kod skryptu prezentuje zasadę działania szeregu Furiera
% poprzez aproksymacja fali prostokątnej z użyciem programu MATLAB(R)
%% Zdefiniowanie nieparzystych harmonicznych oraz wyświetlenie wykresu
function fourier_demo
 t = 0:.1:pi*4; % przedział czasu
 y = sin(t); % wyliczenie funkcji sin
 updatePlot(t,y); % funkcja dodatkowa rysująca wykres

 %%
 % Przy każdej iteracji pętli określ nieparzystą harmoniczną
 % _k_, która będzie zwiększana w podanym zakresie
 for k = 3:2:9
 %%
 % Wykonaj obliczenie podane poniżej dla każdej z iteracji
 %
 % $$ y = y + \frac{\sin(k*t)}{k} $$
 %
 y = y + sin(k*t)/k;
 disp(['Gdy k = ', num2str(k)]);
 disp('Wykres wygląda następująco')
 updatePlot (t,y)
 end
end

%% Efekt Gibbsa
% Efekt objawia się tym, że w niektórych punktach wartość odtworzonego
% sygnału może przekroczyć wartość sygnału oryginalnego,
% powodując niepożądane, a nawet _groźne_ zniekształcenia

function updatePlot(t,x)
% Funkcja dodatkowa do aktualizowania wykresu
 cla
 plot(t,x)
end
```

---

**Tytuł** → Uzyskiwanie przebiegu kwadratowego z przebiegu sinusoidalnego

**Opis kodu** → Kod skryptu prezentuje zasadę działania szeregu Fouriera poprzez aproksymacja fali prostokątnej z użyciem programu MATLAB®

**Spis treści** →

- Zdefiniowanie nieparzystych harmonicznych oraz wyświetlenie wykresu
- Efekt Gibbsa

**Zdefiniowanie nieparzystych harmonicznych oraz wyświetlenie wykresu**

**Kod** →
`function fourier_demo

t = 0:1:pi*4; % przedział czasu
y = sin(t); % wyliczenie funkcji sin
updatePlot(t,y); % funkcja dodatkowa rysująca wykres`

**Wykres** →

Przy każdej iteracji pętli określi nieparzystą harmoniczną k, która będzie zwiększana w podanym zakresie
`for k = 3:2:9`

Wykonaj obliczenie podane poniżej dla każdej z iteracji

$$y = y + \frac{\sin(k \cdot t)}{k}$$

```

y = y + sin(k*t)/k;
disp(['Gdy k = ', num2str(k)]);
disp('Wykres wygląda następująco');
updatePlot(t,y)

```

**Wzór** →

**Wyniki z okna poleceń** →

Gdy k = 7  
Wykres wygląda następująco

Gdy k = 9  
Wykres wygląda następująco

**Kolejna funkcja** →
`function updatePlot(t,x)
% Funkcja dodatkowa do aktualizowania wykresu
cla
plot(t,x)
end`

Published with MATLAB® 7.14

Rys. 4.2. Raport wygenerowany na podstawie skryptu 2

## 5. Publikowanie z linii poleceń

MATLAB umożliwia użytkownikowi publikowanie dokumentów z poziomu linii poleceń, za pomocą funkcji `publish` z odpowiednimi argumentami:

- `publish(file)`  
np. `publish(test.m)` generuje raport z pliku `test.m` dla domyślnych ustawień.
- `publish(file, format)`  
np. `publish(test.m, 'doc')` generuje plik w formacie doc z pliku `test.m` przy domyślnych ustawieniach reszty parametrów.
- `publish(file, Name, Value)`  
np. `publish(test.m, 'figureSnapMethod', 'entireFigureWindow')` generuje plik w domyślnym formacie ale uwzględniając podane w poleceniu opcje.
- `publish(file, options)`  
należy najpierw zdefiniować strukturę ustawień np.:  

```
options_doc_nocode.format = 'doc';
options_doc_nocode.showCode = false;
```

a następnie wygenerować plik poleceniem: `publish('test.m', options_doc_nocode);`

Dokładny opis wszystkich opcji znajduje się w pomocy programu MATLAB dotyczącej hasła `publish`.

## 6. Przydatne linki

- Film prezentujący zastosowanie dzielenia kodu na komórki [http://www.mathworks.com/help/techdoc/matlab\\_env/f6-14058.html](http://www.mathworks.com/help/techdoc/matlab_env/f6-14058.html),
- Pomoc techniczna dotycząca funkcji `publish` [http://www.mathworks.com/support/2012a/matlab/7.14/demos/RapidCodeIterationUsingCells\\_viewlet\\_swf.html](http://www.mathworks.com/support/2012a/matlab/7.14/demos/RapidCodeIterationUsingCells_viewlet_swf.html),
- Film prezentujący zastosowanie funkcji `publish` [http://www.mathworks.com/support/2012a/matlab/7.14/demos/PublishingfromtheEditor\\_viewlet\\_swf.html](http://www.mathworks.com/support/2012a/matlab/7.14/demos/PublishingfromtheEditor_viewlet_swf.html),
- *Nie za krótkie wprowadzenie do systemu  $\LaTeX 2_{\epsilon}$*  <ftp://sunsite.icm.edu.pl/pub/CTAN/info/lshort/polish/lshort2e.pdf>.