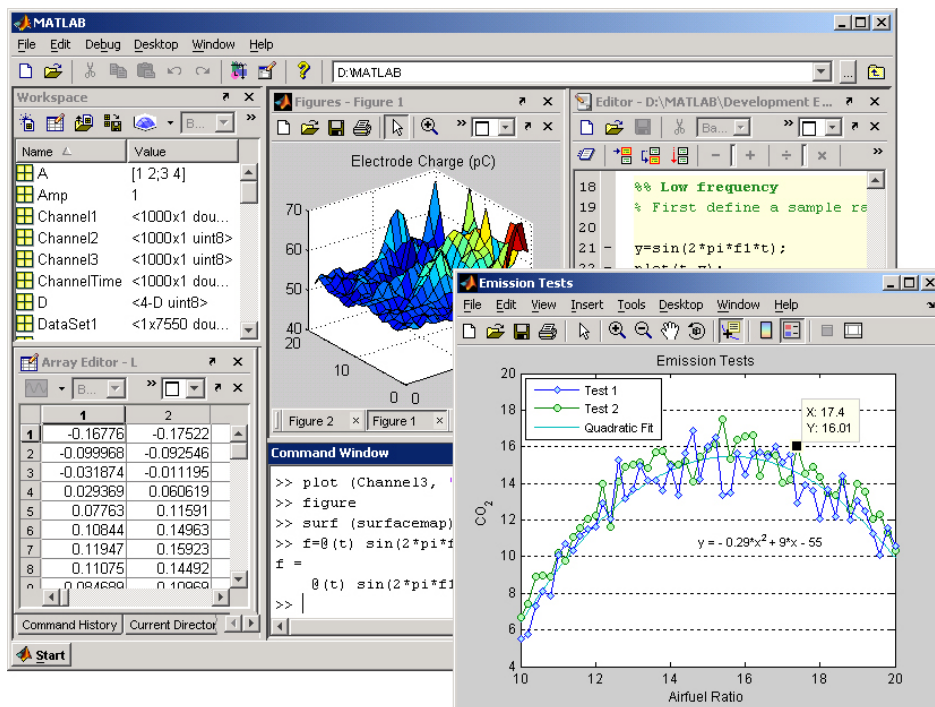


Математическое моделирование РТУ и С

Лекция 2. Основы использования MATLAB



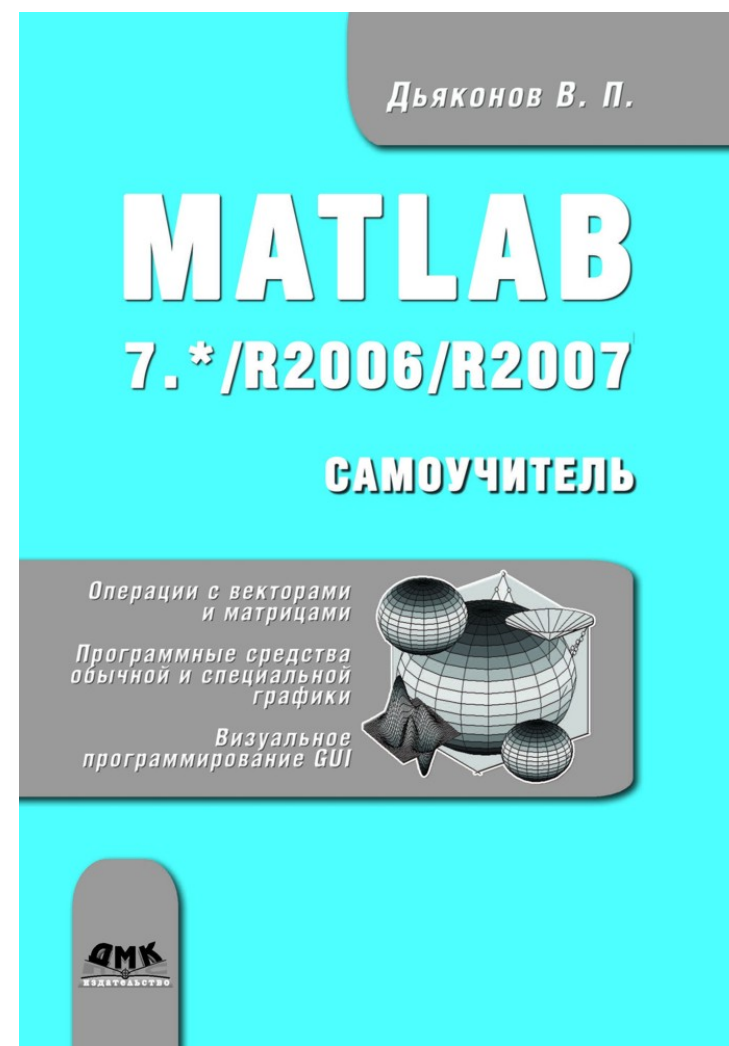
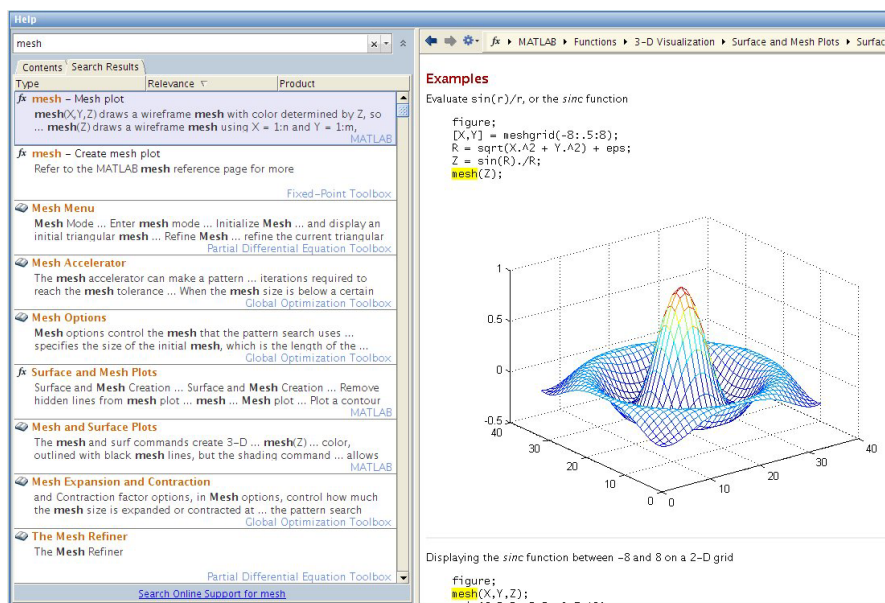
Преподаватель:
Корогодин Илья
korogodin@srns.ru

Цели

- сравнение MATLAB с другими инструментами
- обзор пользовательского интерфейса
- введение в организацию данных
- введение в нотацию языка
- обзор наиболее часто используемых функций

Информация по теме

Дьяконов В. П.
MATLAB 7.*/R2006/R2007:
Самоучитель. – М.: ДМК
Пресс, 2008. – 768 с.: ил.



