

Компьютерные технологии в научных исследованиях и образовании

Юрий Анатольевич Богоявленский, заведующий кафедрой Информатики и математического обеспечения, к.т.н., доцент, ybgv

Построение графиков

Введение

Инструменты высокого уровня

Трёхмерные графики

Функции mesh и surf

Функции meshgrid и plot3

Аннотирование графиков

Трёхмерные графики

Функция mesh строит трёхмерный график (объект типа surface) с помощью многогранной полигональной сетки (англ. polygon mesh) четырёхугольников и может быть задана в следующих формах:

```
mesh (x, y, z)
```

```
mesh (z)
```

```
mesh ( . . . , c)
```

```
mesh ( . . . , prop, val, . . . )
```

```
mesh (hax, . . . )
```

```
h = mesh ( . . . )
```

Назначение параметров.

- x, y — одномерные вектора, задающие значения по осям абсцисс и ординат для построения графика;
- z — матрица значений по оси аппликат визуализируемой поверхности $z=f(XY)$ где XY и YY двумерные матрицы, представляющие полную

двумерную сетку прямоугольников. Эти матрицы обычно вычисляются функцией `meshgrid(x,y)` так, что строки являются копиями ее параметра — вектора x , а столбцы YY — копиями параметра — вектора y . При этом нужно задать матричное вычисление $z=f(XX,YY)$, которое определит элементы двумерной матрицы аппликат z для соответствующих пар элементов XX и YY . Если задан только единственный параметр — матрица z , то ее элементы трактуются как значения аппликат, а сетка прямоугольников формируется из векторов $x = 1:\text{columns}(z)$, $y = 1:\text{rows}(z)$, где `columns(z)` вычисляет количество столбцов, а `rows(z)` — количество строк матрицы. При этом столбцы z соответствуют значениям абсцисс графика, а строки — значениям ординат. Цвет элементов графика вычисляется путем линейного масштабирования значений z в соответствии с диапазоном текущей цветовой палитры, которую можно изменить, например, функцией `colorbar`;

- `c` — матрица цвета, задающая цвет элементов графика независимо от значений матрицы z ;
- `prop, val` — пары свойство/значение передаются подлежащему построению графика — объекту типа `surface`;
- `hax` — если задан этот фактический параметр — значение дескриптора объекта `axes` (оси), то график будет строиться в осях, определенных этим объектом, а не в определенных текущим объектом `axes` (оси), ссылка на дескриптор которого задается функцией `gca`;
- `h` — необязательное возвращаемое значение — дескриптор созданного объекта типа `surface`;

Полный список свойств задокументирован в [Док, с. 463 — 466].

Есть еще три функции с полностью аналогичным синтаксисом, но имеющие в конструируемой поверхности следующие отличия:

`meshc` — строит также контурные линии;

`meshz` — строит еще вертикальные линии аппликат;

`surf` — заполняет четырехугольники цветом.

