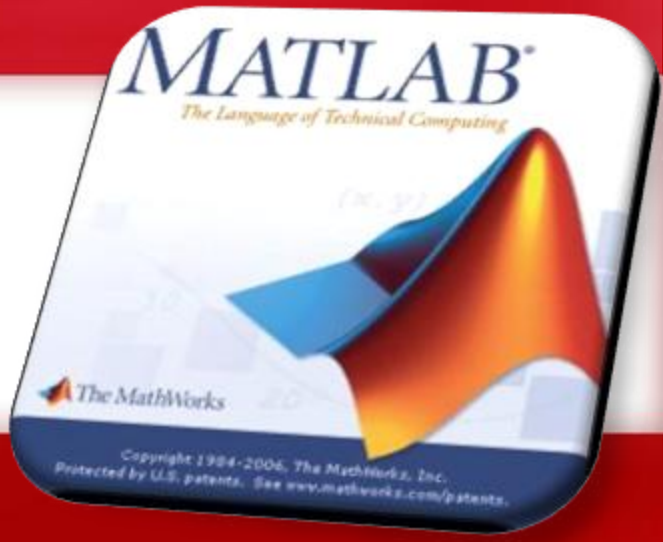
MATLAB



Dubacki Mariusz

WFAiIS 2012

PLAN

PLAN PREZENTACJI:

I. Kilka słów o Matlabie
II. Główne możliwości Matlaba
III. Pierwsze starcie, nawigacja po programie
IV. Tworzenie publikacji
V. Moduły Matlaba (toolboxy)
VI. GUI
VII. Profiler
VIII. Debuger
IX. Korzystanie z pomocy (Help)
X. Zastosowanie Matlaba, przykładowe programy

Kilka słów o Matlabie

- **Pakiet oprogramowania matematycznego firmy Mathworks Inc (www.mathworks.com)**
- **Interaktywne środowisko programistyczne**
- **Język obliczeń technicznych, język programowania wysokiego poziomu**
- **Narzędzie do szybkiej analizy i wizualizacji danych**
- **Nazwa pochodzi od słów MATrix LABoratory, gdyż początkowo program był przeznaczony do numerycznych obliczen macierzowych**
- **W Matlabie możliwe jest programowanie zorientowane obiektowo (klasy, metody, klasy abstrakcyjne, dziedziczenie !!!)**
- **Historia:**
  - **1980 - pre-początek Matlaba – Cleve Moler (Fortran)**
  - **1983 – Moler,Bangert, Little – rozwój projektu, Fortran->C, powstaje firma MathWorks**
  - **1985 – Pierwsza wersja programu**

Kilka słów o Matlabie

RODZAJE PLIKÓW

- **M-pliki (*.m) – są to główne pliki Matlaba (jak w C++ pliki *.cpp). Po wywołaniu m-pliku w którym znajduje się więcej funkcji wywoływana jest pierwsza funkcja w pliku**

- **Mex-pliki - funkcje napisane w C lub Fortranie możemy wywoływać w środowisku Matlab. MEX-plik jest to dynamicznie ładowana biblioteka (w Microsoft Windows są to pliki *.dll) którą można**
- **uruchomić z wiersza poleceń w oknie Matlaba tak jak zwykły m-plik. (http://gpsuj.zut.edu.pl/lab/zpowzis/pdf/Lab08_interfejsC.pdf)**

- **Mat-pliki (*.mat) – tymczasowe lub końcowe wyniki obliczeń możemy zapisywać do pliku tekstowego ASCII o dowolnej nazwie lub do pliku binarnego z rozszerzeniem *.mat – wówczas wszystkie liczby (całkowite i zmiennoprzecinkowe) zapisywane są w formacie zmiennoprzecinkowym z podwójną precyzją.**
- **PRZYKŁAD:**

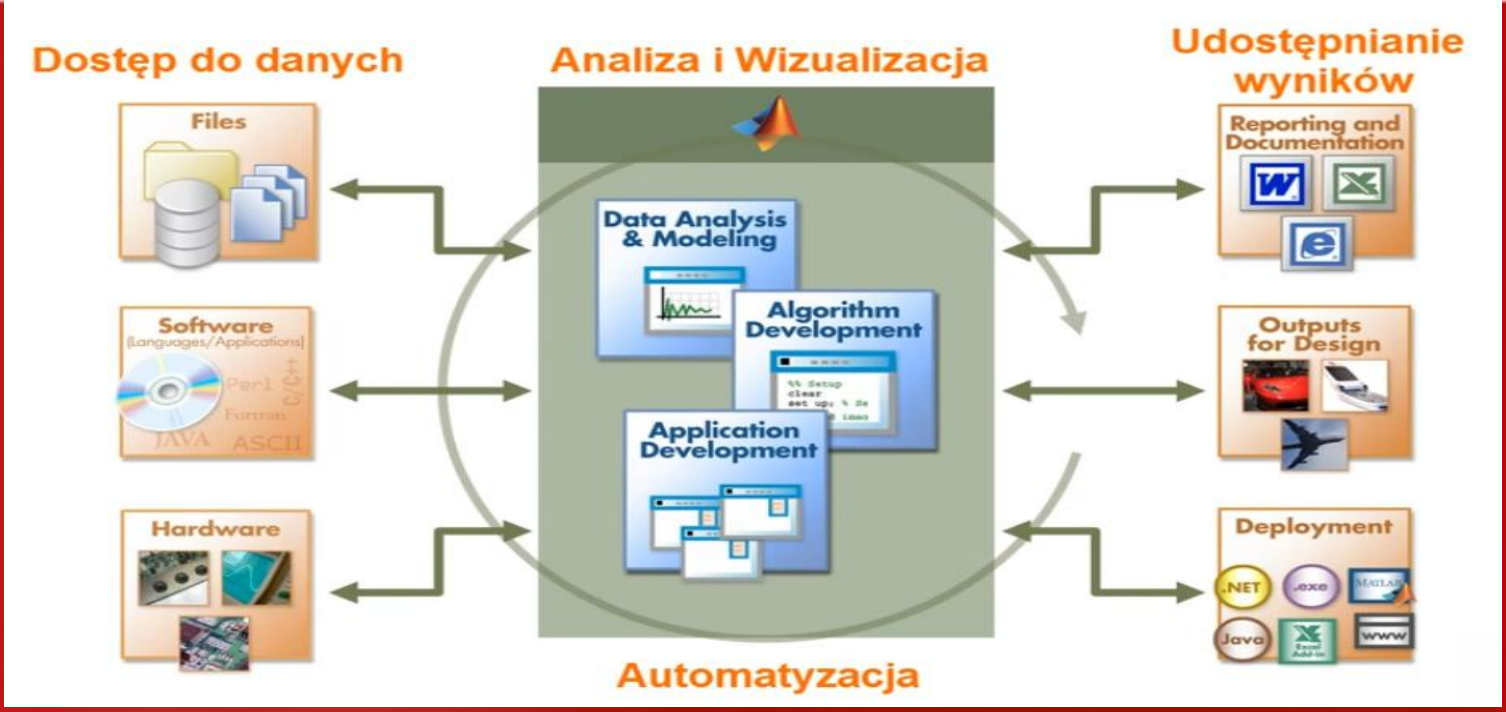
```
x=1:100;
y=sin(x);
save dwie_zmienne x y
```

Kilka słów o Matlabie

RODZAJE PLIKÓW

- **Pliki fig (*.fig) - Wykresy i inne formy graficzne wygenerowane w matlabie można zapisać do wybranego formatu graficznego lub do pliku binarnego z rozszerzeniem *.fig. Zaletą tej drugiej formy zapisu jest możliwość późniejszej modyfikacji zapisanego obiektu w programie.**

Główne możliwości Matlaba



1. Matlab stanowi wsparcie na każdym etapie pracy. Umożliwia łatwy import danych z pliku, import z baz danych, integracje z innymi aplikacjami, odczyt ze sprzetu pomiarowego takiego jak np oscyloskop.

2. Po zaimportowaniu danych można przejść do głównego etapu pracy, czyli tworzenie algorytmów i wizualizacji.

3. Ostatnim etap - udostępnianie wyników w formie raportu, kodu C, czy gotowej aplikacji okienkowej.

4. Wszystkie etapy maja miejsce w jednym środowisku, wiec łatwa jest automatyzacja i stopniowe udoskonalanie aplikacji

Główne możliwości Matlaba

W oknie poleceń Matlaba możemy wykonywać komendy systemowe:

Przykłady:

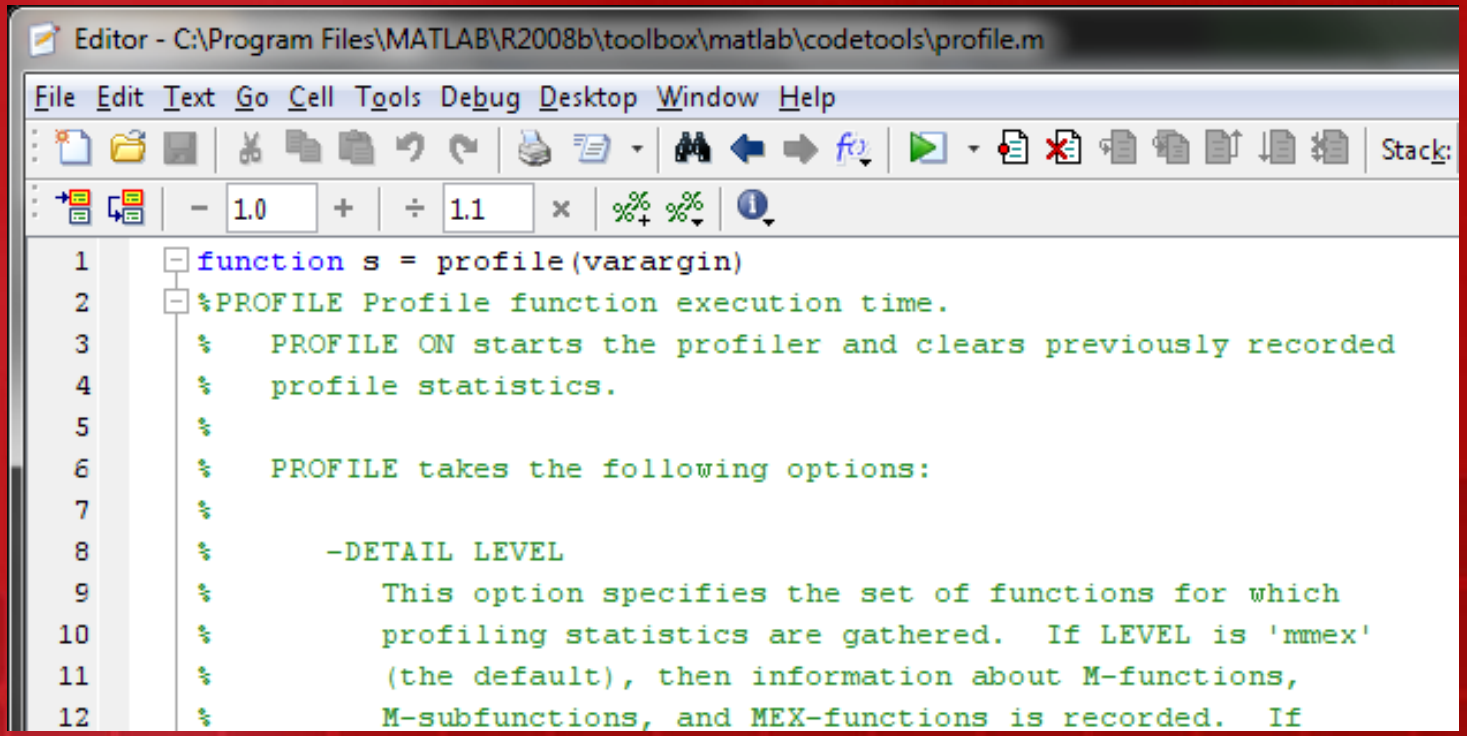
- **!dir - wylistuj zawartość (bieżącego) katalogu**
- **!cd - zmiana bieżącego katalogu roboczego**
- **!mkdir - tworzenie nowego katalogu**
- **!rmdir- usuniecie katalogu**

Główne możliwości Matlaba

W Matlabie funkcje z których korzystamy są jawnie widoczne dla użytkownika (dobrze, czy źle?)