Информация по теме

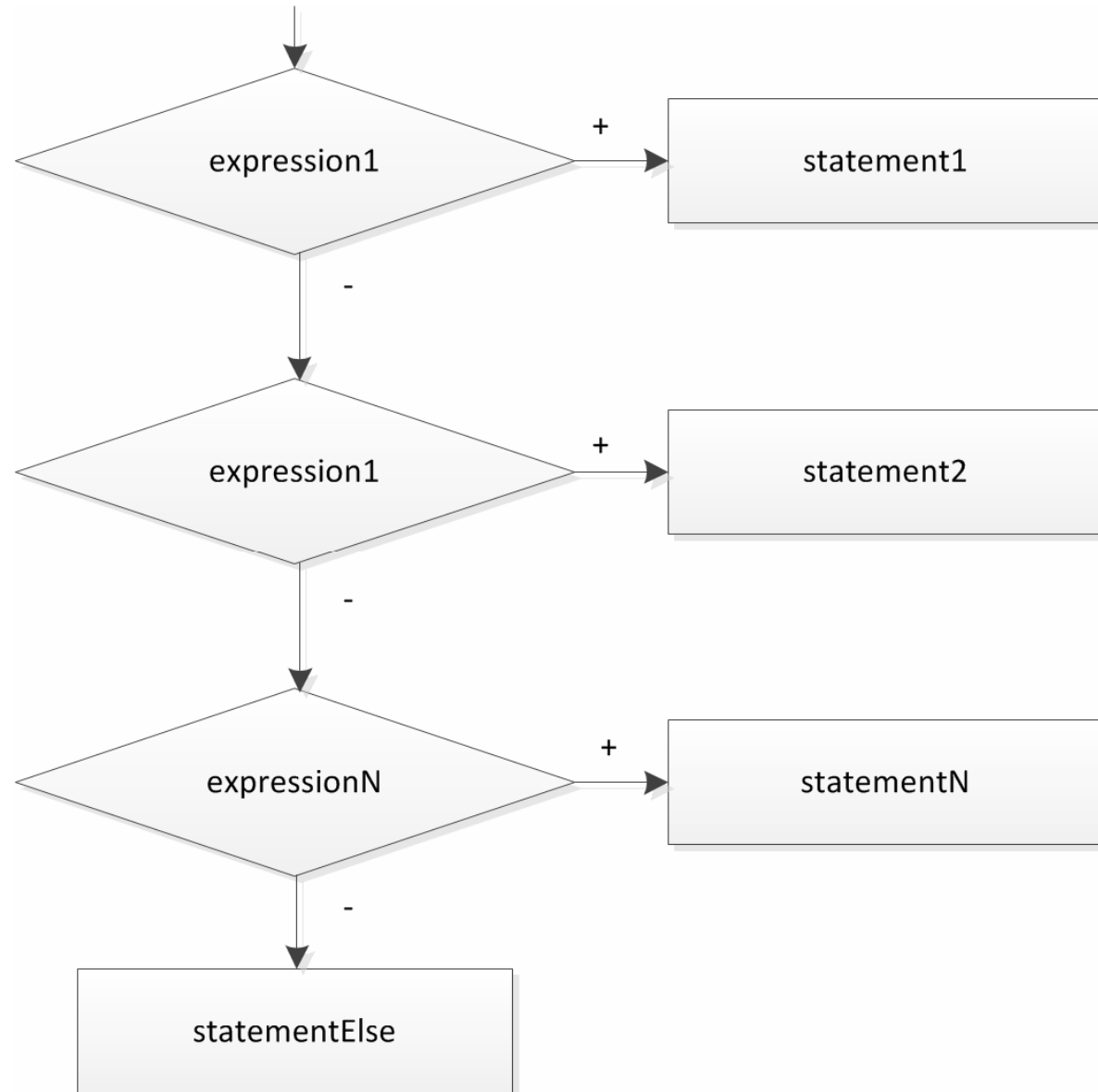
Видео- и интерактивные курсы:

1. <http://www.mathworks.com> > Academia > Interactive Tutorials
 - MATLAB Tutorial (~ 3 часов)
 - Simulink Tutorial (~ 2 часов)
 - Signal Processing Tutorial (~ 2.5 часов)
 - Computational Mathematics Tutorial (~ 2.5 часов)
2. <http://www.teachvideo.ru> > Разное > Больше курсов > Научный софт
 - MATLAB (25 уроков, всего ~1 часа)
3. <http://www.exponenta.ru> > Matlab
4. <http://youtube.com> > Канал MATLABinRussia

Оператор ветвления if

```
if expression1
  statement1
elseif expression2
  statement2
...
else
  statementElse
end
```