Рассмотрим простой пример.

```
ybgv@ybgv-home:~> octave -q -p ~/MyOct -p ~/MyOct/Plotting/
octave:1> Simpl3D
+ echo on
+
+ fileread("Simpl3D.m")
warning: fopen: '/home/ybgv/MyOct/Simpl3D.m' found by searching
load path
warning: called from
    fileread at line 42 column 7
    Simpl3D at line 5 column 1

ans = # mesh и surf - простые примерв

echo on

fileread("Simpl3D.m")

X=linspace(-1,2,8)
Y=linspace(-3,3,8)

# получение 2D сетки

[V W]=meshgrid(X,Y)

# аппликаты "седла"

Z=3*(V.^4 - W.^2);

f = figure('position', [10 10 1100 1400]);
```

```

subplot(2,2,1)
mesh(Z);title ("Simple mesh");
subplot(2,2,2)
meshz(Z);title ("+ applicats");
subplot(2,2,3)
meshc(Z);title ("+ contour lines");
subplot(2,2,4)
surf(Z);title ("Shaded rectangles");
print simple3d.pdf;print simple3d.png

# конец файла Simpl3D.m
+
+ X=linspace(-1,2,8)
X =

    -1.0000    -0.5714    -0.1429     0.2857     0.7143     1.1429     1.5714     2.0000

+ Y=linspace(-3,3,8)
Y =

    -3.0000    -2.1429    -1.2857    -0.4286     0.4286     1.2857     2.1429     3.0000

+
+ # получение 2D сетки
+
+ [V W]=meshgrid(X,Y)
V =

    -1.0000    -0.5714    -0.1429     0.2857     0.7143     1.1429     1.5714     2.0000
    -1.0000    -0.5714    -0.1429     0.2857     0.7143     1.1429     1.5714     2.0000
    -1.0000    -0.5714    -0.1429     0.2857     0.7143     1.1429     1.5714     2.0000
    -1.0000    -0.5714    -0.1429     0.2857     0.7143     1.1429     1.5714     2.0000
    -1.0000    -0.5714    -0.1429     0.2857     0.7143     1.1429     1.5714     2.0000

```

```
-1.0000 -0.5714 -0.1429 0.2857 0.7143 1.1429 1.5714 2.0000
-1.0000 -0.5714 -0.1429 0.2857 0.7143 1.1429 1.5714 2.0000
-1.0000 -0.5714 -0.1429 0.2857 0.7143 1.1429 1.5714 2.0000
```

W =

```
-3.0000 -3.0000 -3.0000 -3.0000 -3.0000 -3.0000 -3.0000 -3.0000
-2.1429 -2.1429 -2.1429 -2.1429 -2.1429 -2.1429 -2.1429 -2.1429
-1.2857 -1.2857 -1.2857 -1.2857 -1.2857 -1.2857 -1.2857 -1.2857
-0.4286 -0.4286 -0.4286 -0.4286 -0.4286 -0.4286 -0.4286 -0.4286
0.4286 0.4286 0.4286 0.4286 0.4286 0.4286 0.4286 0.4286
1.2857 1.2857 1.2857 1.2857 1.2857 1.2857 1.2857 1.2857
2.1429 2.1429 2.1429 2.1429 2.1429 2.1429 2.1429 2.1429
3.0000 3.0000 3.0000 3.0000 3.0000 3.0000 3.0000 3.0000
```

+

```
+ # аппликаты "седла"
```

+

```
+ Z=3*(V.^4 - W.^2);
```

+

```
+ f = figure('position', [10 10 1100 1400]);
```

```
+ subplot(2,2,1)
```

```
+ mesh(Z);title ("Simple mesh");
```

```
+ subplot(2,2,2)
```

```
+ meshz(Z);title ("+ applicats");
```

```
+ subplot(2,2,3)
```

```
+ meshc(Z);title ("+ contour lines");
```

```
+ subplot(2,2,4)
```

```
+ surf(Z);title ("Shaded rectangles");
```

```
+ print simple3d.pdf;print simple3d.png
```

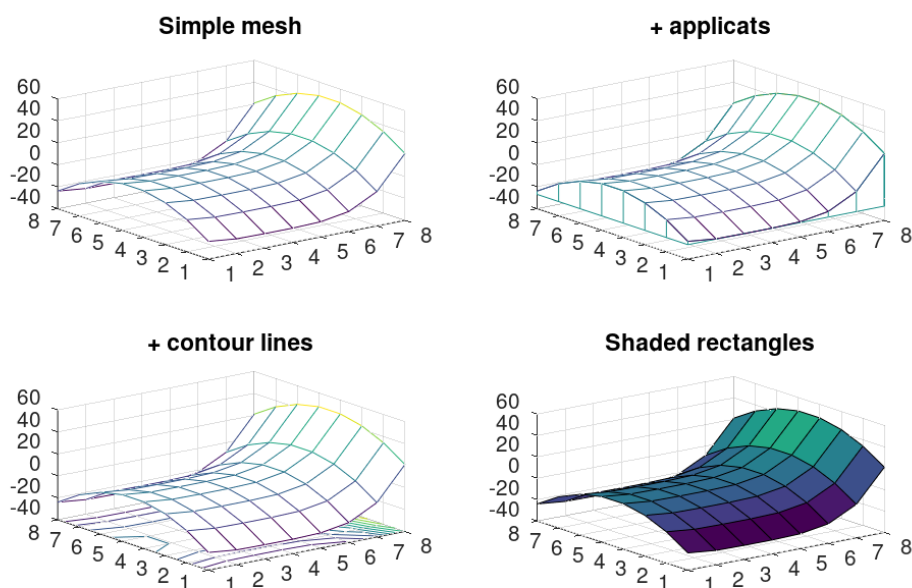
+

```
+ # конец файла Simple3D.m
```

+

```
octave:2>
```

