

Wprowadzenie

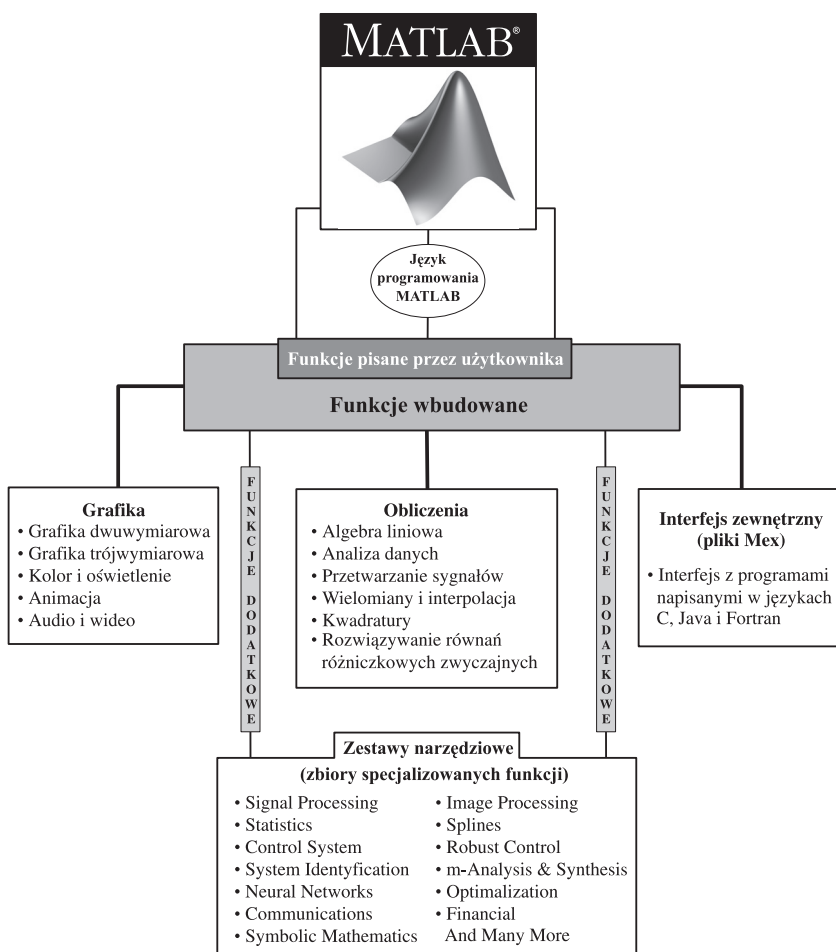
1.1. Czym jest MATLAB?

MATLAB to pakiet programistyczny, którego zadaniem jest wykonywanie złożonych obliczeń numerycznych i wizualizacja wyników. Program stanowi interaktywne środowisko oferujące setki wbudowanych technicznych funkcji obliczeniowych, graficznych oraz animacyjnych. Co jednak ważniejsze, daje użytkownikowi możliwość samodzielnego rozbudowywania zakresu zastosowań poprzez tworzenie własnych skryptów i programów w wybranym języku programowania wysokiego poziomu. Nazwa MATLAB pochodzi od MATrix LABoratory.

Diagram na rysunku 1.1 przedstawia najważniejsze funkcje i zdolności MATLAB-a. Wbudowane funkcje programu stanowią doskonałe narzędzia do obliczeń algebry liniowej, analizy danych, przetwarzania sygnałów, optymalizacji, zwyczajnych równań różniczkowych (ODE), całkowania oraz różnego typu kalkulacji naukowych. Większość tych funkcji wykorzystuje najwyższej klasy algorytmy. W programie można znaleźć wiele funkcji do obsługi grafiki dwu- i trójwymiarowej, a także animacji. Oprócz tego ci, którzy nie wyobrażają sobie pracy bez możliwości pisania kodu w Fortranie czy C, mogą korzystać z zewnętrznego interfejsu umożliwiającego uruchomienie tych programów w MATLAB-ie. Możliwości, jakie ma użytkownik, wcale nie muszą ograniczać się do wbudowanych funkcji MATLAB-a; może on tworzyć własne funkcje, pisząc je w języku programu. Funkcje napisane przez użytkowników działają w taki sam sposób jak te wbudowane. Opanowanie języka MATLAB-a oraz jego użycie nie stanowią większego problemu.

Programiści MATLAB-a udostępnili też kilka *opcjonalnych* „zestawów narzędziowych”. Są to zbiory funkcji przeznaczonych do wykonywania zadań specjalnych, takich jak obliczenia symboliczne, przetwarzanie obrazów, obliczenia statystyczne, projektowanie systemów sterujących, tworzenie sieci neuronowych i tak dalej. Lista dostępnych zestawów wciąż rośnie. Obecnie obejmuje ponad 50 pozycji. W książce tej ograniczymy się jednak do omówienia tylko jednego z nich, a mianowicie Symbolic Math Toolbox (rozdział 8).