Przykład:

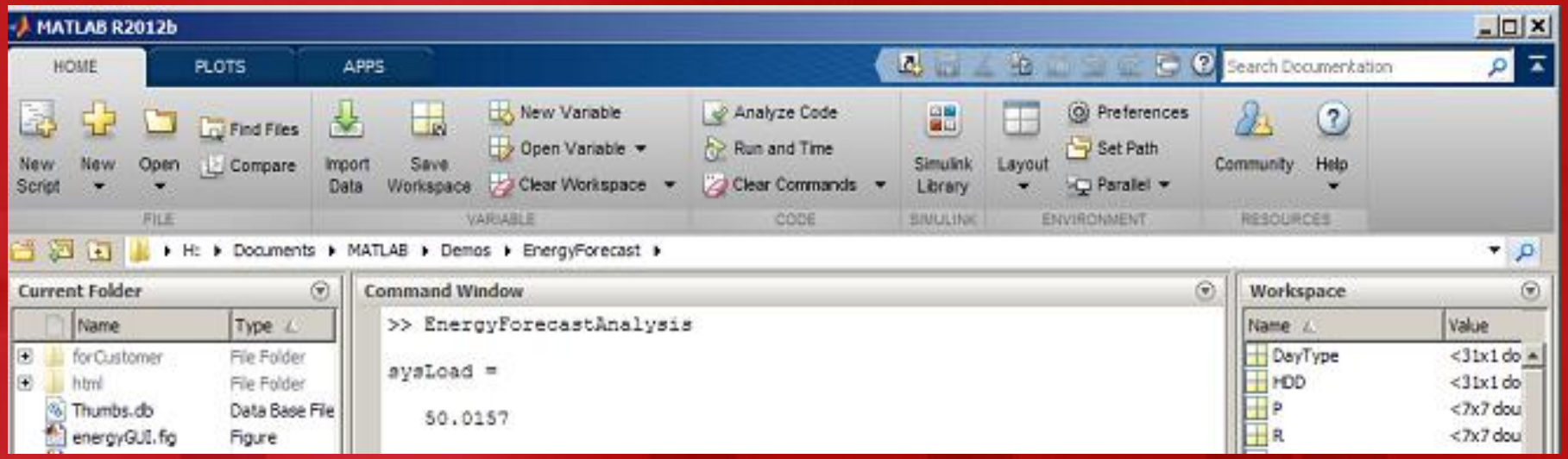
- **edit fplot**
- **edit profile**



Główne możliwości Matlaba

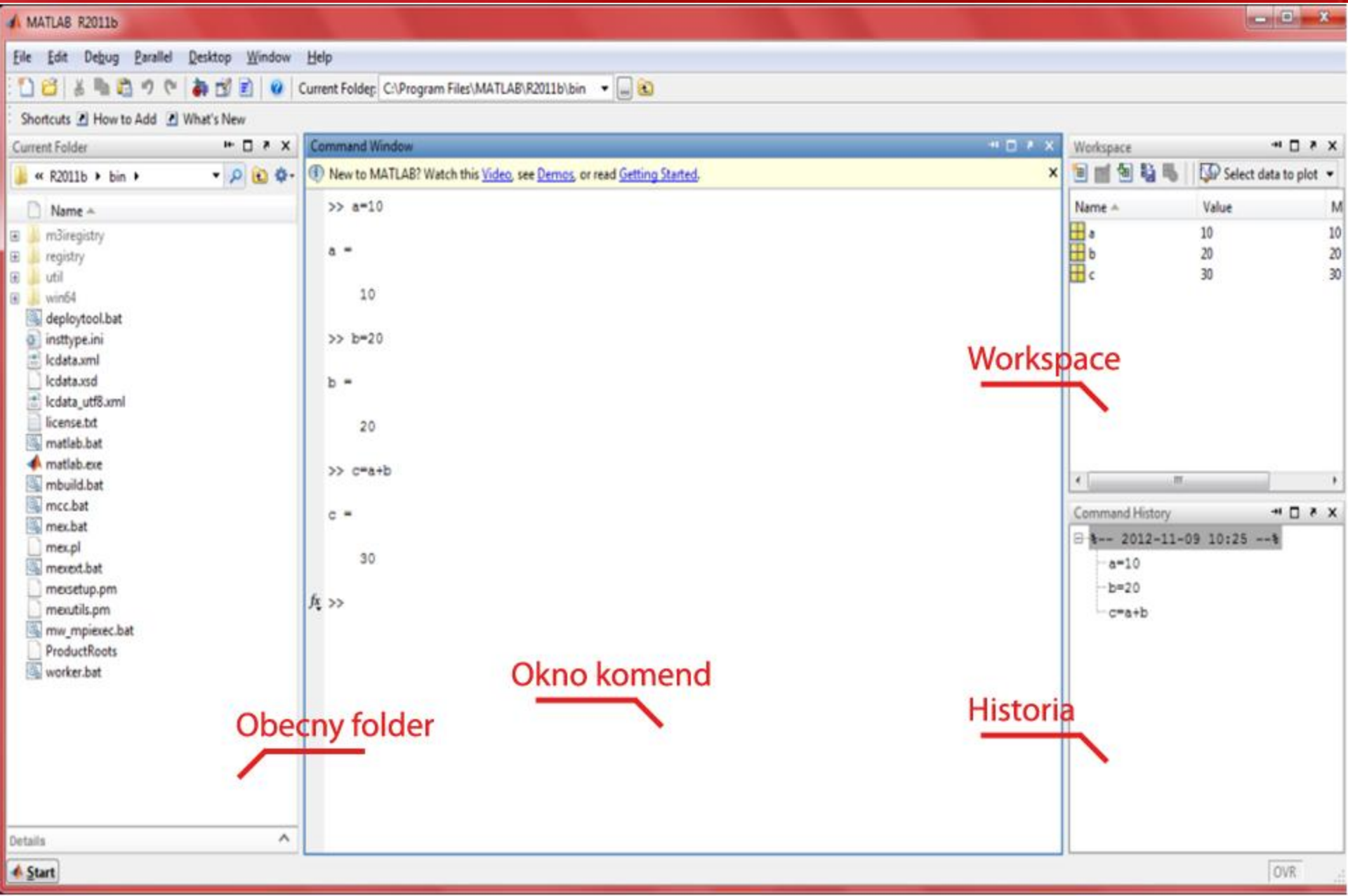
Od wersji 2012 w Matlabie stosowany jest interfejs „ribbon” – element interfejsu użytkownika, zastępujący tradycyjne menu i paski narzędzi.

Ideą wstążki było stworzenie funkcjonalnej części interfejsu użytkownika, która umieściłaby wszystkie funkcje oprogramowania w jednym miejscu, czyniąc je łatwymi do znalezienia i użytkowania.



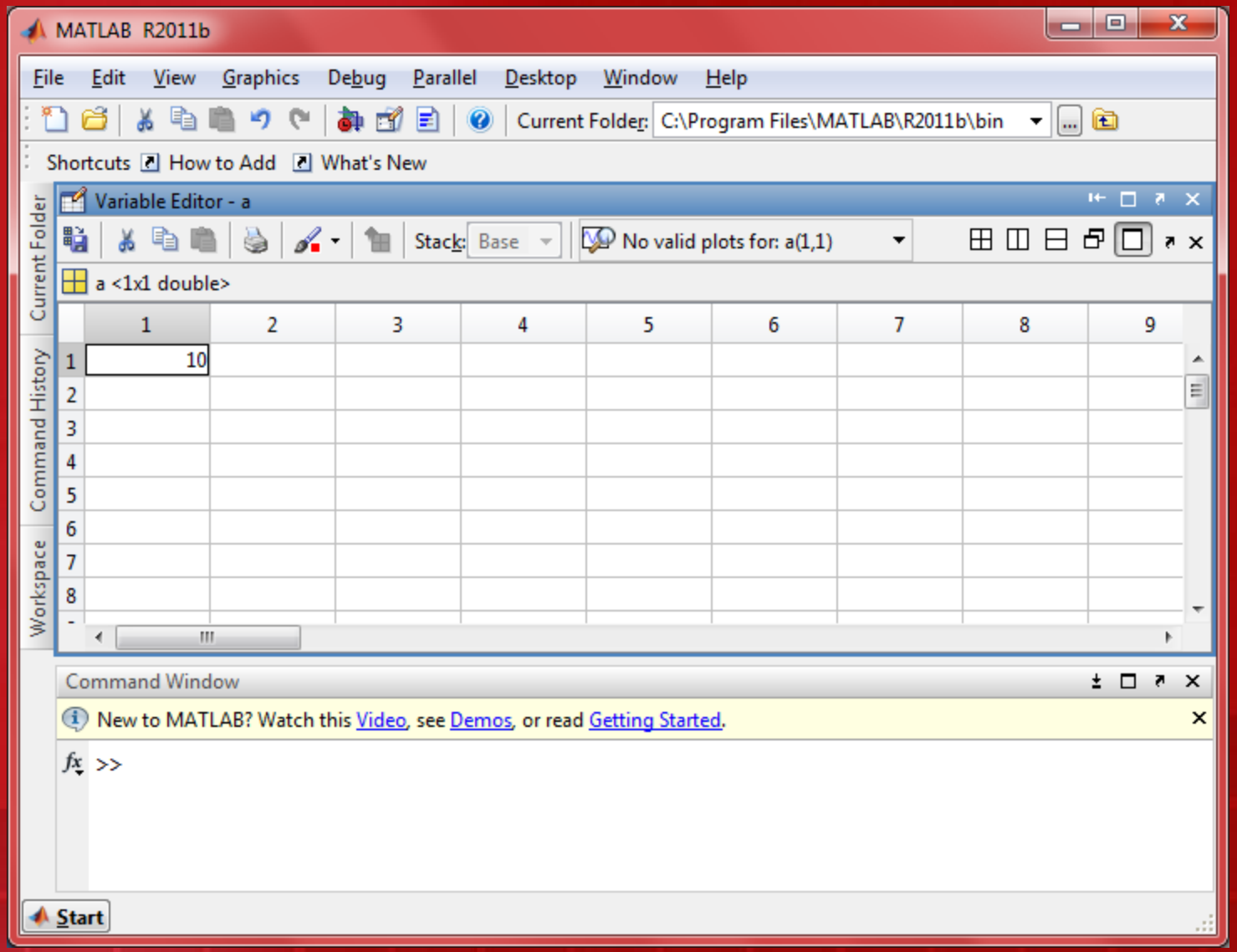
Niestety w dzisiejszych czasach większe znaczenie przykłada się do wyglądu niż do wygody użytkowania aplikacji

Pierwsze starcie, nawigacja po programie



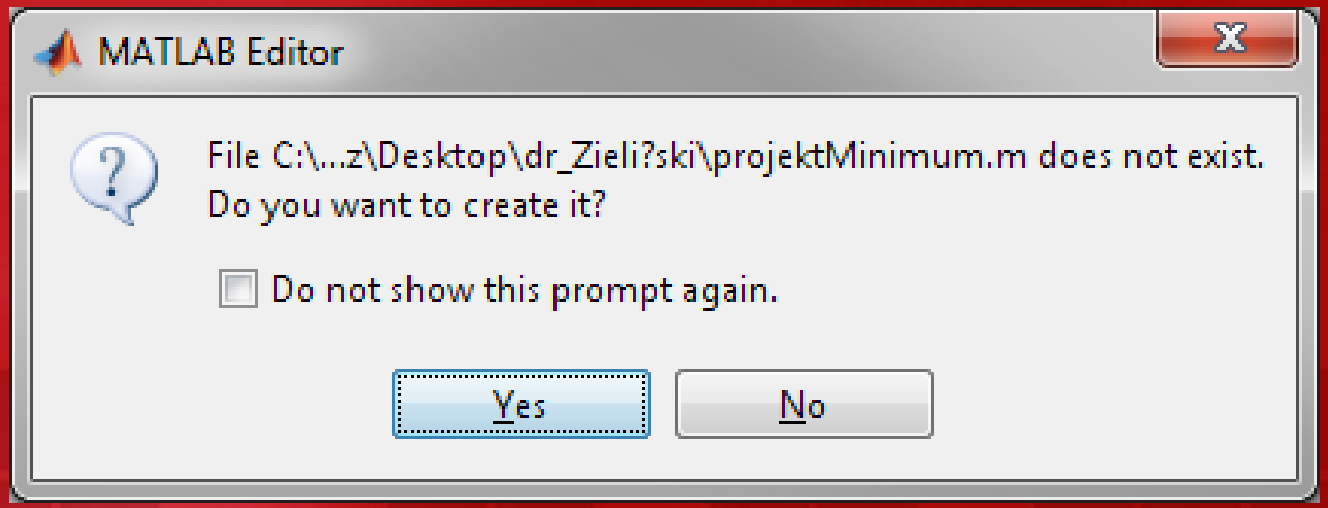
Pierwsze starcie, nawigacja po programie

Edycja w Workspace



Pierwsze starcie, nawigacja po programie

Przed przystąpieniem do tworzenia nowego projektu należy upewnić się, że folder w którym będziemy przechowywać nasze pliki nie posiada żadnych spacji lub znaków polskich.