Эта программа построит графики:



Отметим также, что функции `surf` с аналогичным синтаксисом, строит такую же поверхность как функция `surf` с дополнением контурными линиями.

Функции `meshgrid` и `plot3`

Функция `meshgrid` строит матрицы, представляющие полную двумерную сетку прямоугольников и может быть задана в формах:

```
[xx, yy] = meshgrid (x, y)
```

```
[xx, yy, zz] = meshgrid (x, y, z)
```

```
[xx, yy] = meshgrid (x)
```

```
[xx, yy, zz] = meshgrid (x)
```

Основное назначение функции — формирование значений абсцисс и ординат трехмерного графика, попарные комбинации которых используются как аргументы для матричных вычислений значений аппликат графика.

Назначение входных параметров.

- x, y — одномерные векторы, задающие требуемые значения по осям абсцисс и ординат графика;
- z — необязательный одномерный вектор, задающий требуемые значения по оси аппликат графика;

Назначение выходных параметров для форм $[xx, yy] = \text{meshgrid}(x, y)$ и $[xx, yy] = \text{meshgrid}(x)$.

xx — двумерная матрица, строки которой совпадают с x ;

yy — двумерная матрица, столбцы которой совпадают с y если входной параметр y не задан, то он принимается равным x .

Если заданы входной параметр z и/или выходной zz , то результатом будут трехмерные матрицы xx , yy и zz . Если входной параметр z не задан, но присутствует выходной параметр zz , то z принимается равным y .

Отметим, что для некоторых функций нужно использовать представляющие сетки многомерные матрицы, генерируемые функцией `ndgrid`, которую мы не рассматриваем.

Функция `plot3` отображает произвольные трехмерные данные, не требуя, чтобы они формировали поверхность. Отметим, что в ее описании явно представлены случаи, когда фактическими параметрами являются комплексные числа. Функция может быть задана в следующих формах:

```
plot3 (x, y, z)
```

```
plot3 (x, y, z, prop, value, . . . )
```

```
plot3 (x, y, z, fmt)
```

```
plot3 (x, cplx)
```

```
plot3 (cplx)
```

```
plot3 (hax, . . . )
```

```
h = plot3 ( . . . )
```

Назначение параметров.

x , y , z — в первых трех формах задают координаты подлежащих от отображению точек. Если все три параметра векторы (требуется одинаковое число элементов), то отображается непрерывная кривая, проходящая через точки. Если все три параметра матрицы (требуется одинаковое количество столбцов и одинаковое число элементов в столбцах), то каждый столбец отображается как отдельная кривая;

`cplx` — вектор или матрица с комплексными элементами. При вызове в форме `plot3 (x, cplx)` значение действительной части трактуется как значение ординаты, а мнимой — как аппликаты отображаемой точки. При вызове в форме `plot3 (cplx)` за значения абсцисс отображаемой точки принимается номер строки а значения действительной и мнимой части трактуются как в предыдущем случае.

`prop`, `value` (пары свойство-значение) и `fmt` — полностью эквивалентны аналогичным параметрам функции `plot`.