Podstawowym elementem konstrukcyjnym w MATLAB-ie jest macierz, a *tablica* – fundamentalnym typem danych. Wektory, skalary, macierze rzeczywiste oraz zespolone są obsługiwane automatycznie, jako specjalne przypadki podstawowego typu danych. Co istotne, niemal nigdy nie trzeba deklarować wymiarów tablic. MATLAB po prostu ubóstwia macierze i wykonywane na nich obliczenia. Wbudowane funkcje zostały zoptymalizowane pod kątem operacji wektorowych. Dzięki temu polecenia i kod *wektoryzowany*¹ są w MATLAB-ie wykonywane znacznie szybciej.



Rysunek 1.1. Schematyczny diagram głównych funkcji MATLAB-a

¹ Wektoryzacja to określenie metody obliczeniowej, w której operacje są przeprowadzane jednocześnie na wszystkich argumentach z listy (wektor), a nie kolejno, jedna po drugiej. Przykładowo przyjmijmy, że θ jest listą 100 liczb. Wówczas $y = \sin \theta$ jest wyrażeniem wektoryzowanym, w odróżnieniu od ciągu wyrażeń $y_1 = \sin \theta_1, y_2 = \sin \theta_2$ itd.

1.2. Czy MATLAB wykonuje obliczenia symboliczne?

(MATLAB a Mathematica i Maple)

Pytanie takie zadaje sobie większość nowych użytkowników MATLAB-a. Pierwszą rzeczą, z jakiej powinni oni zdawać sobie sprawę, jest to, że MATLAB jest programem przeznaczonym głównie do obliczeń numerycznych, chociaż przy zastosowaniu zestawu narzędziowego Symbolic Math Toolbox (standardowy element edycji Student Edition; patrz podrozdział 8.1) może on wykonywać także symboliczne obliczenia algebraiczne². Głównie do tego typu obliczeń przystosowane są programy Mathematica i Maple. Oczywiście, wykonują one również obliczenia *numeryczne*. Prawdę mówiąc, kto poznał któryś z tych programów, ten nie będzie miał większych trudności w posługiwaniu się MATLAB-em. A zatem, po co uczyć się obsługi MATLAB-a? No cóż, jego podstawową zaletą jest łatwość używania. Ponadto przyswajanie umiejętności posługiwania się nim nie wymaga dużego wysiłku, w odróżnieniu od wspomnianych programów. Ponieważ MATLAB był projektowany przede wszystkim z myślą o obliczeniach numerycznych, a nie symbolicznych, to obliczenia te wykonuje on znacznie szybciej – często tak szybko, jak kod C czy Fortran. Istnieją inne pakiety, takie jak Xmath, o podobnym profilu, ale są one popularne raczej wśród osób zajmujących się wysoko specjalizowanymi dziedzinami. Najważniejsze jednak jest to, że jeśli chodzi o obliczenia numeryczne, szczególnie zaś o obsługę wektorów i tablic, MATLAB bije na głowę wszystkich konkurentów pod względem łatwości użycia, dostępności wbudowanych funkcji, prostoty programowania i szybkości. Dowodem na to jest fenomenalny wprost wzrost liczby użytkowników MATLAB-a w ciągu kilku ostatnich lat. Na liście zarejestrowanych nabywców figuruje ponad 2000 uniwersytetów i tysiące firm. Obecna popularność MATLAB-a zmusiła autorów takich programów jak Mathematica do zaimplementowania w nich obsługi plików w formacie MATLAB!

1.3. Czy MATLAB będzie działał na moim komputerze?

W większości przypadków odpowiedź brzmi „tak”, ponieważ MATLAB współpracuje niemal z każdą platformą. Oprócz wersji dla Windowsa, dostępne są wersje dla takich systemów operacyjnych, jak UNIX, Sun Solaris, Linux czy Mac OS X. Starsze wersje MATLAB-a mogą pracować też na platformach Mac OS i Open VMS. Aby dowiedzieć się więcej na temat dostępności programu dla konkretnego komputera, należy odwiedzić witrynę internetową, o której będzie mowa w podrozdziale 1.4.

² Obliczenia symboliczne są wykonywane z wykorzystaniem symboli lub zmiennych, a nie liczb. Na przykład, jeżeli wpisujemy $(x+y)^2$, a komputer odpowie nam, że wyrażenie to jest równoznaczne z zapisem $x^2 + 2xy + y^2$, wówczas można stwierdzić, że dokonał on symbolicznego obliczenia algebraicznego. Pakiety programowe przystosowane do wykonywania takich obliczeń są często określane mianem *komputerowych systemów algebraicznych*.