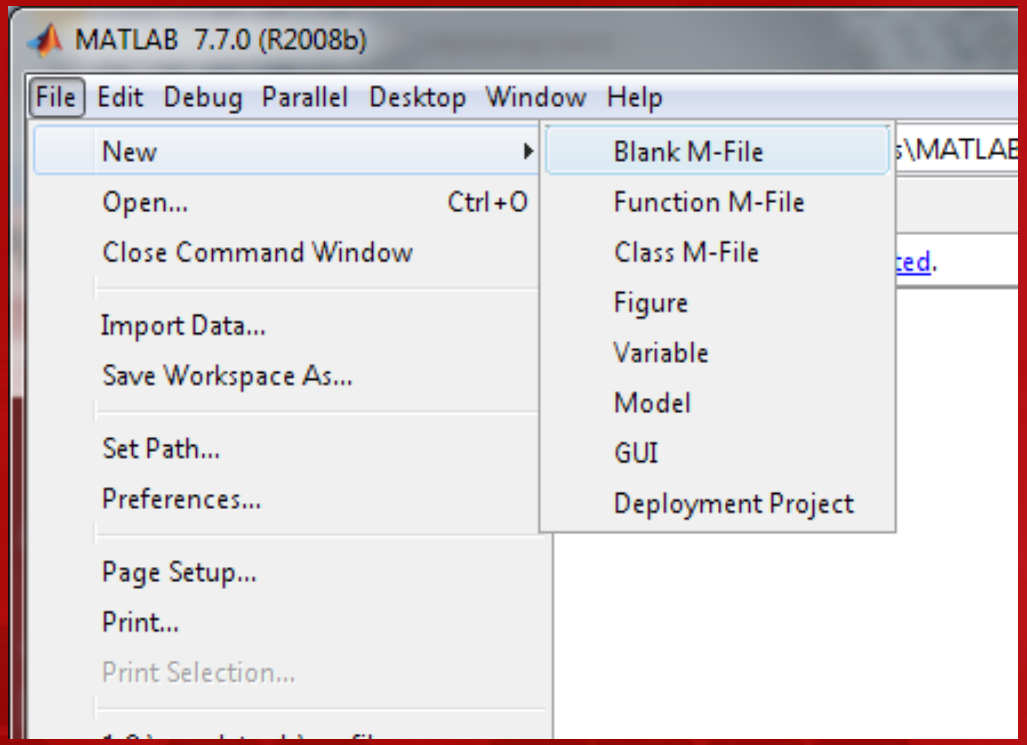


Pierwsze starcie, nawigacja po programie

Pierwszy projekt

Aby utworzyć projekt:

- **Klikamy File -> New - > Blank M-File**



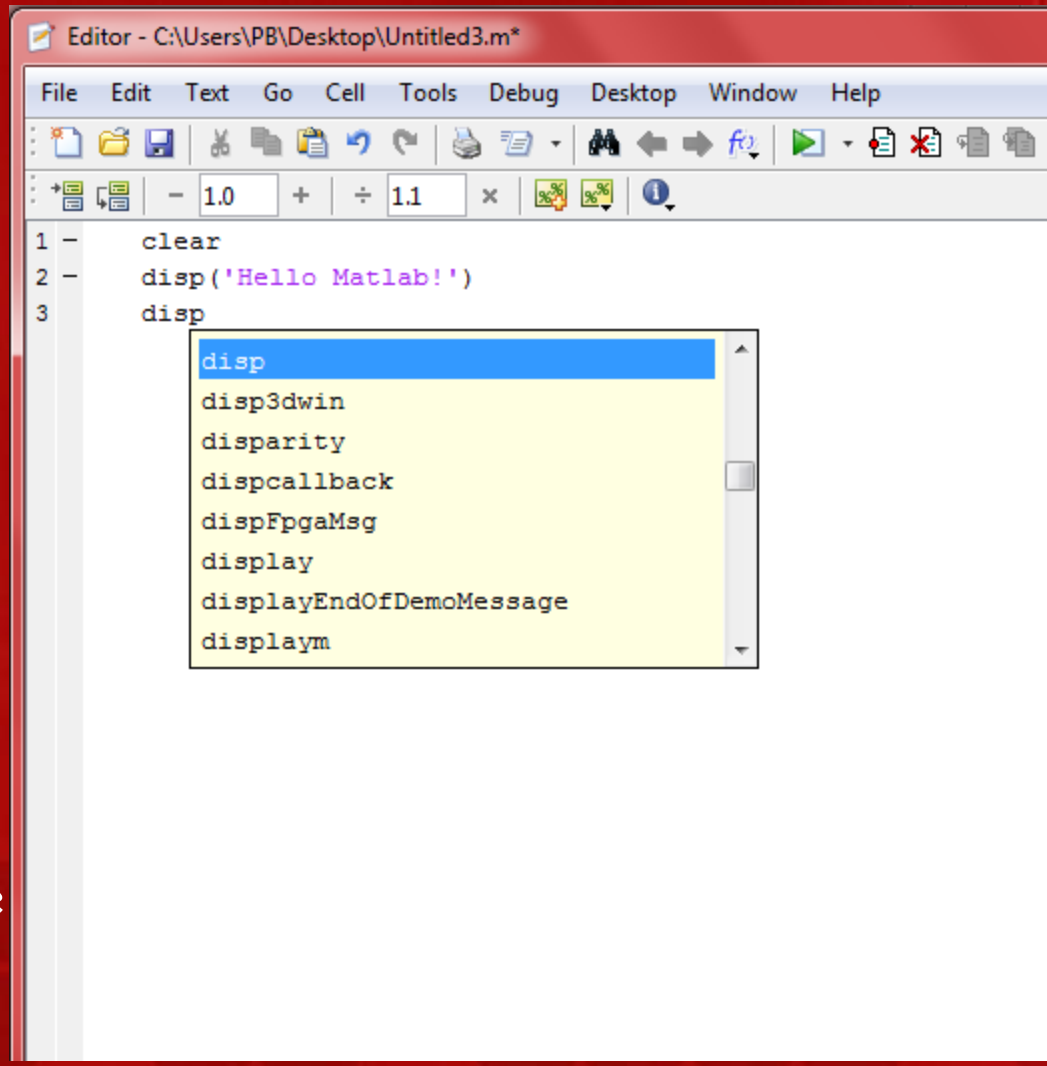
Pierwsze starcie, nawigacja po programie

Pierwszy program

Pierwszy program zawiera standardowe już słowa ‘Hello World’

Przy pisaniu poleceń można skorzystać z ‘pomocy’ w postaci rozwijalnego ‘comboboxa’.

Można go włączyć wpisując początek komendy i klikając na klawiaturze klawisz TAB.



Pierwsze starcie, nawigacja po programie

Tworzenie niezależnych bloków kodów

Każdy nowy blok kodu zaczynamy od podwójnego znaku procentu (%%).

Aby skompilować dany blok wystarczy znajdować się w bloku, który chcemy skompilować i kliknąć  **lub skrót klawiszowy CTRL + ENTER**

Editor - C:\Users\PB\Desktop\npr\bloki.m

File Edit Text Go Cell Tools Debug Desktop Window

1.0 1.1

```
1     %% pierwszy blok
2 -   clc
3 -   clear
4 -   A=ones(4,4)
5 -   B=A*4
6     %% drugi blok
7 -   clc
8 -   clear
9 -   a=10
10 -  b=exp(3)
11 -  c=( a*exp(3) )^2
12    %% trzeci blok
13 -  clc
14 -  clear
15
16 -  x = -2*pi:0.1:2*pi;
17 -  y = sin(x);
18 -  plot(x,y);
```

Tworzenie publikacji

Aby po publikacji plik wyglądał dobrze, trzeba go odpowiednio sformatować. Do tego używamy ponownie znaków pojedyńczego (%) i podwójnego procenta (%%):

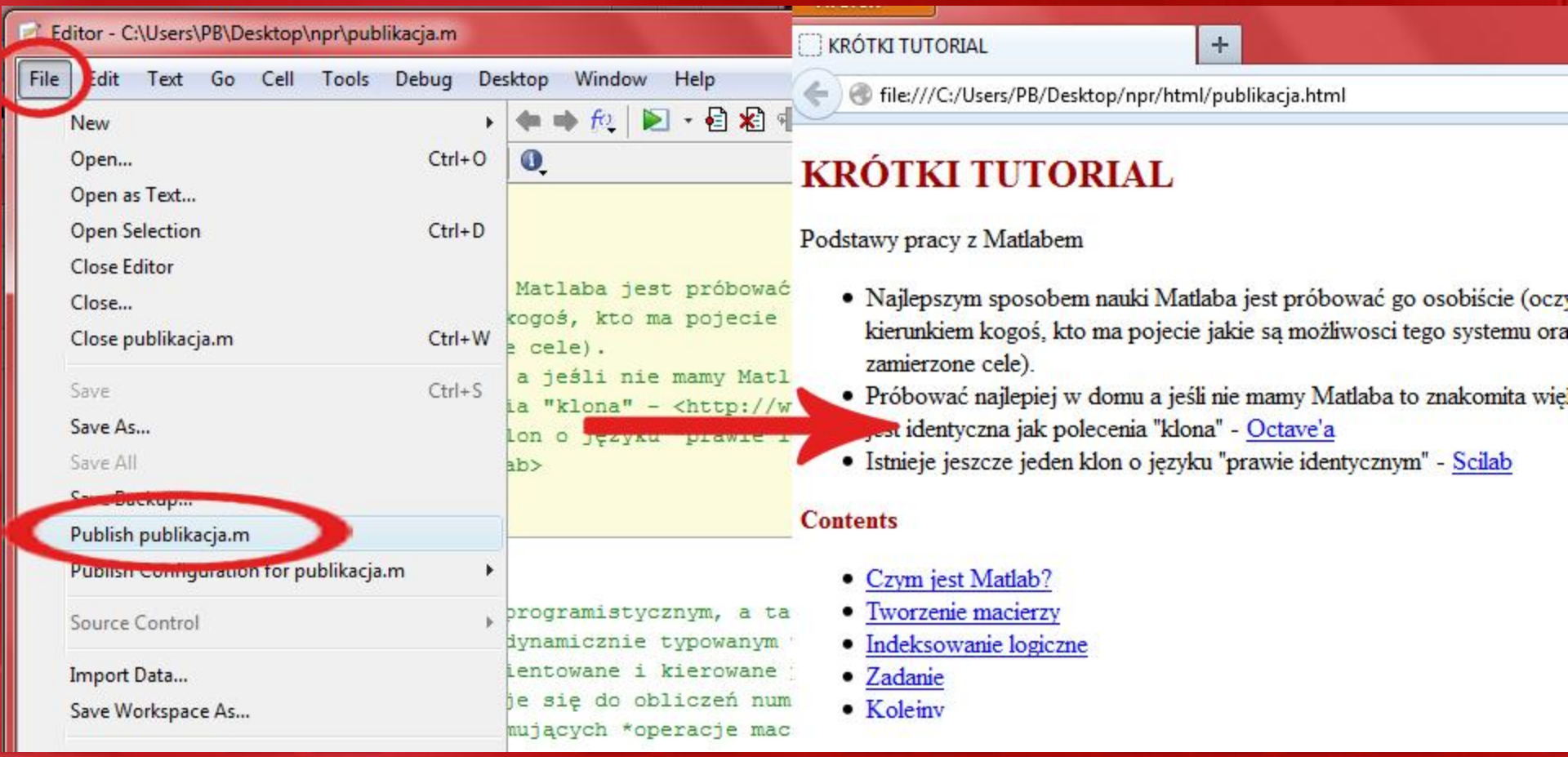
- **Aby utworzyć nagłówek używamy %%**
- **Aby utworzyć zwykły tekst używamy %**
- **Ab otrzymać tekst pogrubiony używamy gwiazdki np. * tekst ***
- **Aby utworzyć listę punktowaną zaczynamy od znaku ***