`hax` и выходной параметр `h` — также эквивалентны аналогичным параметрам функции `plot`.

Параметры также могут быть заданы группами по три:

```
plot3 (x1, y1, z1, x2, y2, z2, ...).
```

При этом каждая их тройка трактуется как отдельная кривая или набор кривых.

Примеры.

```
# График трехмерной спирали
t = 0:0.1:10*pi;
r = linspace (0, 1, numel (t));
z = linspace (0, 1, numel (t));
plot3 (r.*sin (t), r.*cos (t), z);
xlabel ("r.*sin (t)");
ylabel ("r.*cos (t)");
zlabel ("z");
title ("plot3 display of 3-D helix");
```


plot3 display of 3-D helix

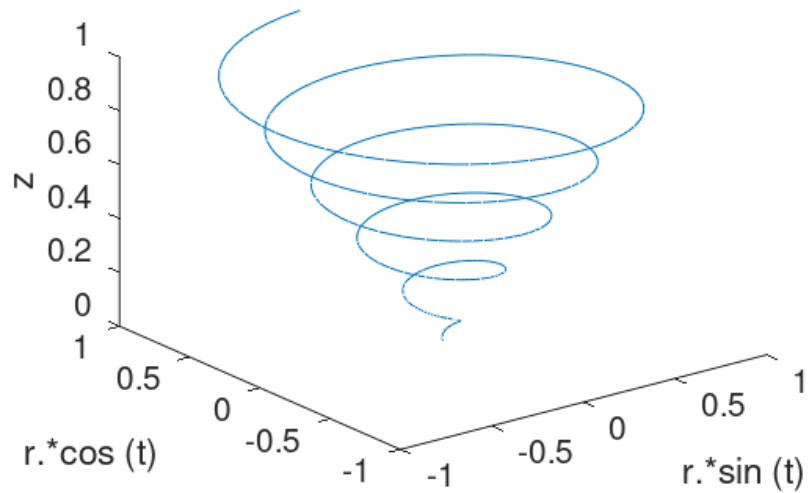


График вертикальной спирали

```
z = [0:0.05:5];
```

```
plot3 (cos (2*pi*z), sin (2*pi*z), z, ";helix;");
```

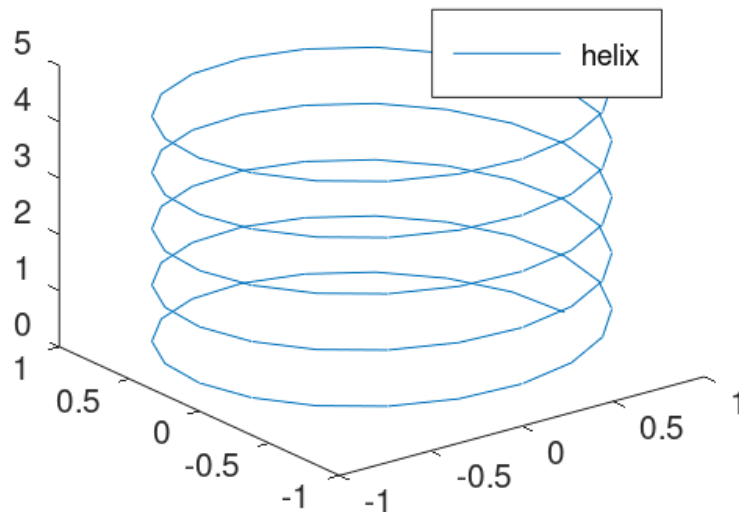
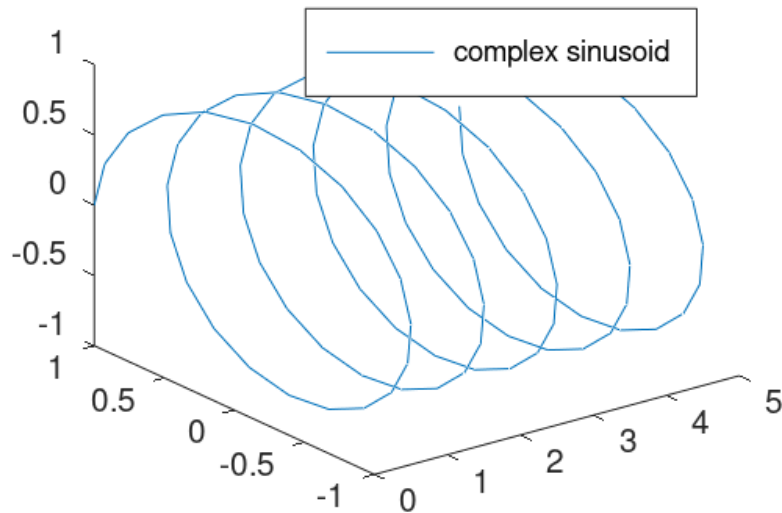