1.4. Gdzie mogę zdobyć program MATLAB?

MATLAB jest produktem przedsiębiorstwa The MathWorks Incorporated. Zapytania dotyczące produktu, a także zamówienia należy kierować pod adres:

The MathWorks Inc.
3 Apple Hill Drive, Natick, MA 01760-2098
tel.: (508) 647-7000, faks: (508) 647-7001
e-mail: info@mathworks.com
witryna internetowa: <http://www.mathworks.com>

1.5. W jaki sposób korzystać z tej książki?

Książka ta ma stanowić wprowadzenie do obsługi MATLAB-a. Jej celem jest zapoznanie użytkownika z programem w sposób możliwie najprostszy. MATLAB jest bardzo złożonym programem. Aby zrozumieć jego potęgę, trzeba czasu. Niestety, większość równie rozbudowanych aplikacji często zniechęca początkujących użytkowników. Dlatego właśnie powstała niniejsza książka – aby pomóc pokonać strach, ułatwić szybki start i wspomagać użytkownika w każdej potrzebie. Najpierw zostaną omówione najbardziej użyteczne i najłatwiej *dostępne* funkcje. Pozwoli to już na samym początku rozszerzyć zakres umiejętności czytelnika oraz zaznajomić go z programem. Niektóre z tych funkcji zostaną opisane bardziej szczegółowo niż pozostałe, co ma zachęcić czytelników do samodzielnego eksplorowania zaawansowanych możliwości MATLAB-a. W omówieniu tych funkcji i w przykładach zastosowano pewne konwencje.

- **Style typograficzne:**
 - wszystkie polecenia i instrukcje MATLAB-a będą drukowane **w ten sposób**;
 - symbole zastępujące zmienne lub nazwy w poleceniach będą wyróżnione *kursywą*; a zatem polecenie `help temat` oznacza, że czytelnik musi wpisać hasło w miejscu słowa *temat*;
 - *kursywa* będzie wykorzystywana także do *podkreślenia* i wyróżnienia nowych terminów.
 - **Przykłady.** Przykłady skopiowane z interfejsu MATLAB-a będą prezentowane w szarych ramkach, objaśnienia zaś znajdują się w białych prostokątach, co demonstruje rysunek 1.2. Zadaniem tych ramek jest pomoc niecierpliwym użytkownikom poprzez uwidocznienie efektów działania programu. Jeśli jednak czytelnik chciałby samodzielnie sprawdzić poprawność przykładu, może wykonać operacje przedstawione w ramce. Każdy przykład zaprojektowano tak, by czytelnik mógł (w mniejszym lub większym stopniu) wykonać go, nie czytając całego tekstu. Wszystkie przykłady są niezależne od systemu. Po wykonaniu przykładu należy zapoznać się z opisującym go tekstem.
-

- **Pomoc on-line.** Zaleca się korzystanie z systemu pomocy on-line programu. W przypadku najważniejszych zagadnień na marginesie strony czytelnik znajdzie odpowiednie hasło tematu.

Aby uzyskać pomoc na omawiany temat, wpisz:
help topic

Wpisanie polecenia **help temat** w MATLAB-ie skutkuje wyświetleniem listy funkcji i poleceń związanych ze wskazanym tematem. Użytkownik może uzyskać dalsze informacje o każdej z wymienionych pozycji.

The screenshot shows a MATLAB command window with the following text:

```
>>
>> 2+2
ans =
     4
>> area = pi*2.15^2
area =
    14.5220
```

On the right side of the window, there are three white rectangular boxes with gray borders containing the following text:

- Wiersz gotowości MATLAB-a
- Polecenie
- Odpowiedź MATLAB-a

Rysunek 1.2. Przykłady praktyczne wykonywane w MATLAB-ie są prezentowane na szarym tle. Teksty w białych prostokątach to objaśnienia

Nie zalecamy czytania tej książki w sposób bierny. Najlepszą metodą przyswojenia poleceń programu jest wypróbowanie ich działania. Wierzmy w to głęboko, w taki sposób działamy i zachęcamy użytkowników, by postępowali tak samo. Niecierpliwi mogą natychmiast po przeczytaniu punktów 1.6.1–1.6.3 przejść do przykładów przedstawionych w rozdziale 2.

1.6. Podstawy MATLAB-a

Omówimy tu podstawowe funkcje i polecenia programu. Na początek przyjrzyjmy się ogólnej strukturze środowiska MATLAB-a.

1.6.1. Okna MATLAB-a

W niemal wszystkich systemach program można obsługiwać za pośrednictwem trzech okien, które są przedstawione na rysunku 1.3 i opisane poniżej.