Po opublikowaniu utworzy nam się folder html , w którym znajdować się będą:

- **plik nazwa.html, w którym znajdować się będzie kod html**
- **pliki, które wykorzystaliśmy w naszej publikacji (mogą to być pliki figure(obrazki), wzory – w większości obrazków Matlab stosuje rozszerzenie *.png**

Tworzenie publikacji

Aby utworzyć publikacje klikamy FILE - > Publish nazwapliku.m



Tworzenie publikacji

KRÓTKI TUTORIAL

Podstawy pracy z Matlabem

- Najlepszym sposobem nauki Matlaba jest próbować go osobiście (oczywiscie pod kierunkiem kogoś, kto ma pojecie jakie są możliwosci tego systemu oraz jak osiągnąć zamierzone cele).
- Próbować najlepiej w domu a jeśli nie mamy Matlaba to znakomita większość jego poleceń jest identyczna jak polecenia "klona" - Octave'a
- Istnieje jeszcze jeden klon o języku "prawie identycznym" - Scilab

Contents

- Czym jest Matlab?
- Tworzenie macierzy
- Indeksowanie logiczne
- Zadanie
- Kolejny
- Kolejny 2

Czym jest Matlab?

Matlab jest środowiskiem programistycznym, a także językiem wysokiego poziomu, (interpretowanym, dynamicznie typowanym wspierającym funkcjonalnie obiektowo zorientowane i kierowane przez zdarzenia paradygmaty). Świetnie nadaje się do obliczeń numerycznych, w szczególności obliczeń obejmujących **operacje macierzowe i algebrę liniową**.

Matlab® jest produktem The Mathworks™, firmy, i koniecznie należy skorzystać z materiałów tam udostępnianych http://www.mathworks.com.

Tworzenie macierzy

Istnieje wiele sposobów tworzenia macierzy w Matlabie. Można je wprowadzic do przestrzeni

Moduły Matlaba (toolboxy)

Toolboxy – biblioteki uzupełniające

Przykładowe toolboxy:

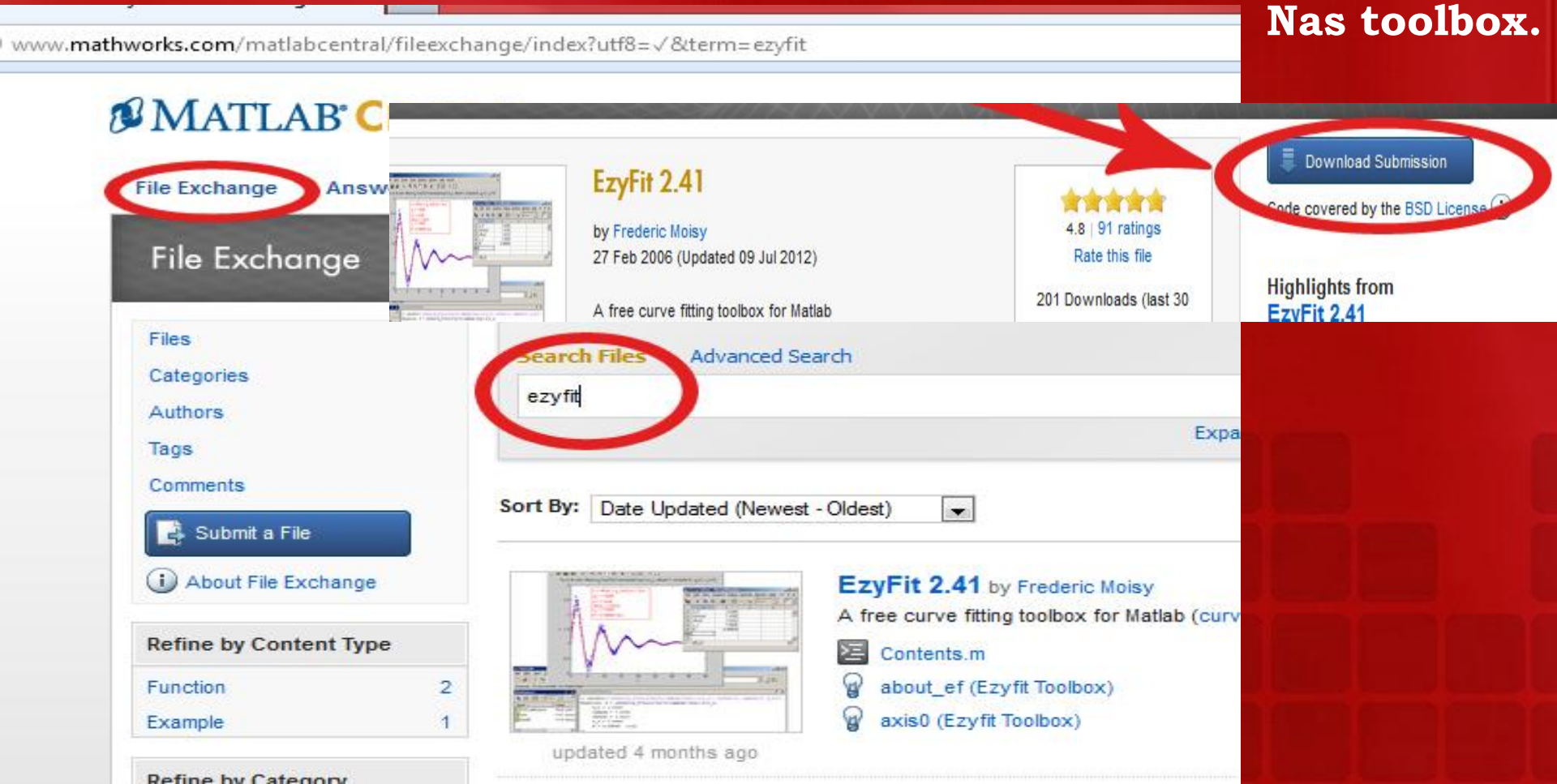
- **Signal Processing Toolbox -** przetwarzanie sygnałów; projektowanie i analiza filtrów cyfrowych; estymacja widma (analiza FFT)
- **Control System Toolbox -** systemy sterowania i regulacji; odpowiedzi czasowe i częstotliwościowe układów; przekształcenie Laplace'a i Fouriera
- **Simulink -** symulacja systemów dynamicznych
- **System Identification Toolbox** - identyfikacja; estymacja modeli
- **Optimization Toolbox -** zagadnienia optymalizacyjne z ograniczeniami
- **Neural Network Toolbox** - sieci neuronowe

Moduły Matlaba (toolboxy)

Skąd pobrać i jak wgrać toolboxy?

Wchodzimy na stronę: www.mathworks.com/matlabcentral/

Wybieramy FileExchange i w polu Search wyszukujemy interesujący Nas toolbox.

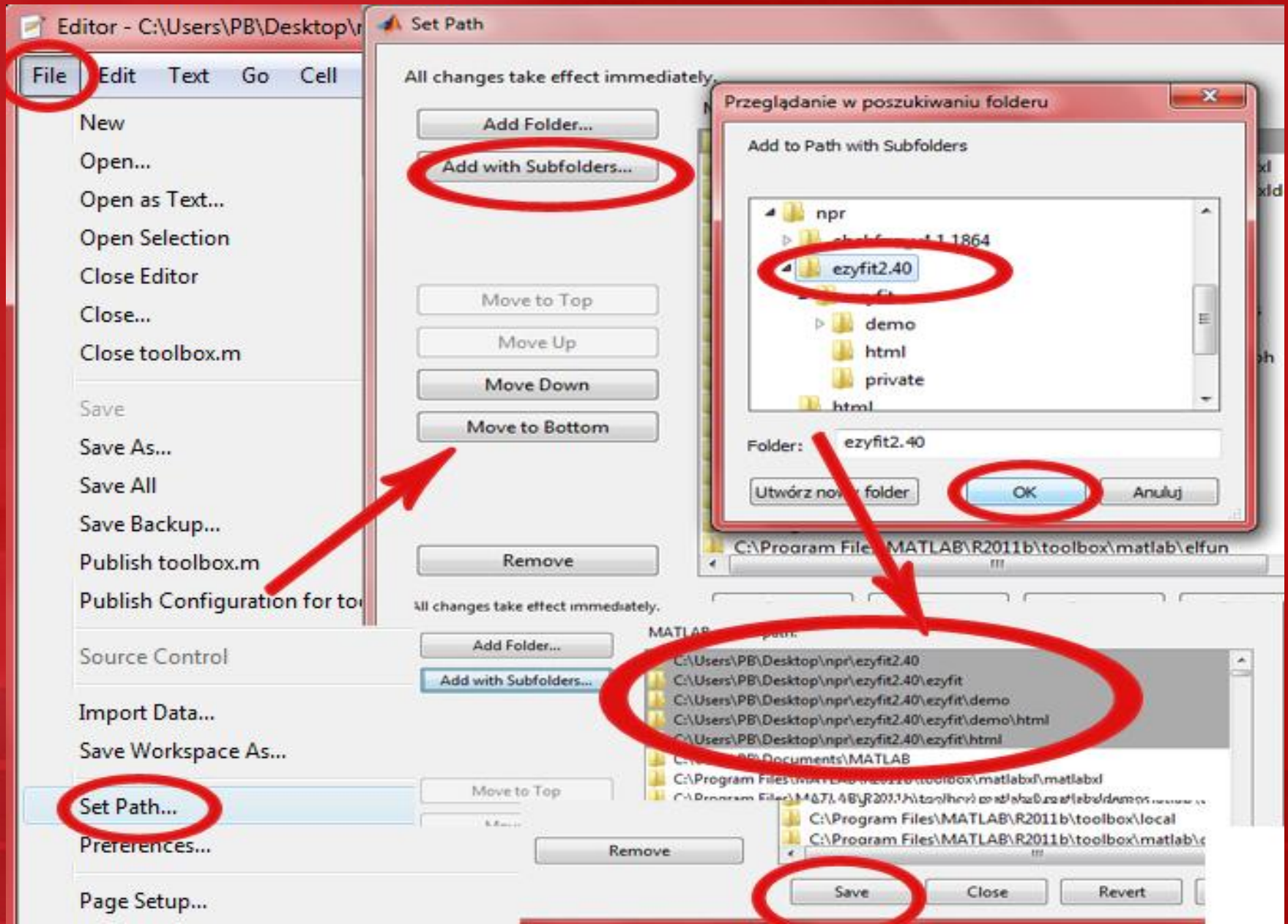


Moduły Matlaba (toolboxy)

Jeśli pobraliśmy toolbox, rozpakowujemy go i w Matlabie wybieramy :

- **FILE - > SET PATH, a następnie:**
- **ADD WITH SUBFOLDERS, wybieramy lokalizacje naszego toolboxa (u nas Ezyfit), następnie zaznaczamy wszystkie foldery z naszego toolboxa, klikamy SAVE.**

Moduły Matlaba (toolboxy)