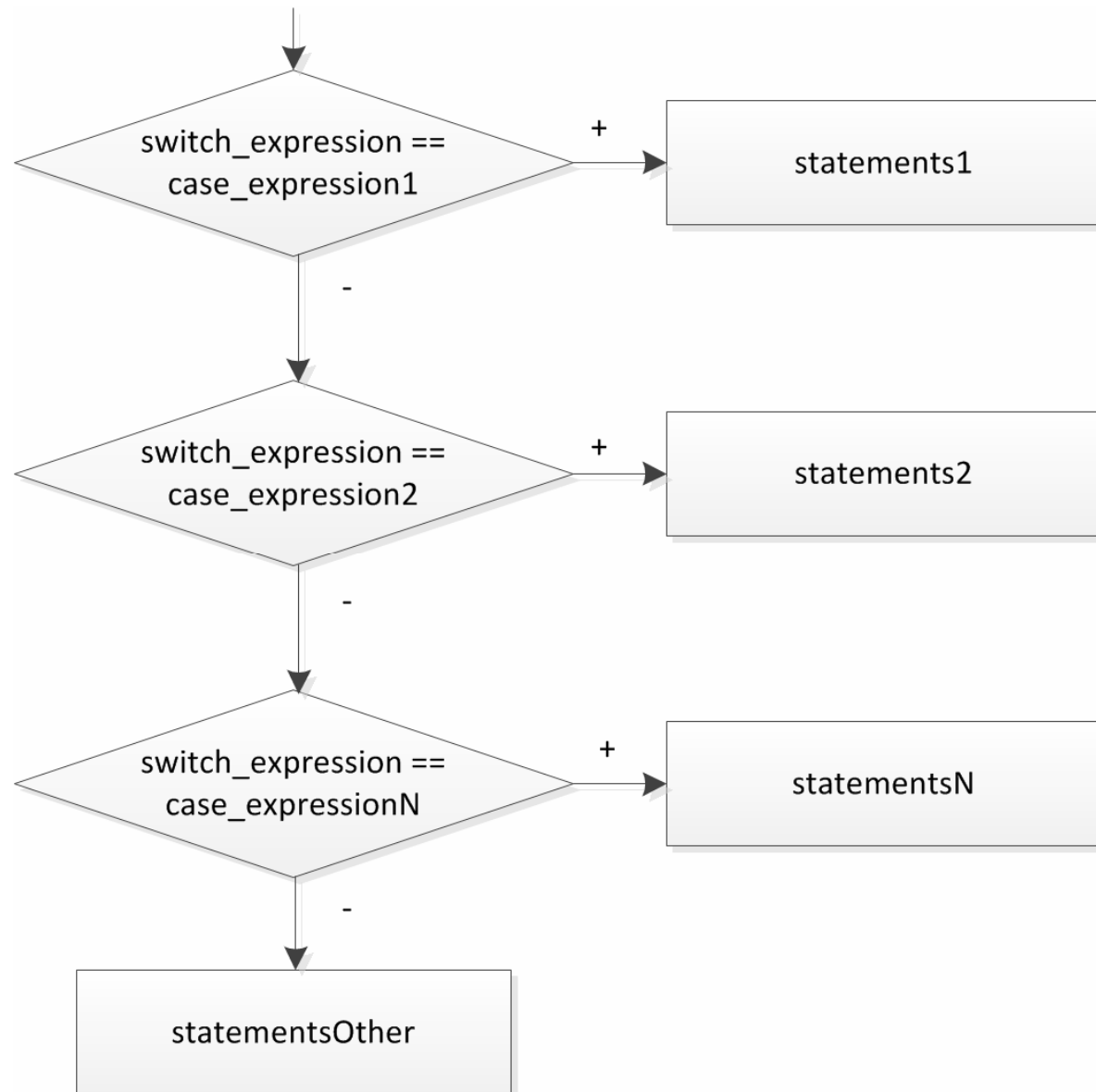
```
if r == c
  myData(r,c) = 2;
elseif abs(r - c) == 1
  myData(r,c) = -1;
else
  myData(r,c) = 0;
end
```



Оператор ветвления switch

```
switch switch_expression
case case_expression1
statements1
case case_expression2
statements2
case case_expressionN
statementsN
otherwise
statementsOther
end
```

```
switch mynumber
case -1
disp('negative one');
case 0
disp('zero');
otherwise
disp('other value');
end
```



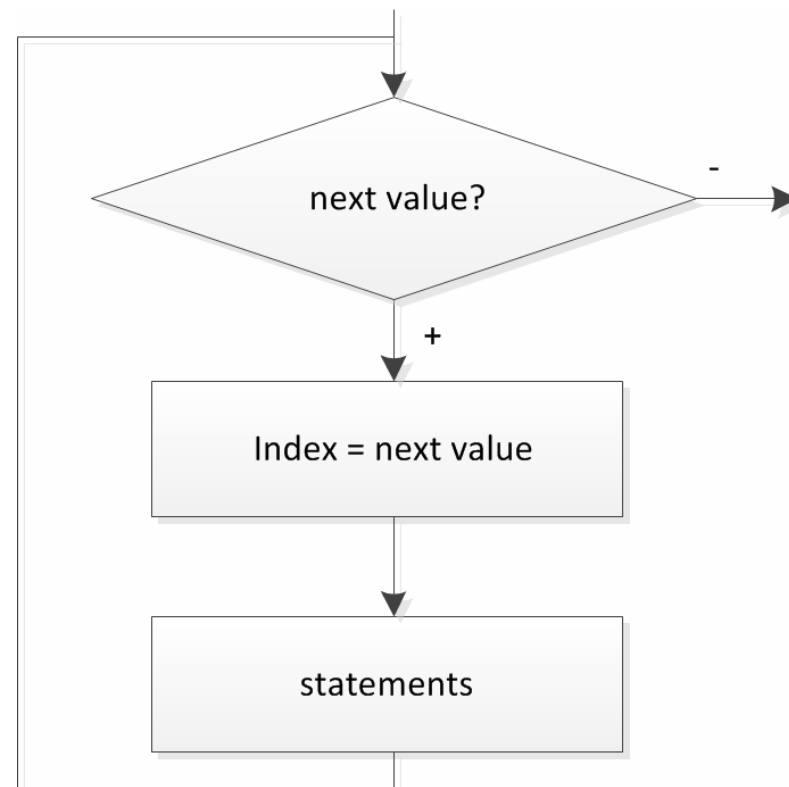
Оператор цикла for

```
for index = values  
    statements  
end
```

```
for m = 1:k  
    for n = 1:k  
        hilbert(m,n) = 1/(m+n -1);  
    end  
end
```

```
for s = 1.0: -0.1: 0.0  
    disp(s)  
end
```

```
for s = [1,5,8,17]  
    disp(s)  
end
```