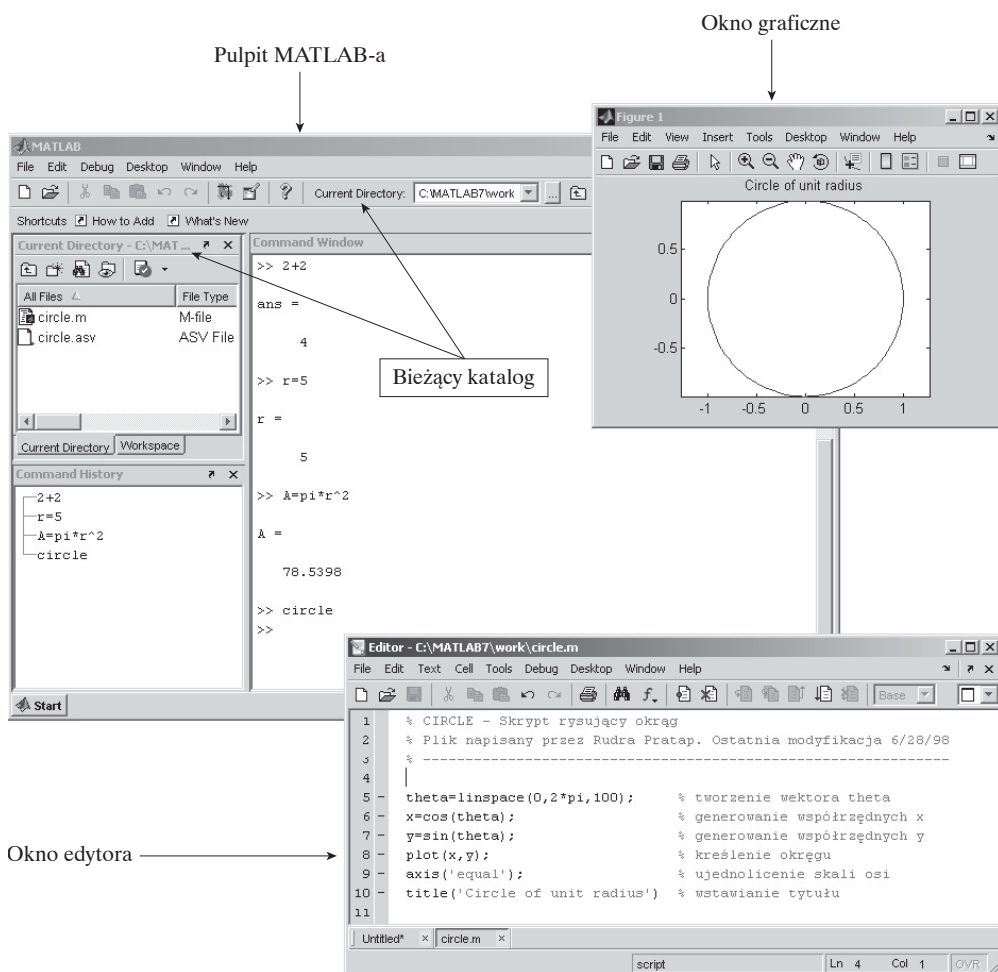
1. **Pulpit MATLAB-a.** Jest to miejsce otwierane podczas uruchamiania programu (rysunek 1.3). Pulpit MATLAB-a domyślnie tworzą następujące podokna:

- **Okno poleceń.** Jest to okno główne. Charakteryzuje się tym, że wyświetlany jest w nim znak gotowości (`>>`). Kiedy uruchamiasz program, MATLAB otwiera to okno w pierwszej kolejności. W nim właśnie wpisuje się wszystkie polecenia, w tym uruchamiające programy użytkownika. W MATLAB-ie 7 okno to, wraz z trzema mniejszymi, stanowi część głównego okna programu (rysunek 1.3). Jeśli użytkownik ma możliwość użycia okna poleceń, to powinien zignorować pozostałe. W miarę rozwoju aplikacji, w tym także i MATLAB-a, stają się one coraz bardziej rozbudowane, gdyż ich autorzy wzbogacają kolejne wersje o funkcje mające zadowolić najbardziej wymagających użytkowników. Niestety, sytuacja ta komplikuje życie początkującym – wywołuje coraz więcej frustracji, piętrzy przeszkody i zwiększa strach przed aplikacjami. Choć omówimy tutaj także dalsze podokna, wyświetlane wraz z oknem poleceń, to jednak spodziewamy się, że aż do lekcji 3 w rozdziale 2 nie będą one czytelnikowi zbyt przydatne.
- **Okno bieżącego katalogu (Current Directory).** W domyślnym układzie pulpitu MATLAB-a znajduje się ono po lewej stronie okna poleceń. W oknie tym są wyświetlane wszystkie pliki zawarte w katalogu bieżącym. Pozwala ono na nawigowanie wśród plików. Trzeba sprawdzić, czy pracujemy w katalogu, aby MATLAB miał dostęp do naszych plików i aby w tym miejscu zapisywać nowe prace. Jeśli zmienimy katalog bieżący (nawigując w systemie plików), trzeba sprawdzić, czy wybrany katalog jest widoczny w małym oknie powyżej okna poleceń, pod nazwą Current Directory. To małe okno oraz okno bieżącego katalogu są ze sobą powiązane; zmiana katalogu w jednym z nich zostaje odzwierciedlona w drugim.

W oknie tym mamy też kilka opcji, których można użyć w stosunku do zaznaczonego (kliknięciem) pliku. Aby wyświetlić te opcje, prawym przyciskiem myszy należy kliknąć wyselekcjonowaną nazwę pliku. Za ich pomocą można uruchamiać M-pliki, zmieniać ich nazwy, usuwać itd.
- **Okno szczegółów pliku (Details).** Poniżej okna Current Directory znajduje się okno Details, w którym widać szczegóły pliku wybranego w bieżącym katalogu. Są one zwykle ograniczone do informacji z pliku MAT (danych binarnych omówionych dalej), które pokazują nazwy M-plików oraz nagłówki komórek, jeśli są one w M-plikach. Zrozumienie tych szczegółów nie jest nam na razie potrzebne.
- **Okno przestrzeni roboczej (Workspace).** W tym oknie są wymieniane wszystkie utworzone zmienne, wraz z podaniem ich typu i rozmiaru. Zmienne te można poddawać różnym działaniom, między innymi kreśleniu, zaznaczając ich nazwy i wybierając opcje z menu podręcznego, które pojawia się po kliknięciu prawym przyciskiem myszy.
- **Okno historii poleceń (Command History).** Wszystkie polecenia, które użytkownik wpisuje w oknie poleceń MATLAB-a, nawet w ciągu kilku sesji (w poniedziałek, w czwartek, w następną środę i tak dalej), są rejestrowane i wyświetlane w oknie historii. Polecenia te można wybierać i uruchamiać dwukrotnym kliknięciem.