GUI

GUI – Graphical User Interface, czyli graficzny interfejs użytkownika

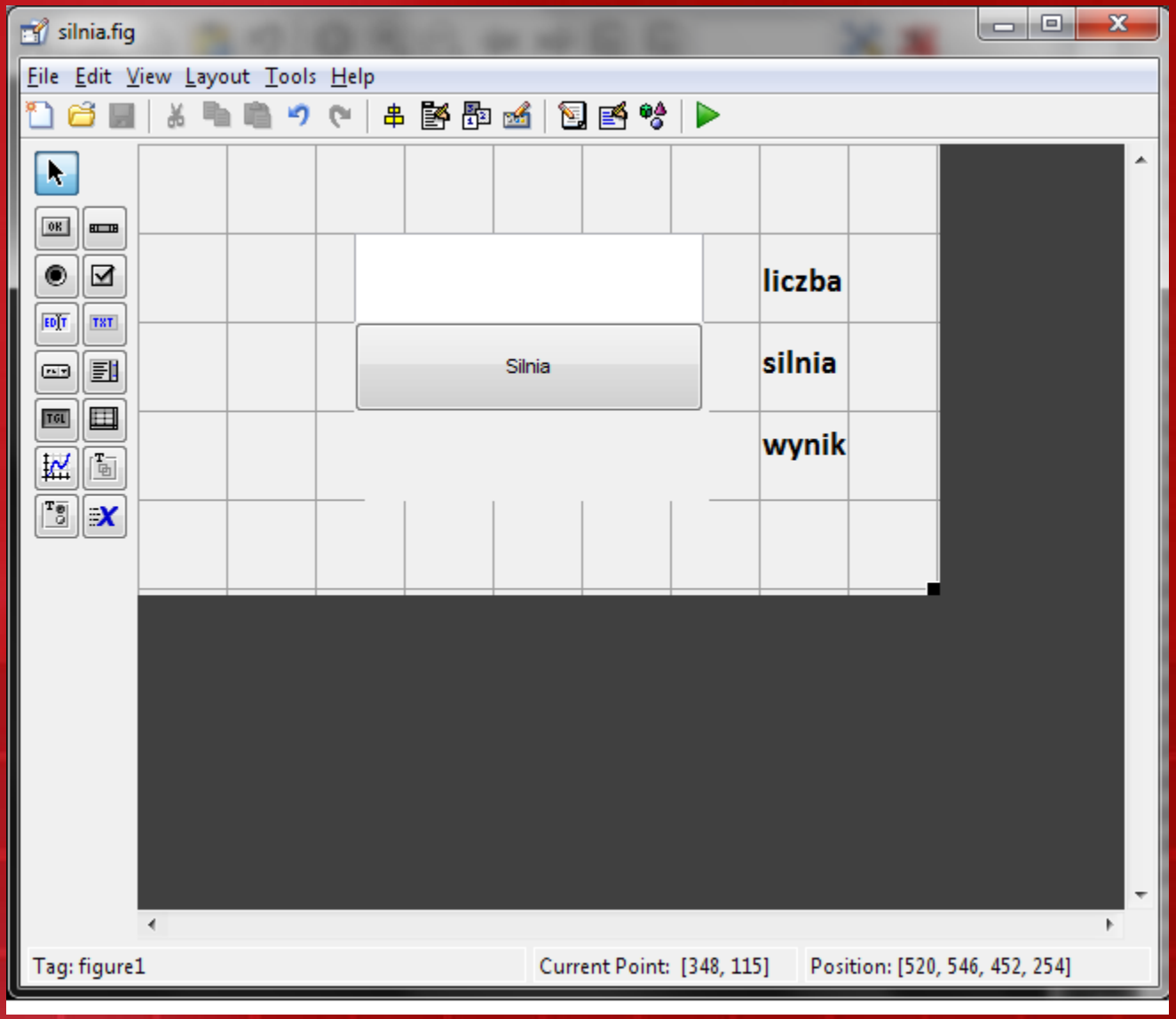
W Matlabie do projektowania GUI służy program GUIDE (Graphical User Interface Development Enviroment).

GUIDE można otworzyć na kilka sposobów:

- **FILE - > NEW - > GUI**
- **W oknie komend wpisać guide**

GUI

Tworzymy program graficzny, który będzie obliczał nam silnię

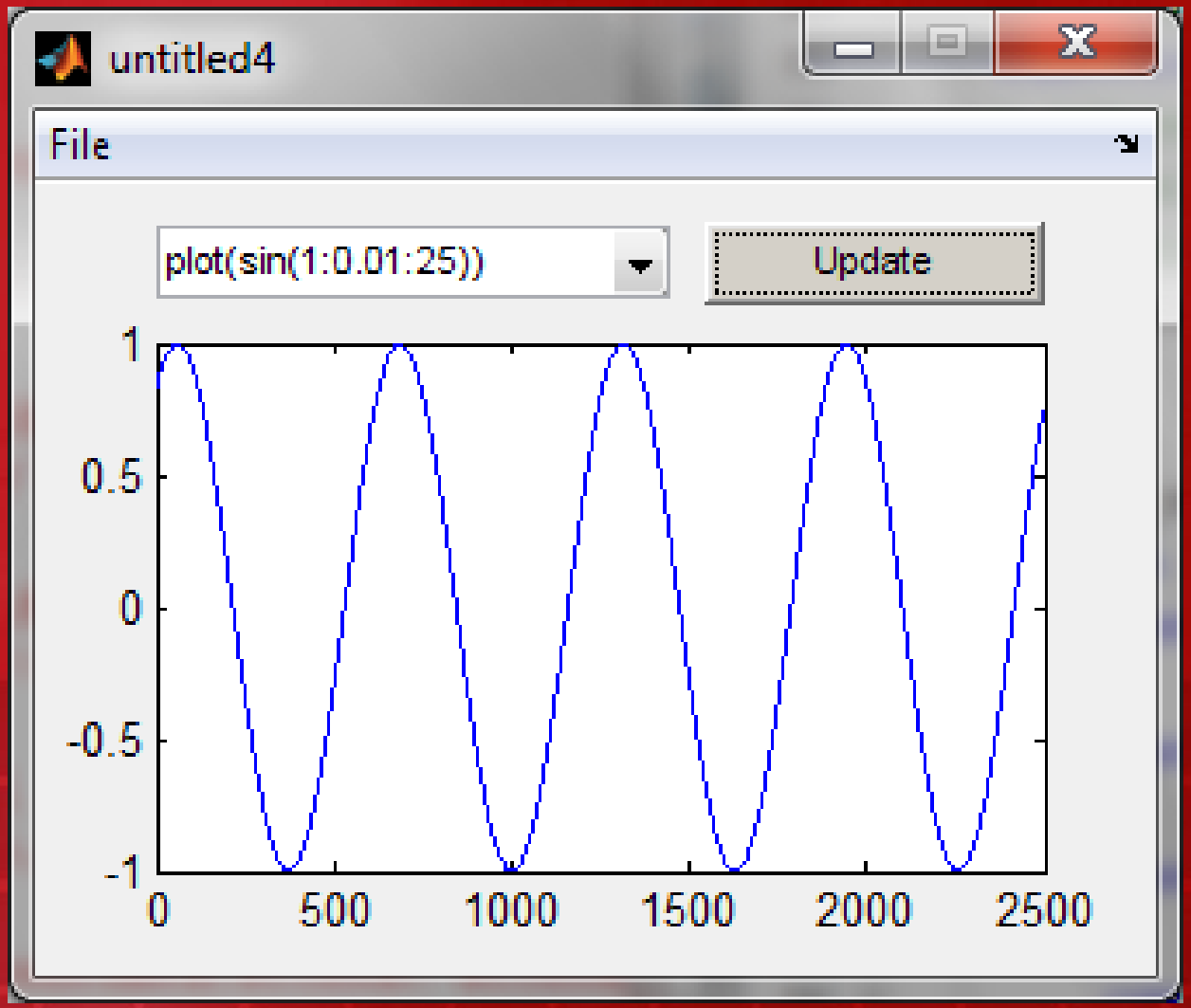


GUI

```
function silnia_Callback(hObject, eventdata, handles)
n=str2num(get(handles.liczba,'string'));
f=1;
for i=1:n
    f=f*i;
end

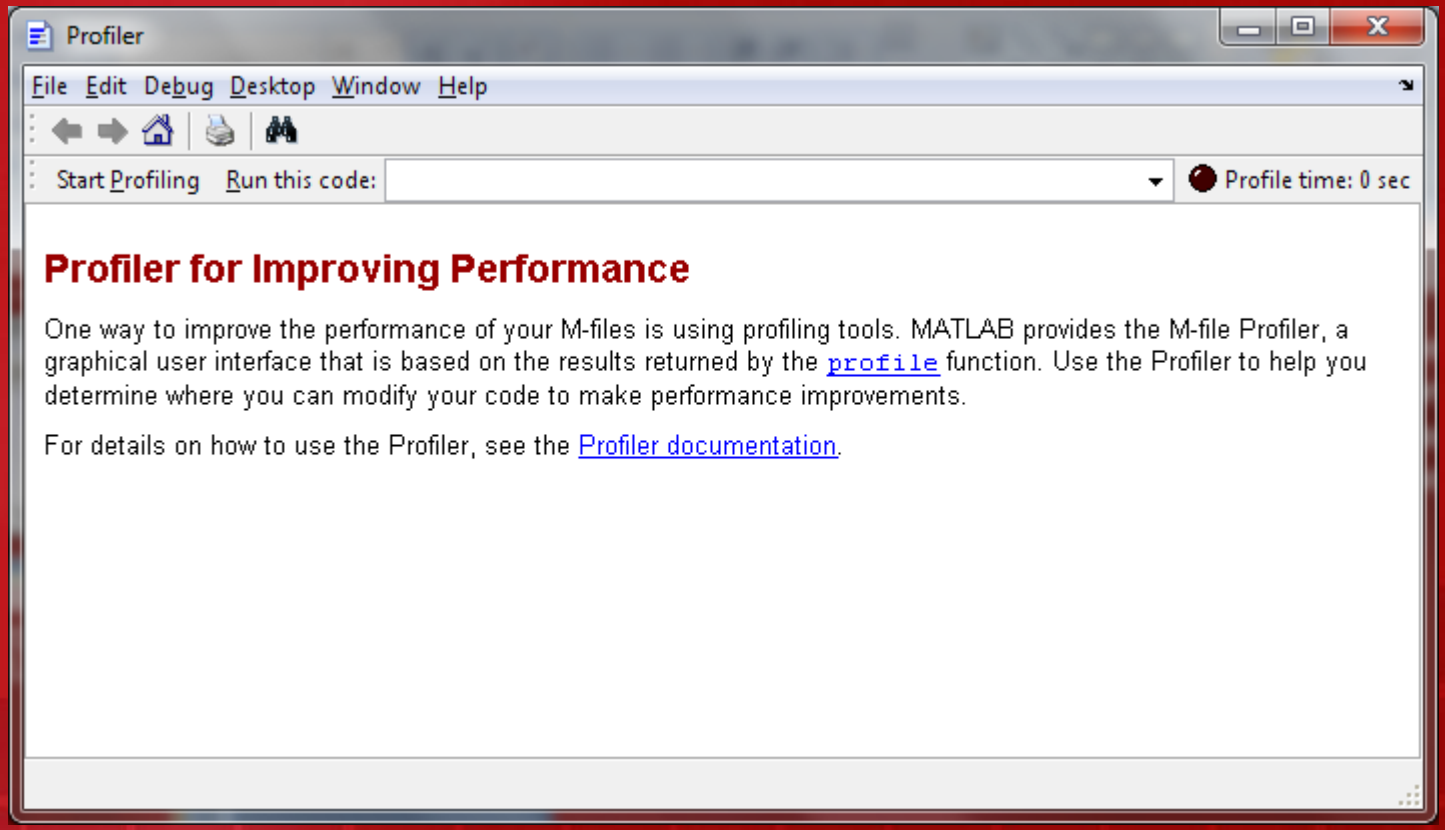
ff=num2str(f);
set(handles.wynik,'string',ff);
```

