

## Лабораторная работа №2. Программирование в MatLab

### Цикл с известным количеством повторений

MatLAB предоставляет пользователю два способа организации цикла. Первый из них цикл с известным количеством повторений. Блок-схема этого цикла представлена на рис.1.5.

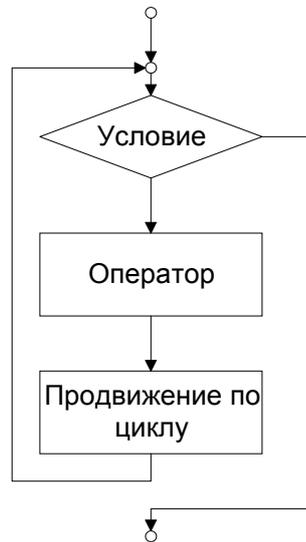


Рис. 1.5. Блок-схема цикла for

Формат оператора:

**for** переменная = начальное значение : [шаг :] конечное значение  
операторы  
**end**

Поле шаг в конструкции оператора не является обязательным. Например:

```
for i=1 : 100
    x(i) = sin(2*PI*i /100) ;
end
```

Допустимо задание переменной цикла в виде вектора:

```
for i= [1 2 4 8 16 32]
    x(i) = sin(2*PI*i /100) ;
end
```

Тело цикла обязательно заканчивается служебным словом **end**.

При работе с циклом **for** допустимо использование оператора прерывания цикла **break**. При выполнении оператора **break** работа цикла завершается и управление передается на следующий после конца цикла оператор. Блок-схема иллюстрирующая использование оператора **break** приведена на рис.1.5.

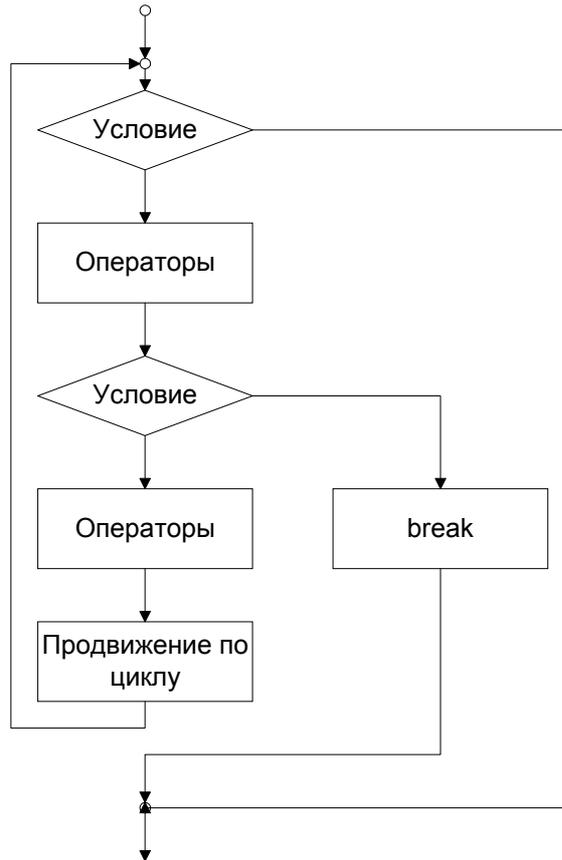


Рис. 1.5. Иллюстрация использования оператора break

Текст программы соответствующий рис.1.5.

```

for i=1:100
    операторы
    if a(i) == 0
        break
    end
    операторы
end
  
```

### ***Цикл с неизвестным количеством повторений***

Кроме цикла **for** в программировании на MatLAB используется цикл **while**. В отличие от цикла **for** в структуре цикла **while** не предусмотрены элементы для изменения переменной, по которой организован цикл. Эти элементы должен предусмотреть сам пользователь в операторах цикла.

Блок-схема цикла представлена на рис.1.6.

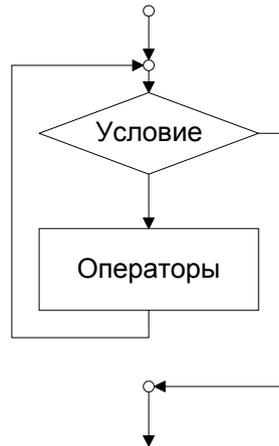


Рис. 1.6. Блок-схема цикла while

Формат оператора цикла с неизвестным количеством повторений:

```
while условие
    операторы
end
```

Тело цикла обязательно заканчивается служебным словом **end**.

Например, программа определения точности вычислений:

```
a=1 ;
while a+1 != 1
    a=a/2 ;
end
a
```

Последняя строчка программы выведет на экран значение переменной **a**.

### Создание собственных функций

Шаблон описания функции в MatLAB выглядит следующим образом:

```
function результат = имя ( аргументы )
    операторы
```

Первая строчка является обязательной, указывающей что это функциональный **M** файл. После нее располагаются операторы - тело функции.

Имя функции обязательно должно совпадать с именем файла в котором она размещается. Правила на задание имен функций такие же как и на имена переменных. Аргументы - это то что передается в функцию в качестве параметров. Результат - то что будет возвращено из функции в вышестоящую программу или в рабочую область самого MatLAB. Если возвращается одна переменная (массив) то указывается ее имя.

Пример:

```
function y=minf(x, n)
y=x(1) ;
for i=1:n
    if x(i)<y
        y=x(i) ;
    end
end
```

Последнее значение, присвоенное переменной результата **y** будет передано в вышестоящую программу или в рабочую область самого MatLAB.

Если из функции необходимо вернуть несколько значений (или массивов), то они перечисляются в прямоугольных скобках.

```
function [y, k]=minf(x, n)
```

```
y=x(1);
```

```
k=1;
```

```
for i=1:n
```

```
    if x(i)<y
```

```
        y=x(i);
```

```
        k=i;
```

```
    end
```

```
end
```

Отдельным вопросом при использовании функций являются **S** функции - фактически являющиеся программами на языке MATLAB. От обычной функции их отличают отсутствие строки описания **function**. Созданные в них переменные являются глобальными для модуля вызвавшего соответствующую **S** функцию.

### ***Задание на лабораторную работу***

Необходимо написать программу в MatLab. При этом для операций с матрицами не допускается использование стандартных функций.

В качестве подготовки к лабораторной работе необходимо составить алгоритм программы в виде блок-схемы.

Оглавление

Лабораторная работа №2. Программирование в MatLab.....	1
Цикл с известным количеством повторений .....	1
Цикл с неизвестным количеством повторений.....	2
Создание собственных функций .....	3
Задание на лабораторную работу.....	4