

AUTOMATYZACJA OBLICZEŃ INŻYNIERSKICH

Dr hab. inż. Jacek Kucharski, prof. PŁ

Dr inż. Piotr Urbanek

CEL OGÓLNY

Zdobycie umiejętności efektywnego wykorzystywania wybranych narzędzi informatycznych dla potrzeb rozwiązywania typowych problemów obliczeniowych, spotykanych w szeroko rozumianej praktyce inżynierskiej

CELE SZCZEGÓŁOWE

- Poznanie możliwości wybranych pakietów oprogramowania w zakresie wspomagania rozwiązywania problemów inżynierskich
- Łączenie różnych form programowania i umiejętnego wykorzystania funkcji „wysokiego poziomu”
- Zdobyć umiejętności „transponowania” problemu z dziedziny inżynierskiej do środowiska informatycznego
- Wskazanie typowych operacji matematycznych spotykanych w praktyce inżynierskiej

EFEKTY KSZTAŁCENIA

- Po zakończeniu przedmiotu student potrafi:
- transponować formalny opis problemu do przestrzeni pakietu obliczeniowego
 - wykorzystywać struktury danych dostępne w pakiecie do opisu rzeczywistych informacji o problemie
 - implementować operacje matematyczne z zakresu arytmetyki, algebry macierzy, rachunku zespolonego, funkcji elementarnych itp. w pakiecie obliczeniowym
 - identyfikować operacje matematyczne w kodzie programu i wykonywać ich modyfikacje
 - łączyć struktury programistyczne z wbudowanymi funkcjami i operacjami pakietu

OBSZARY PROBLEMOWE

- obliczanie wyrażeń matematycznych
- statystyczna analiza danych
- analiza szeregów czasowych
- dopasowanie krzywej, regresja, interpolacja
- zagadnienia optymalizacyjne
- całkowanie i różniczkowanie numeryczne
- rozwiązywanie równań algebraicznych
- rozwiązywanie równań różniczkowych
- ...

PAKIETY OPROGRAMOWANIA

- EXCEL
- MATLAB
- MATHCAD

MATLAB

ogólna charakterystyka

Pakiet programowy do wykonywania złożonych obliczeń numerycznych dla potrzeb naukowych i inżynierskich.

- setki wbudowanych funkcji wykorzystujących najwyższej klasy algorytmy
- bogaty zestaw tematycznych zestawów narzędziowych (Toolbox)
- możliwość pisania własnych funkcji w wewnętrznym języku programowania
- interfejs umożliwiający uruchamianie programów napisanych w Fortranie lub C
- ...

MATLAB

rys historyczny

- lata 70-te w USA powstają biblioteki do obliczeń macierzowych napisane w Fortanie
- 1980r. powstaje program (też w Fortanie) umożliwiający wykorzystywanie tych bibliotek bez potrzeby programowania w Fortanie (pierwowzór)
- od 1983r. trwają prace nad zastąpieniem Fortranu językiem C i dodaniem zintegrowanej grafiki
- 1985 r. pojawia się pierwsza wersja Matlaba (firma The Mathworks Inc.)

Tryby pracy

- **Tryb interaktywny** – potężny kalkulator z możliwością realizacji operacji graficznych
- **Programowanie w Matlab** – interpretowany język programowania wysokiego poziomu
- **SIMULINK** – symulacje komputerowe z wykorzystaniem programowania wizualnego

Wybrane typy plików

- **M-pliki** (pliki tekstowe zawierające programy napisane w Matlabie)
- **Mat-pliki** (pliki binarne zawierające dane w formacie Matlaba)
- **Fig-pliki** (binarne pliki wykresów)
- **P-pliki** (skompilowane M-pliki)
- **Mex-pliki** (programy zapisane w Fortranie lub C wywoływane z Matlaba)
- ...

Środowisko pracy

- pulpit Matlab'a
 - okno poleceń
 - okno bieżącego katalogu
 - okno przestrzeni roboczej
 - okno historii poleceń
- okno graficzne
- okno edytora

Charakterystyka środowiska

- Matlab rozróżnia wielkość liter
- Pomoc: **help temat**, **lookfor łańcuch**
- Przestrzeń robocza (who, clear, save, load)
- Po wprowadzeniu polecenia Matlab sprawdza:
 - czy jest to zmienna
 - jeżeli nie to czy jest to funkcja wbudowana
 - jeżeli nie to czy jest to funkcja w bieżącym katalogu
 - jeżeli nie to czy jest to funkcja w ścieżce dostępu
- Ścieżka dostępu
 - zawarta w zmiennej **path** generowanej przez plik pathdef.m
 - rozszerzanie: polecenia **addpath**, **savepath**

Klasy (typy danych)

TABLICA
(n-wymiarowa)

elementy składowe

h o m o g e n i c z n e

h e t e r o g e n i c z n e

logiczne
logical

liczbowe

tekstowe
char

wskaźnikowe
function handle
(@)

komórkowe
cell

strukturalne
struct

floating-point

integer

single
single

double
double

signed
int8
int16
int32
int64

unsigned
uint8
uint16
uint32
uint64

Klasy cd.