Przykładowa aplikacje GUIDE



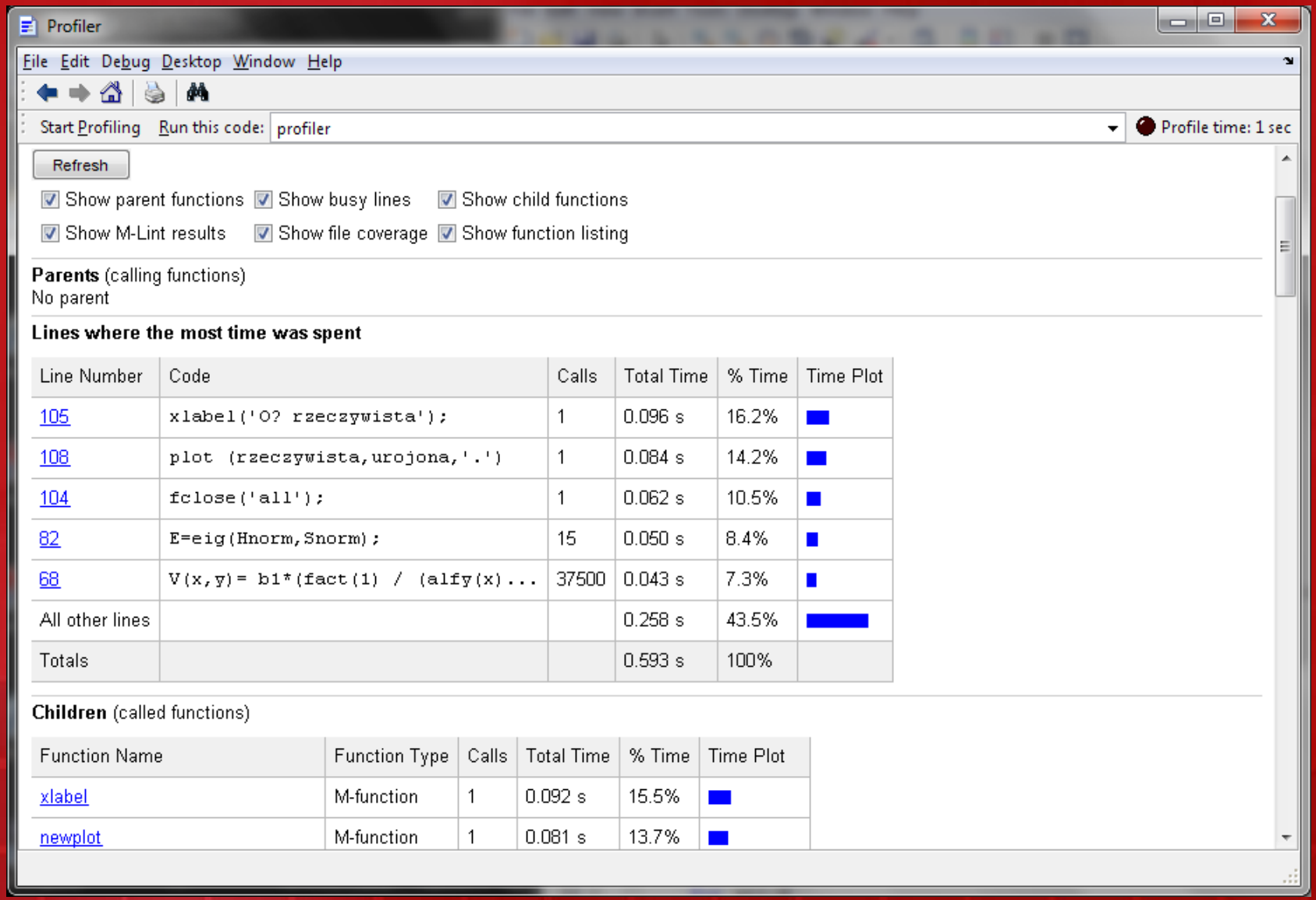
Profiler

Profiler to narzędzie do analizy programu pod kątem czasu złożoności programu, czasu trwania wywołań funkcji lub poszczególnych instrukcji. Najczęściej profiler wykorzystywany jest do optymalizacji programu.



