

ЛАБОРАТОРНАЯ РАБОТА №2

Работа с графикой

Указатели на функции. В М-языке функции могут выступать в роли переменных. Ключевое отличие от языков C/C++ здесь состоит в механизме получения новых функций. В языке C был тип переменных «указатель на функцию», который был жестко привязан к входным и выходным аргументам и не допускал дальнейшей модификации. Так, если функция `f` принимала два аргумента, то получить из указателя на неё указатель на функцию одного аргумента (зафиксировав значение второго) стандартными средствами было невозможно. Кроме того, указатели были привязаны не только к входным и выходным параметрам, но и к происхождению функции: указатель на «обычную» функцию не мог указывать на функцию с такими же входными и выходными аргументами, являющуюся членом какого-нибудь класса.

Пример задания переменных-функций:

```
f = @(x,y) exp(-(x.^2 + y.^2));  
g = @(x) f(x,0);
```

После этого `f` и `g` можно использовать в коде, как «обычные» функции вызовами типа `f(4,3)` или `g(rand(18))`. Этот механизм полезен для работы с алгоритмами, которые имеют функцию в качестве входного параметра — например, алгоритмы интегрирования, решения уравнений, построения линий уровня и т.д.

Подробнее см. `help` по разделу «Function handles».

Графические команды. Команды для работы с двумерной графикой Вам уже известны из предыдущих лабораторных работ — в первую очередь, это команда `plot`. Для удобного создания сеток можно пользоваться командами `linspace` и `logspace`. Иногда бывает полезно разместить на одной фигуре несколько осей с различными графиками, для этого служит команда `subplot`.

Для рисования трехмерных параметрических кривых, $(x(t), y(t), z(t)) \in \mathbb{R}^3$, $t \in \mathbb{R}$, используется команда `plot3`.

Для рисования трехмерных поверхностей, $f(x, y) \in \mathbb{R}$, сетка задаётся командой `meshgrid`. Для рисования поверхностей используется команда `surf`.

Для рисования поверхностей уровня, $f(x, y, z) = \text{const}$, используется набор команд с приставкой «iso»: `isosurface`, `isonormals` и прочие.

Рисование анимаций. Рисование анимаций осуществляется с помощью команды `getframe`, которая записывает информацию о текущей фигуре (`figure`). Пример:

```
% Выделим память под кадры анимации
```

```
mov(1:nFrames) = struct('cdata', [], 'colormap', []);
```

```
x = linspace(0,1,100);
```

```
for i = 1:nFrames
```

```
    plot(x, sin(i*x)); % Не используется hold on: каждый новый кадр затирает предыдущий
```

```
    mov(i) = getframe();
```

```
end
```

После записи анимации её можно воспроизвести командой `movie(mov)` или записать в видеофайл командой `avifile`. При этом важно установить поле `'Compression'` в `'none'`, поскольку по умолчанию используется не везде поддерживаемый алгоритм компрессии. При наличии внешних кодеков эту проблему можно решить средствами среды, но обычно бывает проще воспользоваться внешним инструментом для конвертации некомпрессированного файла (которые имеют огромный размер). Например, в *NIX-подобных системах для этого можно воспользоваться утилитой `mencoder`:

```
mencoder orig.avi -o result.avi -of lavf -oac mp3lame -ovc lavc -lavcopts vcodec=mpeg4
```

Альтернативный механизм записи видео предоставляется командами `VideoWriter` и `writeVideo`.