График комплексной синусоиды

```
z = [0:0.05:5];  
plot3 (z, exp (2i*pi*z), ";complex sinusoid;");
```



Аннотирование графиков

В octave включены разнообразные удобные инструменты размещения на графиках дополнительной поясняющей информации. Например программа:

```
# Аннотирование. Пример 1.  
  
# размер рисунка  
  
f = figure('position', [10 10 500 300]);  
  
# задание кривой
```

```
x = -10:0.1:10;
plot (x, sin (x));
title ("Синусоида для x = -10:0.1:10");
xlabel ("x");
ylabel ("sin (x)");

# координаты текста и сам текст

text (-5, 0, "Какой-то текст");
legend ("sin (x)");

# включить сетку

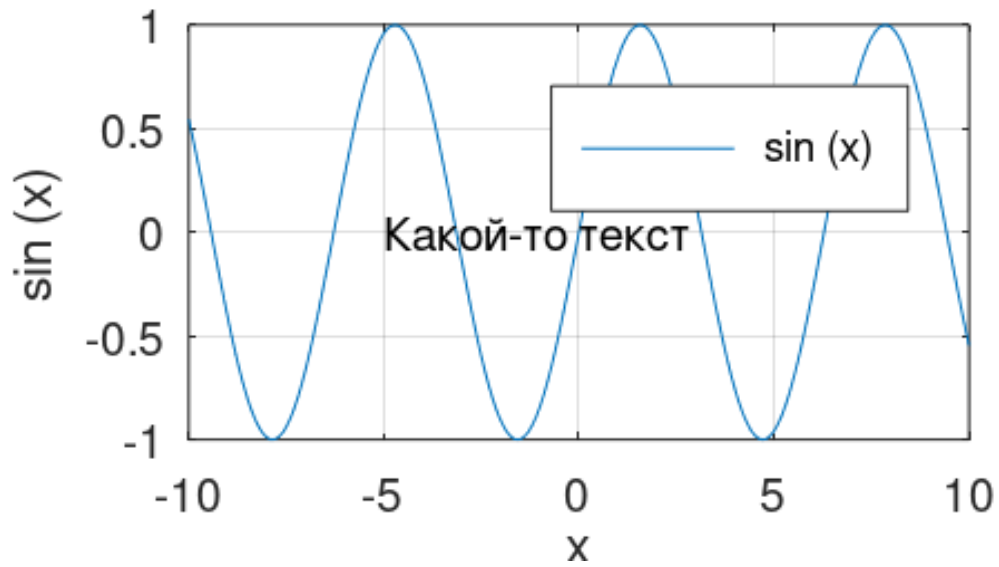
grid on;

# -svgconvert - ключ для символов в коде UTF-8.

print -svgconvert ann1.pdf;print -svgconvert ann1.png
```

построит график:

Синусоида для $x = -10:0.1:10$



Для аннотирования графиков реализованы следующие функции: title, legend, text, xlabel, ylabel, zlabel, clabel, box, grid, colorbar, annotation. Мы не рассматриваем описания этих функций, представленные в [Док, с. 387 — 395].