Możliwe jest również wybieranie zestawów poleceń i tworzenie M-plików. Opcja ta jest dostępna w menu podręcznym, otwieranym poprzez kliknięcie prawym przyciskiem myszy.

- Okno graficzne.** Rezultaty wykonania wszystkich poleceń graficznych, wpisywanych w oknie poleceń, są prezentowane w osobnym oknie rysunku, z szarą ramką i białym (domyślnie) tłem. Użytkownik może otwierać tyle okien rysunkowych, na ile pozwoli mu pojemność pamięci systemowej.
- Okno edytora.** Okno to umożliwia pisanie, edytowanie, tworzenie i zapisywanie własnych programów użytkownika, czyli tzw. M-plików. Edycji M-plików można też



Rysunek 1.3. Środowisko MATLAB-a obejmuje pulpit, okno graficzne oraz okno edytora. Okna rysunku i edytora pojawiają się tylko po wywołaniu ich przy użyciu stosownych poleceń

dokonać w dowolnym edytorze tekstowym. W większości systemów MATLAB oferuje własny, wbudowany edytor. Nie przeszkadza to jednak użyciu innego edytora systemowego. Wpisanie w oknie MATLAB-a polecenia poprzedzonego wykrzyknikiem powoduje tymczasowe przekazanie sterowania lokalnemu systemowi operacyjnemu, który wykonuje polecenie. Po zakończeniu edycji sterowanie jest z powrotem przekazywane MATLAB-owi. Na przykład w systemie Unix wprowadzenie polecenia **!vi myprogram.m** w oknie gotowości MATLAB-a (zakończone naciśnięciem klawisza **Return**) powoduje wywołanie edytora *vi* i otwarcie w nim pliku *myprogram.m*. Polecenie **!emacs myprogram.m** wywołuje edytor *emacs*.

1.6.2. Pomoc on-line

- **Dokumentacja on-line.** MATLAB zapewnia system pomocy on-line, objaśniający działanie wszystkich wbudowanych funkcji, a także konstrukcję języka programowania. System ten jest wywoływany poleceniami **lookfor**, **help**, **helpwin** oraz **helpdesk**. Opis systemu znajduje się w podrozdziale 3.5.
- **Demo.** Istnieje wersja demonstracyjna MATLAB-a, która pozwala zapoznać się z wieloma funkcjami programu. Wersja ta zawiera samouczek wprowadzający, z którym warto się zapoznać. Aby wywołać ów program demonstracyjny, należy wpisać **demo** w oknie poleceń i wykonywać instrukcje wyświetlane na ekranie.

1.6.3. Wejście/wyjście

MATLAB obsługuje obliczenia interaktywne (patrz rozdział 3), pobierając dane z ekranu i wyświetlając wyniki na tym samym ekranie. Oprócz tego potrafi odczytywać pliki wejściowe oraz zapisywać wyjściowe (patrz punkt 4.3.7). Dane wejściowe i wyniki są obsługiwane przez opisane niżej funkcje:

- **Typ danych.** Podstawowym typem danych w MATLAB-ie jest macierz. Tablice mogą zawierać dane różnego typu: liczby całkowite, liczby rzeczywiste, macierze, łańcuchy znaków, struktury i komórki³. W większości przypadków nie trzeba zwracać sobie głowy deklarowaniem typu danych. Na przykład użytkownik nie musi deklarować zmiennych jako rzeczywiste czy złożone. Gdy jako wartość zmiennej zostaje wpisana liczba rzeczywista, MATLAB automatycznie przypisuje jej taki właśnie typ (**double**).
- **Wymiarowanie.** Wymiarowanie w MATLAB-ie odbywa się automatycznie. Nie trzeba definiować żadnych wyrażeń wymiarujących ani dla wektorów, ani dla tablic. Wymiary istniejącej tablicy lub wektora można zbadać, wpisując polecenie **size** lub **length** (tylko w przypadku wektorów).

³ *Struktury i komórki*, jako obiekty danych, zostały wprowadzone w MATLAB-ie 5. Ich opis znajdziemy w podrozdziale 4.4. MATLAB pozwala też użytkownikom na definiowanie własnych obiektów oraz skojarzonych z nimi operacji. Jednak tego tematu nie będziemy omawiać w tej książce.