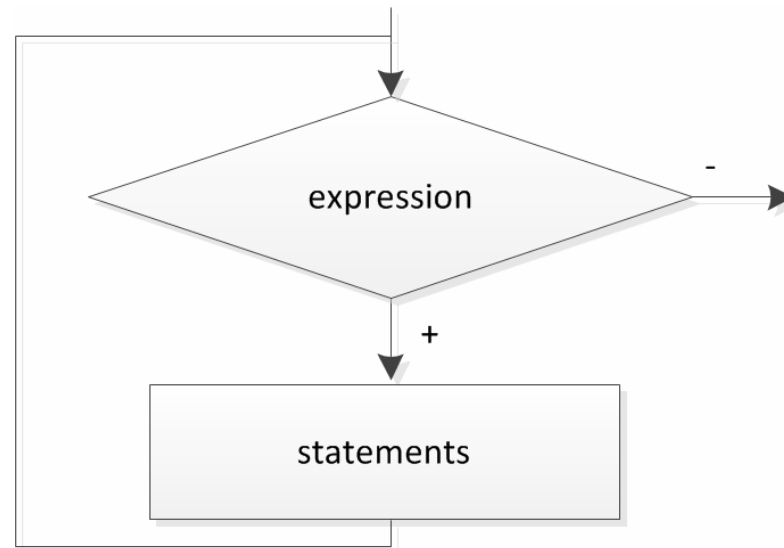
Можно распараллелить с помощью **parfor**

Оператор цикла while

```
while expression  
    statements  
end
```

```
while nFactorial < 1e100  
    n = n + 1;  
    nFactorial = nFactorial * n;  
end
```

```
while exist('myfunction.m') ...  
    && (myfunction(x) >= pi)  
    disp('Condition is true')  
    break  
end
```



Типичная структура программы

```
1 - clear all; clc; close all; % Очистка
2
3 - T = 0.001; % Шаг по времени
4 - qcno_dB = [15 30 45]; % Варьируемый параметр
5 - N = 10000; % Объем статистики
6 - F = [1 T; 0 1];
7 - t = 0:T:100*T; % Ось времени
8
9 - xn = nan(1, N); % Выделение памяти под массивы
10 - xq = nan(1, length(qcno_dB));
11
12 - for q = 1:length(qcno_dB) % Цикл по c/ш
13 -
14 -     for n = 1:N % Цикл набора статистики
15 -         ksi = randn(1,length(t)) / qcno_dB(q);
16
17 -         X = [0; 0]; % Инициализация переменных
18 -         for k = 1:length(t) % Цикл по времени
19 -             X = F*X + ksi(k); % Моделируемый процесс
20 -         end
21
22 -         xn(n) = X(1); % Сохранение статистики
23 -     end
24
25 -     xq(q) = std(xn); % Обработка статистики
26 - end
27
28 - figure(1); % Вывод результата
29 - plot(qcno_dB, xq)
30 - xlabel('q_{c/n0}, dBHz');
31 - ylabel('std x(end)');
```

Подготовка среды

Определение параметров

Выделение памяти

Цикл варьирования параметров

Цикл набора статистики

Цикл реализации

Обработка и вывод результатов