- Identyfikacja klasy (**whos, class, isa**)
- Łączenie różnych klas w jednej tablicy (*preset precedence*)

	character	integer	single	double	logical
character	character				
integer	character	integer			
single	character	integer	single		
double	character	integer	single	double	
logical	<i>invalid</i>	integer	single	double	logical



„pierwszy od lewej”

Liczby

- Rozmiar vs. zakres (**intmin**, **intmax**, **realmin**, **realmax**)
- Liczby szczególne (Inf, NaN)
- Precyzja obliczeń (**eps**)
- Zaokrąglenia (**round**, **fix**, **floor**, **ceil**)
- Liczby zespolone (**complex**, **imag**, **real**, **abs**, **angle**, **conj**)

Zmienne i wyrażenia

- Podstawowy typ zmiennej to TABLICA!
- Nazwy zmiennych (**namelengthmax**)
- Deklaracje zmiennych (prealokowanie pamięci)
- Wyrażenia
 - dwa rodzaje (ans)
 - angażują całe tablice (wektoryzacja!)

Tablice płaskie (macierze)

- Tworzenie macierzy
 - konstruktor [] (macierz pusta, skalar, wektor)
 - wyliczanie elementów
 - generowanie ciągów liczb
 - funkcje
 - konkatencja tablic
- Informacje o tablicy
 - **length, size, ndims, numel**
 - **isa, is...** (numeric, integer, float, real, logical, char, ...)
- Przekształcanie macierzy
 - do postaci wektora (:)
 - **reshape, rot90, fliplr, flipud, circshift**

Tablice płaskie (macierze) cd.

- Indeksowanie
 - $A(\text{row}, \text{col})$
 - indeksowanie liniowe
 - indeksowanie za pomocą tablicy
 - indeksujące tablice liczbowe
 - indeksujące tablice logiczne
 - indeksowanie w podstawianiu
- Usuwanie fragmentów tablicy
 - wiersze, kolumny
 - elementy, bloki elementów
- Macierze rzadkie (**sparse**)

Tablice wielowymiarowe

- Tworzenie
 - indeks $A(:,:,2)=[\dots]$
 - funkcje (**ones**(3,2,3),...)
 - konkatencja (**cat**(dim,A,B,...))
- Indeksowanie
- Operacje
 - funkcje operujące na wektorach
 - funkcje operujące *element-by-element*

Tablice komórkowe

- Tworzenie
 - konstruktor { }
 - pusta tablica (**cell**)
- Indeksowanie (rola () i {})
- Podstawianie do komórek
 - $A\{3,3\}=[\dots]$
 - $A(3,3)={[\dots]}$
- Usuwanie komórek
- Wyświetlanie tablicy komórkowej (**celldisp**, **cellplot**)

Tablice strukturalne

- Tworzenie struktury
 - str.pole=wart
 - str=**struct**(pole1,wart1,pole2,wart2,...)
- Tablica struktur
 - str(row,col).pole=wart
 - str=[**struct**(pole,wart,...);**struct**(pole,wart,...)]
- Funkcje (**fieldnames**, **setfield**, **getfield**, **rmfield**, **isfield**)
- Dynamiczne odwoływanie się do pól

Operatory

- Operatory arytmetyczne
 - macierzowe (*scalar expansion*)
 - tablicowe
- Operatory relacji
 - Relacje liczb rzeczywistych
 - Relacje liczb zespolonych
 - Relacje danych tekstowych (**strcmp**)
- Operatory logiczne
- Priorytety operatorów

Elementarne funkcje matematyczne

- Funkcje skalarne (wektoryzacja !)
- Funkcje macierzowe (**expm**, **logm**, **sqrtn**, **funm**)

Język programowania

- Instrukcja warunkowa
 - warunek logiczny
 - funkcje logiczne (**all**, **any**, **exist**, **isempty**, **isfinite**, **isnan**)
- Instrukcja *switch*
- Instrukcje pętli
 - *while*
 - *for*
- Instrukcje *continue*, *break*, *return*
- Instrukcja *try ... catch ... end*

M-pliki

- M-pliki skryptowe
- M-pliki funkcyjne
 - składnia, nazwa
 - współpraca z przestrzenią roboczą (**global, persistent**)
 - komentarz (help, lookfor)
 - liczba zmiennych wej – wyj (**nargin, nargout, varargin, varargout**)
 - subfunkcje
 - funkcje zagnieżdżone
 - funkcje prywatne
- Tworzenie funkcji w wierszu poleceń
 - inline, @, eval, feval