- **Wielkość liter.** MATLAB rozróżnia wielkość liter, to znaczy odróżnia litery małe od wielkich. Dlatego też **a** i **A** są w jego pojęciu osobnymi zmiennymi. Większość poleceń i nazw wbudowanych funkcji MATLAB-a wpisuje się małymi literami. Funkcję rozróżniania wielkości liter można wyłączyć poleceniem **casesen**. Jednak nie jest to zalecane.
- **Wyświetlanie wyników.** Po wykonaniu polecenia wyniki są wyświetlane na ekranie, o ile MATLAB nie jest skonfigurowany inaczej. Średnik umieszczony na końcu polecenia zawiesza wyświetlanie na ekranie wyników, z wyjątkiem graficznych, oraz wywoływanych tematów pomocy. Wyświetlaniem danych na ekranie sterują następujące funkcje:

Stronicowanie. Chcąc sprawić, by MATLAB wyświetlał dane z podziałem na ekrany, w oknie poleceń należy wpisać **more on**. Dopóki polecenie to nie zostanie wprowadzone, MATLAB wyświetla wszystkie wyniki od razu, bez względu na to, czy użytkownik zdąży je przeczytać.

Format wyjściowy. Choć obliczenia w MATLAB-ie są wykonywane z podwójną precyzją, to o wyglądzie liczb zmiennoprzecinkowych na ekranie decyduje wybrany format. W poniższej tabeli przedstawiono wartość 10π w 9 różnych formatach:

Aby uzyskać pomoc na omawiany temat, wpisz:
help format

format short	31.4159
format short e	3.1416e+001
format long	31.41592653589793
format long e	3.141592653589793e+001
format short g	31.416
format long g	31.4159265358979
format hex	403f6a7a2955385e
format rat	3550/113
format bank	31.42

Dodatkowe formaty **format compact** oraz **format loose** sterują odstępami powyżej oraz poniżej wyświetlanych wierszy, a **format +** wyświetla plus, minus lub nie wyświetla żadnego znaku odpowiednio dla liczb dodatnich, ujemnych i zera. Domyślnym formatem jest **format short**. Wyboru innego formatu dokonuje się poprzez wpisanie **format typ** w wierszu poleceń (patrz rysunek 2.1).

- **Historia poleceń.** MATLAB zapisuje wpisywane polecenia w buforze. Polecenia te można przywoływać za pomocą klawisza strzałki w górę (↑). Rozwiązanie to ułatwia edycję poprzednio wydanych poleceń. Inny sposób ich przywoływania polega na wpisaniu kilku pierwszych znaków i naciśnięciu klawisza ↑. Polecenia można także kopiować i wklejać z okna Command History. Są w nim zapisane i wyświetlane wszystkie polecenia wydane programowi w ciągu poprzednich sesji. W większości systemów Unix edytor wiersza poleceń MATLAB-a rozpoznaje standardowe skróty klawiaturowe *emacs*.

1.6.4. Typy plików

MATLAB potrafi odczytywać i zapisywać pliki kilku typów. Jednakże spośród nich pięć, które służą do przechowywania danych i programów, wykorzystuje się najczęściej.

Aby uzyskać pomoc na omawiany temat, wpisz:
help fileformats

M-pliki są standardowymi plikami tekstowymi ASCII, z rozszerzeniem nazw `.m`. Pliki te dzielą się na dwa typy: *pliki skryptów* oraz *pliki funkcji* (patrz podrozdziały 4.1 i 4.2). Większość programów pisanych w MATLAB-ie zapisywanych jest w M-plikach. Wszystkie wbudowane w MATLAB-a funkcje także przechowywane są jako M-pliki, większość z nich zaś jest zapisywana na lokalnym dysku twardym w formie prekompilowanej. Niektórym funkcjom towarzyszy ich kod źródłowy, zapisany w czytelnych M-plikach, co pozwala na ich kopiowanie i modyfikowanie.

Mat-pliki są binarnymi plikami danych, mającymi rozszerzenie `.mat`. Są one zapisywane przez program podczas zachowywania danych poleceniem **save**. Dane są zapisywane w specjalnym formacie, zrozumiałym wyłącznie dla MATLAB-a. Mat-pliki wczytuje się do programu poleceniem **load** (patrz podrozdział 3.7).

Fig-pliki to binarne pliki rysunków, charakteryzujące się rozszerzeniem `.fig`. Po ich otwarciu w MATLAB-ie są wyświetlane zapisane w nich wykresy. Pliki te zapisuje się, wybierając polecenie **File/Save** lub **Save As** (plik/zapisz lub zapisz jako) albo wpisując **saveas** w oknie poleceń. Fig-pliki zawierają wszystkie informacje niezbędne do odtworzenia rysunku czy wykresu. Wczytuje się je, wpisując polecenie **open nazwapliku.fig**.

P-pliki są skompilowanymi M-plikami z rozszerzeniem `.p`, które można wykonywać wprost w MATLAB-ie (bez konieczności wcześniejszego parsowania i kompilowania). Pliki te tworzy się poleceniem **pcode**. Gdy użytkownik opracowuje aplikację, którą pragnie się podzielić z innymi użytkownikami MATLAB-a, ale nie chce ujawniać kodu źródłowego (M-plik), może im przekazać p-kod lub P-plik.