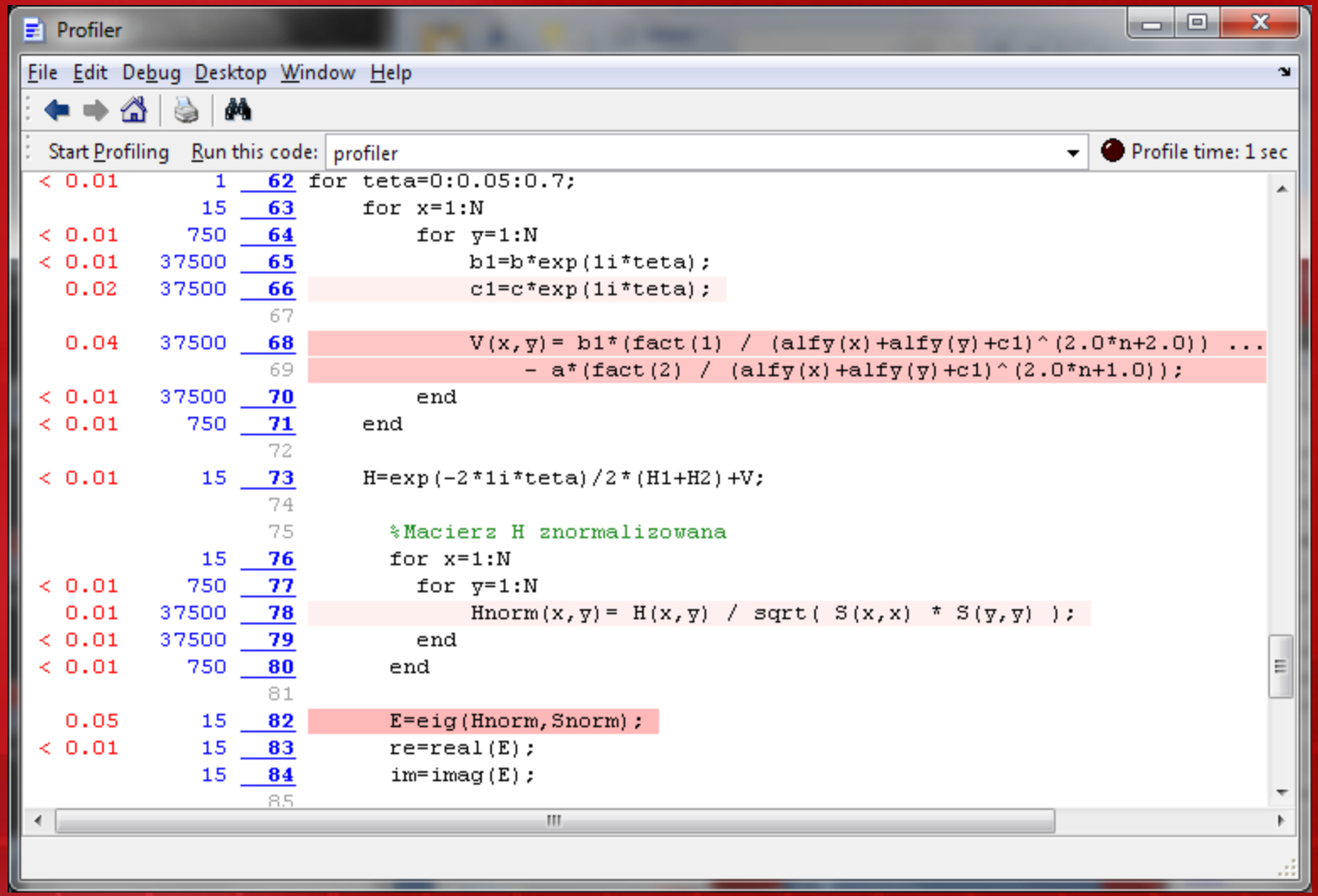
Profiler

1przykład: Czas działania poszczególnych linijek



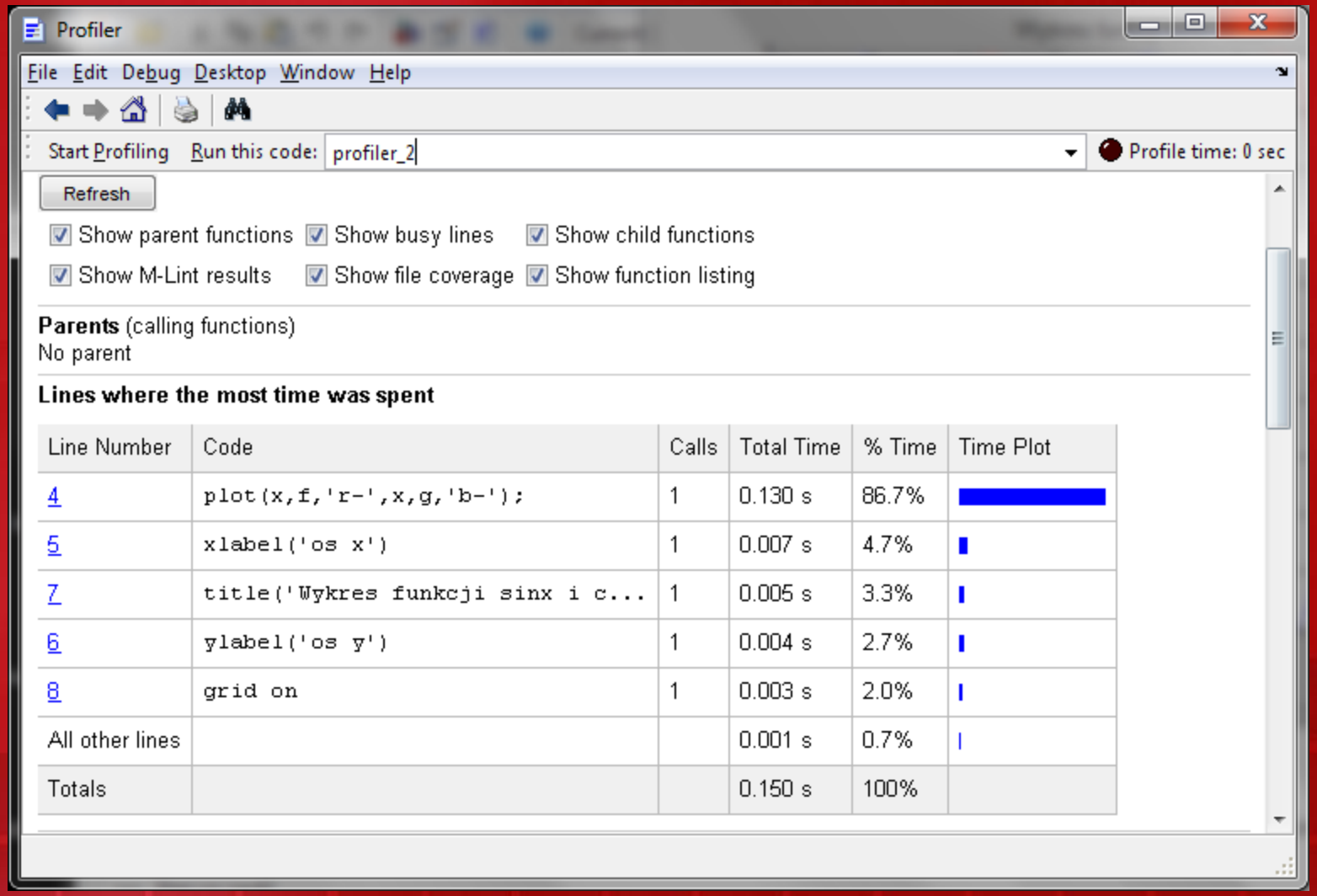
Profiler

Podkreślanie składni i instrukcji, które wykonują się najdłużej



Profiler

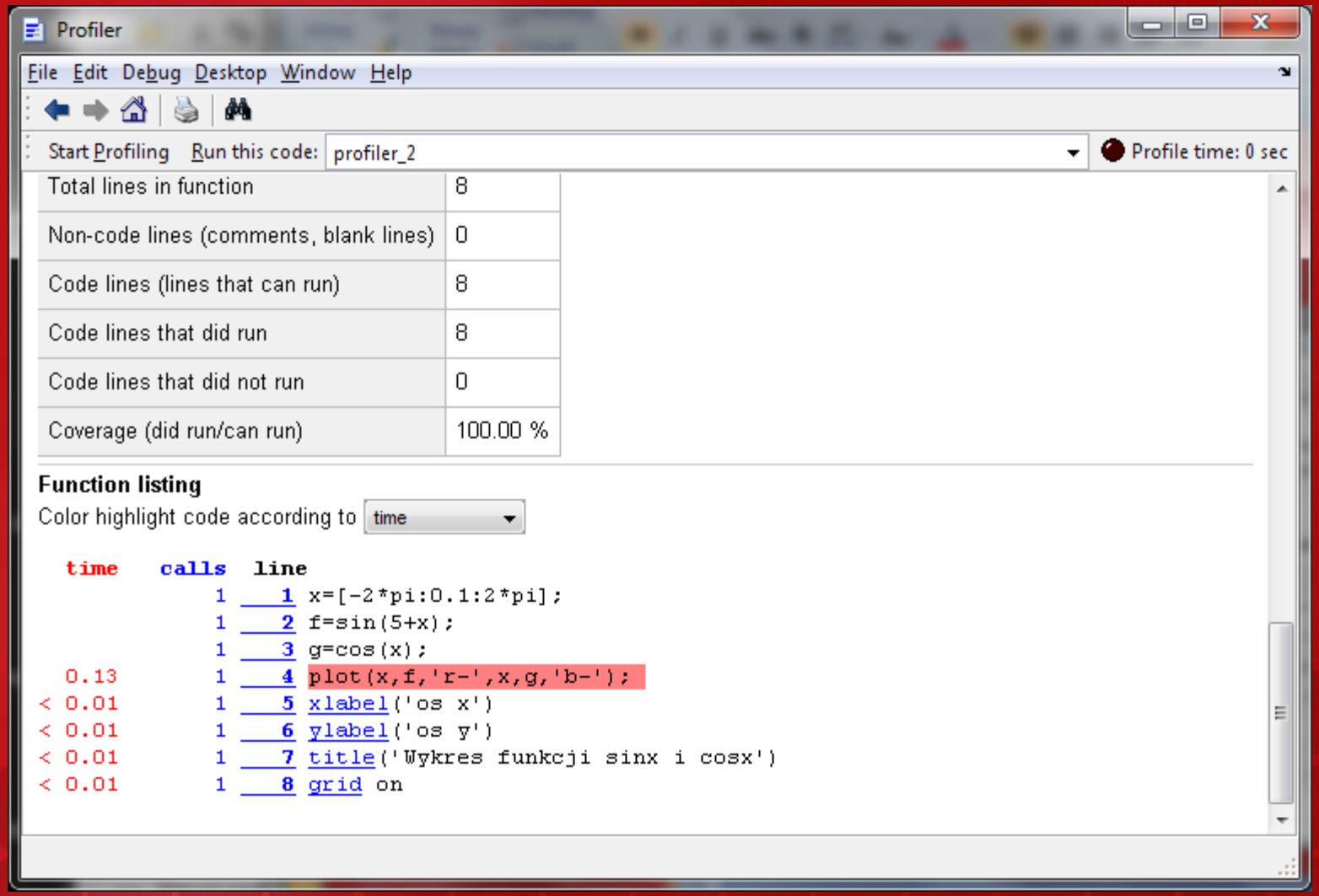
2 przykład



Line Number	Code	Calls	Total Time	% Time	Time Plot
4	`plot(x,f,'r-',x,g,'b-');`	1	0.130 s	86.7%	
5	`xlabel('os x')`	1	0.007 s	4.7%	
7	`title('Wykres funkcji sinx i c...`	1	0.005 s	3.3%	
6	`ylabel('os y')`	1	0.004 s	2.7%	
8	`grid on`	1	0.003 s	2.0%	
All other lines			0.001 s	0.7%	
Totals			0.150 s	100%	

Profiler

Podkreślanie składni i instrukcji, które wykonują się najdłużej



Debugger

Debugger jest to program komputerowy służący do dynamicznej analizy innych programów, w celu odnalezienia i identyfikacji zawartych w nich błędów.

Proces nadzorowania wykonania programu za pomocą debuggera określa się mianem debugowania.

Aby otworzyć Debugger należy z menu wybrać Debug -> Run ...

Błędy składni (syntaktyczne) są wyłapywane tuż przed uruchomieniem skryptu. Matlab podaje dokładną informację o pochodzeniu błędu. W trakcie działania programu wyłapywane są również błędy semantyczne np. niezgodność wymiarów tablic itp.

Debugger

Nawigacja po debuggerze:

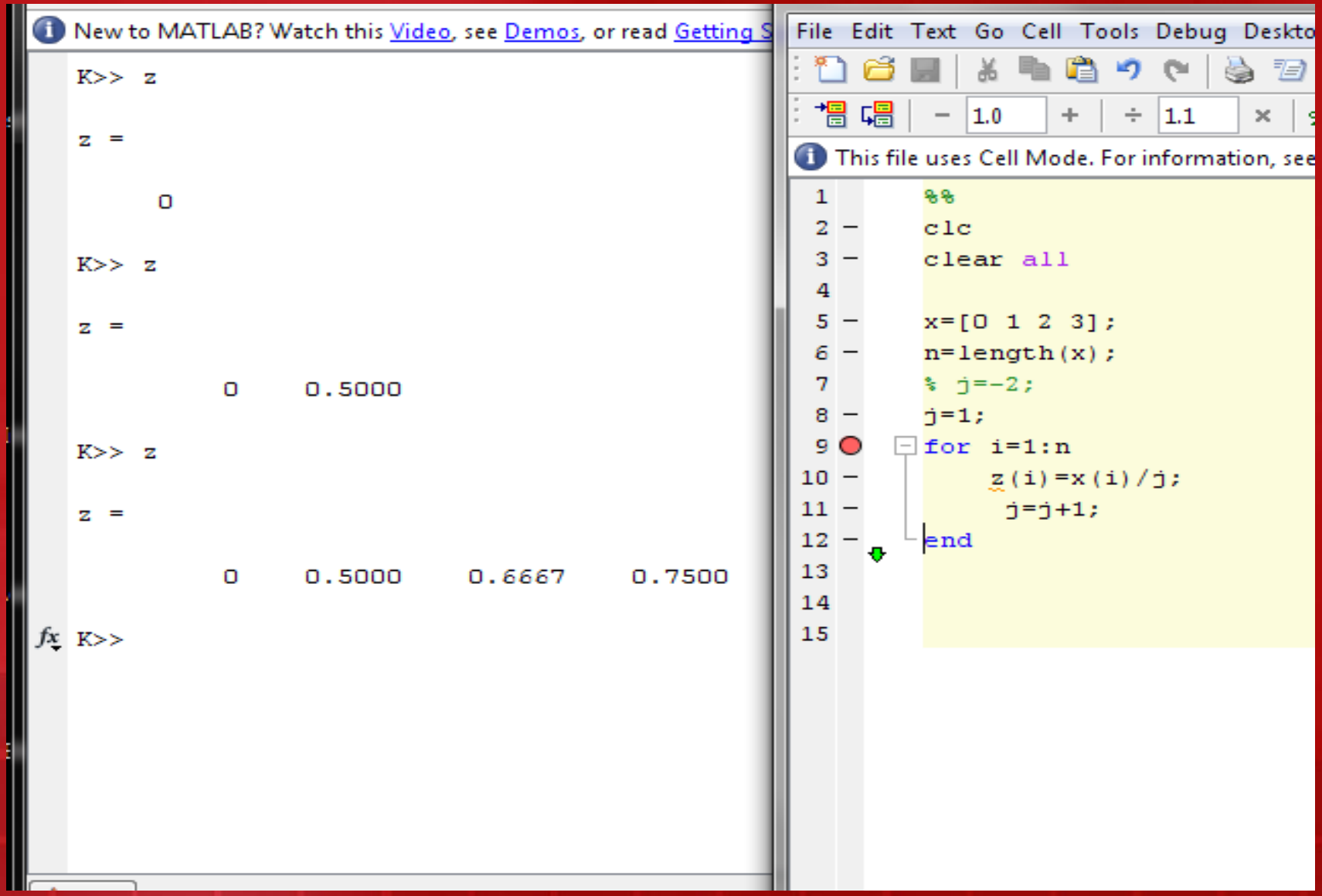
- **Breakpointy**  -
  - **stawianie breakpointów** 
  - **usuwanie breakpointów** 
  - **step** 
  - **step in** 
  - **step out** 

- **Zatrzymaj jeśli błąd**
- **Zatrzymaj jeśli wartość Inf lub NaN**

UWAGA: Przy debuggowaniu najlepiej wyłączyć Workspace bo Matlab może się wyłożyć.

Debugger

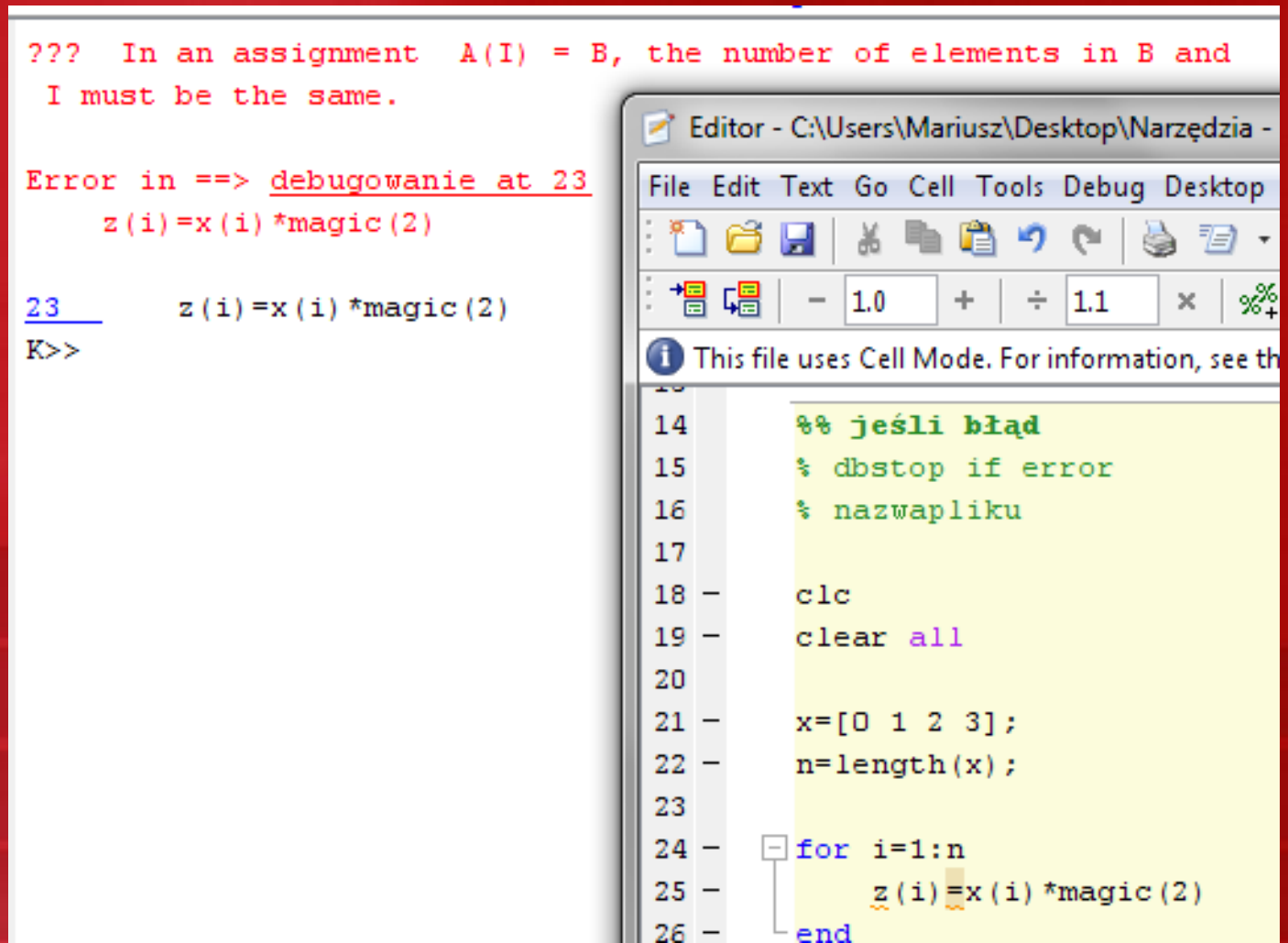
Debugowanie programu:



Debugger

Zatrzymaj jeśli błąd:

- **dbstop if error**
- **nazwapliku**

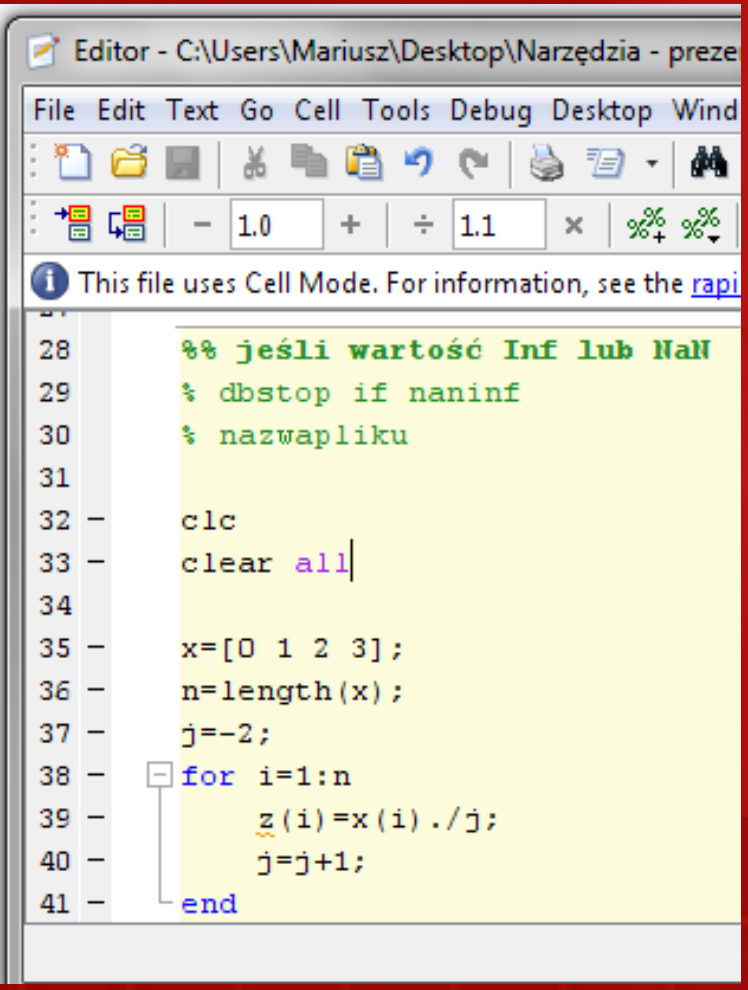


Debugger

Zatrzymaj jeśli wartość Inf lub NaN

- dbstop if naninf
- nazwapliku

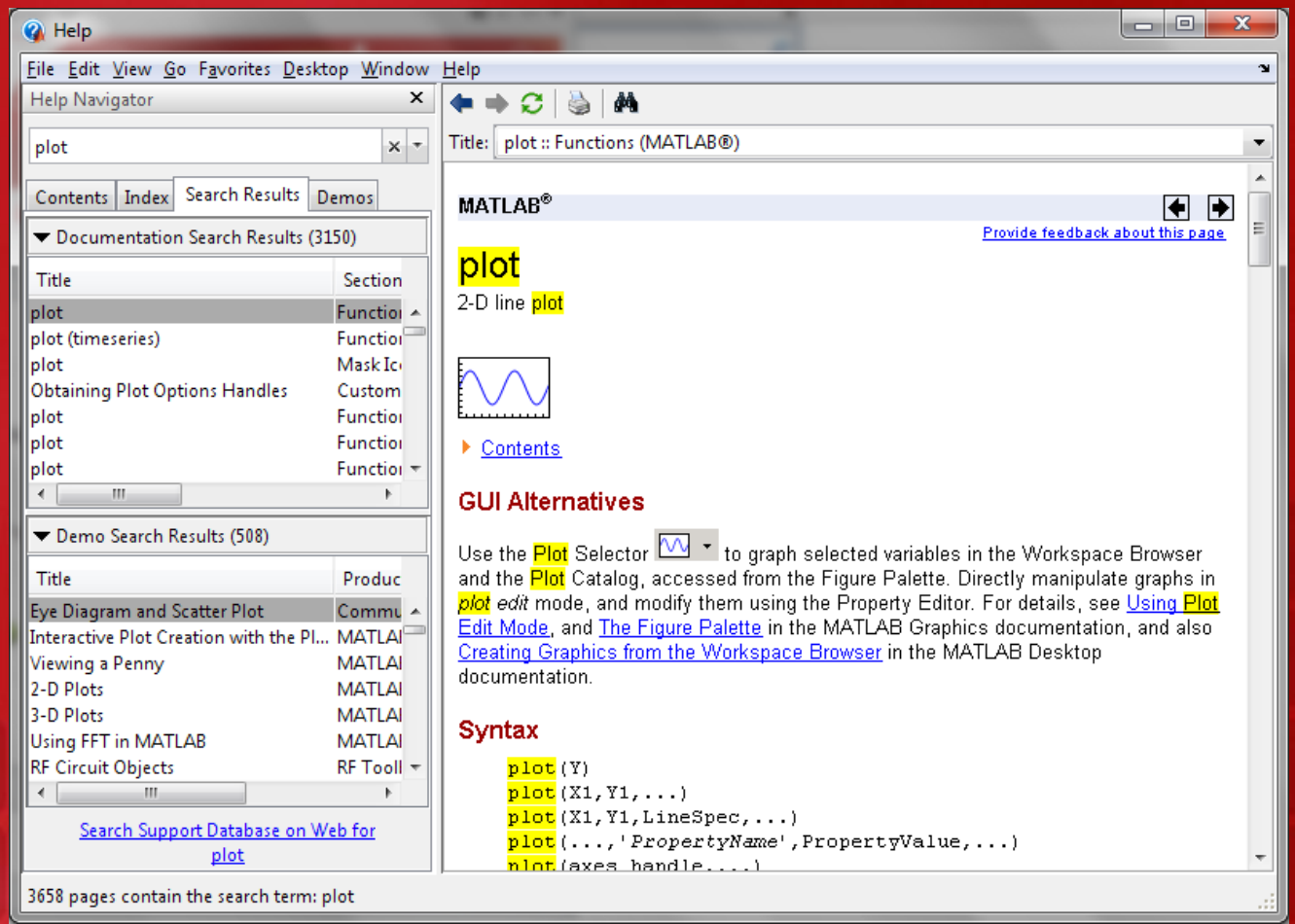
```
NaN/Inf breakpoint hit for debugowanie on line 39.
Stopping at next line.
39      z(i)=x(i)./j;
40      j=j+1;
K>>
```



Korzystanie z pomocy

Jak wejść do pomocy?

- **HELP - > Function Browser**
- **Z linii komend: ‘ doc help ‘**
- **http://www.mathworks.com/help/matlab/index.html**



Zastosowanie Matlaba, przykładowe programy

Gdzie używać Matlaba i po co ?

- **obliczenia numeryczne na macierzach**
- **przedstawianie informacji z obliczeń w postaci wykresóow łatwo edytowalnych**
- **obliczenia obwodowe i przetwarzania sygnałów**
- **aplikacje korzystające z GPS**

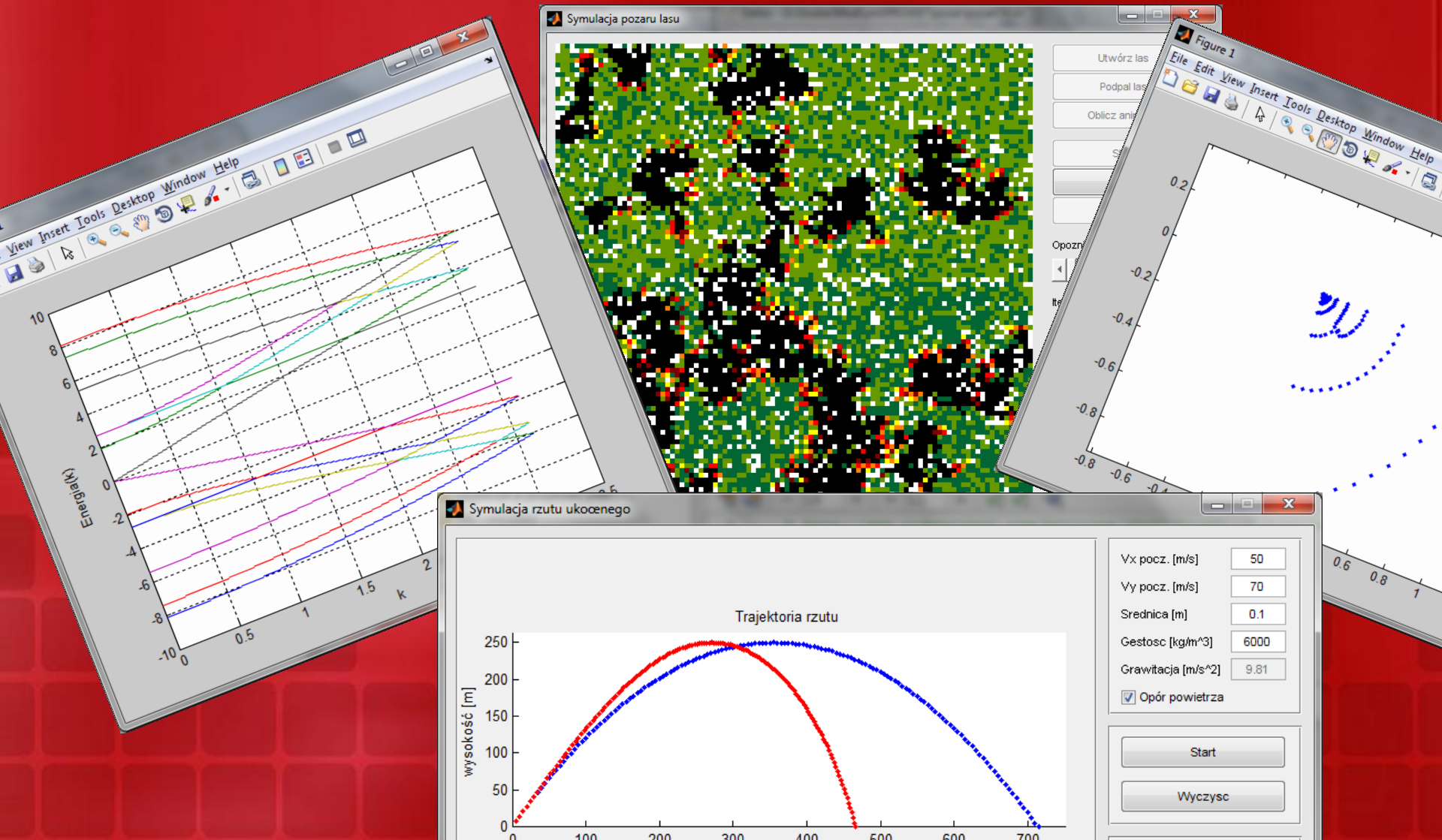
i wiele innych…

ALE ….

Zastosowanie Matlaba, przykładowe programy

ALE ....

- **Na ‘studenckim podwórku’ staje się często niezbędny**



DZIĘKUJĘ ☺