Mex-pliki zawierają wywoływane z MATLAB-a programy napisane w Fortranie lub C. Pliki te mają rozszerzenie `.mex`. Korzystanie z nich wymaga pewnego doświadczenia w obsłudze MATLAB-a oraz cierpliwości. W książce tej nie będziemy omawiać Mex-plików.

1.6.5. Zależności od platformy

Jedną z największych zalet MATLAB-a jest niezależność od platformy. W większości przypadków praca w tym programie wygląda tak samo, bez względu na typ wykorzystywanego komputera. Różnice mogą występować jedynie w postaciach poleceń, na przykład edycji (o ile nie korzysta się z wbudowanego edytora) oraz zapisywania M-plików. Programy pisane w MATLAB-ie działają dokładnie tak samo na wszystkich komputerach. Interfejs użytkownika (oraz sposób komunikowania się z programem) także może wykazywać pewne różnice na poszczególnych platformach.

- **Uruchamianie MATLAB-a.** Jeśli MATLAB został zainstalowany poprawnie, wówczas uruchamia się go w następujący sposób:

na komputerach PC – należy odszukać folder `MATLAB`, następnie program `MATLAB` i dwukrotnie kliknąć jego ikonę; jeżeli program był już uruchamiany wcześniej i został zapisany jakikolwiek M-plik lub Mat-plik, to program można uruchomić również poprzez dwukrotne kliknięcie ikony takiego pliku;

na komputerach z systemem Unix – należy wpisać `matlab` w wierszu gotowości Uniksa i nacisnąć klawisz **Return** lub **Enter**; jeżeli ścieżka dostępu do programu jest zapisana w systemie, to `MATLAB` zostanie uruchomiony, w przeciwnym razie trzeba skontaktować się z administratorem systemu.

- **Tworzenie katalogu i zapisywanie plików.** Gdzie należy zapisywać pliki, by `MATLAB` miał do nich łatwy dostęp? `MATLAB 7` tworzy domyślny katalog o nazwie `work`, w którym, o ile użytkownik nie wskaże innej lokalizacji, zapisuje pliki. Jeśli danego komputera używa tylko jedna osoba, wtedy jest to rozwiązanie w zupełności wystarczające. Pliki zapisywane w tym katalogu (konfiguracja domyślna) są dostępne w każdej chwili. Jeżeli natomiast komputer ma kilku użytkowników, warto, by każdy z nich utworzył własny katalog roboczy.

Teoretycznie katalogi robocze można tworzyć w dowolnych miejscach systemu plików, wskazując `MATLAB`-owi ich położenie. Jednakże najwygodniejszym miejscem przechowywania wszystkich plików użytkownika jest folder umieszczony we własnym katalogu `MATLAB`-a (w przypadku komputerów PC). Zachowane w nim pliki automatycznie stają się dostępne dla programu. Jeśli jednak konieczne jest przechowywanie ich w innym miejscu, wówczas ścieżkę dostępu można zdefiniować poleceniem `path`. Ponadto polecenie `cd` pozwala zmienić domyślny katalog roboczy na wskazany przez użytkownika. Zaleca się stosowanie drugiej metody.

- **Drukowanie**

na komputerach PC – aby wydrukować zawartość aktywnego okna (poleczeń, graficznego czy edycji), należy wybrać polecenie **Print...** z menu **File**, a następnie kliknąć przycisk **Print** w wyświetlonym oknie dialogowym; druk wykresów prezentowanych w oknie graficznym można też uruchamiać poleceniem `print` wpisywanym w wierszu gotowości `MATLAB`-a;

na komputerach z systemem Unix – chcąc wydrukować zawartość pliku z poziomu `MATLAB`-a, należy wpisać stosowne polecenie uniksowe, poprzedzając je znakiem wykrzyknika (!), na przykład polecenie druku pliku `startup.m` powinno mieć postać `!lpr startup.m`; aby wydrukować wykres aktualnie wyświetlany w oknie graficznym, wystarczy wpisać `print` w wierszu gotowości `MATLAB`-a.

1.6.6. Ogólne polecenia, które należy zapamiętać

Pomoc on-line

<code>help</code>	wyświetla listę dostępnych tematów pomocy
<code>helpwin</code>	otwiera interaktywne okno pomocy
<code>helpdesk</code>	otwiera plik pomocy w oknie przeglądarki internetowej
<code>help temat</code>	wyświetla <i>temat</i> pomocy

lookfor *łańcuch* wyświetla listę tematów zawierających wskazany *łańcuch*
demo uruchamia demonstracyjną wersję programu

Informacje o przestrzeni roboczej

who lista aktualnych zmiennych w przestrzeni roboczej
whos lista aktualnych zmiennych w przestrzeni roboczej wraz z rozmiarem
what listuje M-, Mat- oraz Mex-pliki zapisane na dysku
clear czyści przestrzeń roboczą, usuwając wszystkie zmienne
clear x y z usuwa tylko zmienne *x*, *y* oraz *z*
clear all usuwa wszystkie zmienne i funkcje z przestrzeni roboczej
mlock fun blokuje funkcję *fun*, co uniemożliwia usunięcie jej poleceniem **clear**
munlock fun odblokowuje funkcję *fun*, co pozwala na jej usunięcie poleceniem **clear**
clc czyści okno poleceń, historia poleceń jest tracona
home działa tak samo jak **clc**
clf czyści okno graficzne

Informacje o katalogu

pwd wyświetla bieżący katalog roboczy
cd zmienia bieżący katalog roboczy
dir listuje zawartość bieżącego katalogu roboczego
ls listuje zawartość bieżącego katalogu roboczego, jak **dir**
path odczytuje lub definiuje ścieżkę wyszukiwania w MATLAB-ie
editpath modyfikuje ścieżkę wyszukiwania w MATLAB-ie
copyfile kopiuje plik
mkdir tworzy katalog

Informacje ogólne

computer informuje o typie komputera
clock wyświetla godzinę i datę jako wektor
date wyświetla datę w postaci łańcucha
more steruje stronicowaniem wyświetlania danych, odpowiednio do wymiarów ekranu
ver wyświetla informacje na temat licencji i wersji MATLAB-a
bench testuje wydajność działania komputera w MATLAB-ie i porównuje do innych modeli

Zamykanie

^c (Control+C) anulowanie lokalne, zatrzymuje wykonywanie bieżącego polecenia
quit wyłącza MATLAB-a
exit działanie takie samo jak **quit**

1.7. Zajrzyj tu jeszcze

Chcielibyśmy wskazać kilka rzeczy, które być może bardzo trapią początkujących użytkowników MATLAB-a. Choć wiele z tych kwestii w tym momencie może jeszcze nie mieć większego znaczenia, to jednak należy wiedzieć, że one istnieją i prędzej czy później można się z nimi zetknąć.

W przeszłości nawigowanie w MATLAB-ie narażało użytkowników wielu problemów, zwłaszcza nowym. Słyszeliśmy wiele narzekań od studentów, którzy mieli trudności z odnajdywaniem plików, wskazywaniem programowi własnych katalogów, wyszukiwaniem i uruchamianiem edytowanych plików itd. Na szczęście od wersji MATLAB 6 programiści z MathWorks wyposażyli program w wiele nowych funkcji rozwiązujących te problemy. Bieżący katalog zawsze jest wyświetlany tuż nad oknem poleceń. Można go obecnie zmieniać jednym kliknięciem myszy. Po lewej stronie okna poleceń jest także wyświetlane dedykowane podokno katalogu, w którym widnieje lista zapisanych w nim plików. Podokno daje użytkownikowi możliwość otwierania plików, ładowania (Mat-plików), uruchamiania (M-plików), edycji itd. poprzez kliknięcie prawym przyciskiem myszy. Oprócz tego możliwe jest wybieranie innych katalogów i dodawanie nowych katalogów do ścieżki MATLAB-a, co automatycznie pozwala programowi na uzyskanie do nich dostępu.

Jeżeli pliki MATLAB-a nie zostały zapisane w domyślnym katalogu `work`, wówczas należy liczyć się z możliwością wystąpienia następujących problemów:

1. **Niemożliwość odnalezienia właściwego katalogu.** Użytkownik może zapisać wiele programów (M-plików) MATLAB-a, których MATLAB nie będzie w stanie odszukać. Jeśli pliki nie znajdują się w bieżącym katalogu roboczym, MATLAB nie ma do nich dostępu. W takiej sytuacji należy sprawdzić bieżący katalog roboczy, którego nazwa jest wyświetlana na pasku u góry okna **Current Directory** albo wpisać polecenie `pwd`. Jeśli bieżący katalog nie jest katalogiem domyślnym, trzeba wskazać MATLAB-owi ten, w którym pliki zostały zachowane (patrz lekcja 6 w rozdziale 2).
2. **Zapisywanie plików w niewłaściwym katalogu.** Edycja pliku w edytorze/debugerze MATLAB-a i zapisanie go wcale nie musi oznaczać, że okno poleceń programu będzie miało dostęp do katalogu, w którym plik został zachowany. Dlatego też, jeśli próba uruchomienia pliku po zapisaniu go zakończy się niepowodzeniem, należy wskazać programowi właściwy katalog, zgodnie z opisem w punkcie 1.
3. **Edytowane pliki nie zostają nadpisane.** Uruchamiamy program, wywołując M-plik, a gdy rezultaty nie są zgodne z naszymi oczekiwaniami, dokonujemy edycji pliku, po czym ponownie go uruchamiamy – jednak MATLAB udziela takiej samej odpowiedzi! Oznacza to, że plik znów został wykonany w tej samej, poprzednio *zanalizowanej* (kompilowanej) postaci. MATLAB nie uwzględnił dokonanych zmian. Może się tak zdarzyć z wielu powodów. Najbardziej oczywiste rozwiązanie polega na wyczyszczeniu przestrzeni roboczej poleceniem `clear all` i ponownym uruchomieniu pliku.

Jest jeszcze wiele kwestii, które od czasu do czasu mogą sprawić nam trudności. Będziemy zwracać na nie uwagę w całej książce, o ile tylko wystąpi ryzyko ich pojawienia.