

Интерполяция

```
%% Interpolation
```

интерполяция (интерполирование) – процесс нахождения промежуточных значений между узловыми точками.

```
%% Линейная интерполяция
```

```
% vq = interp1(x,v,xq)
```

```
% vq = interp1(x,v,xq,method)
```

```
% vq = interp1(x,v,xq,method,extrapolation)
```

```
% vq = interp1(v,xq)
```

```
% vq = interp1(v,xq,method)
```

```
% vq = interp1(v,xq,method,extrapolation)
```

```
% pp = interp1(x,v,method,'pp')
```

```
x = 0:pi/2:2*pi
```

```
v = sin(x)
```

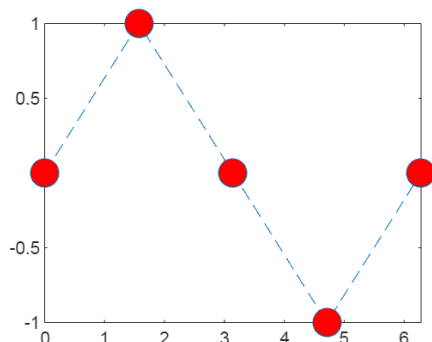
x = 0	1.5708	3.1416	4.7124	6.2832
v = 0	1.0000	0.0000	-1.0000	-0.0000

```
x = 0:pi/2:2*pi
```

```
v = sin(x)
```

```
h=plot(x,v,'o--')
```

```
h.set('MarkerSize',16,'MarkerFaceColor','r')
```



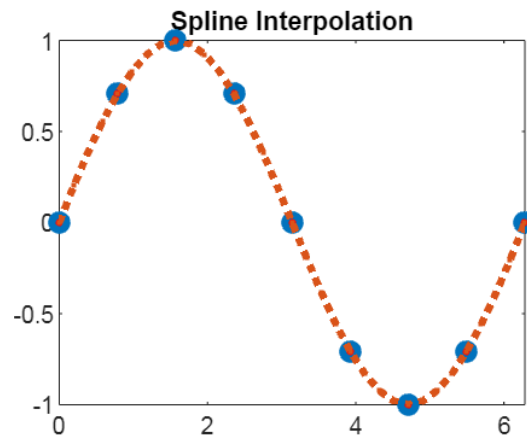
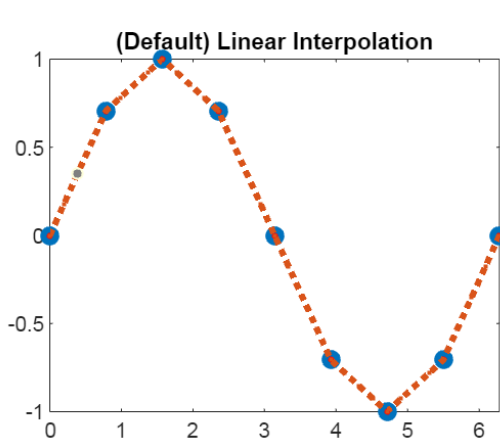
```

%%
figure
x = 0:pi/4:2*pi
v = sin(x)

% В точках xq восстановить значения функции
xq = 0:pi/16:2*pi;
% восстановленные значения
vq1 = interp1(x,v,xq);

h=plot(x,v,'o',xq,vq1,':');
h.set('MarkerSize',6,'MarkerFaceColor','r')
h.set('LineWidth',3)
xlim([0 2*pi]);
title('(Default) Linear Interpolation');

```



```

%% Spline Interpolation
figure
vq2 = interp1(x,v,xq,'spline');

h=plot(x,v,'o',xq,vq2,':');
xlim([0 2*pi]);
title('Spline Interpolation');
h.set('MarkerSize',6,'MarkerFaceColor','r')
h.set('LineWidth',3)

```

```
% Интерполяция по значениям функции
```

```
v = [0 1.41 2 1.41 0 -1.41 -2 -1.41 0];
```

```
% точки в которых нужно восстановить значения функции
```

```
xq = 1.5:8.5;
```

```
% восстановление
```

```
vq = interp1(v,xq);
```

```
% графики
```

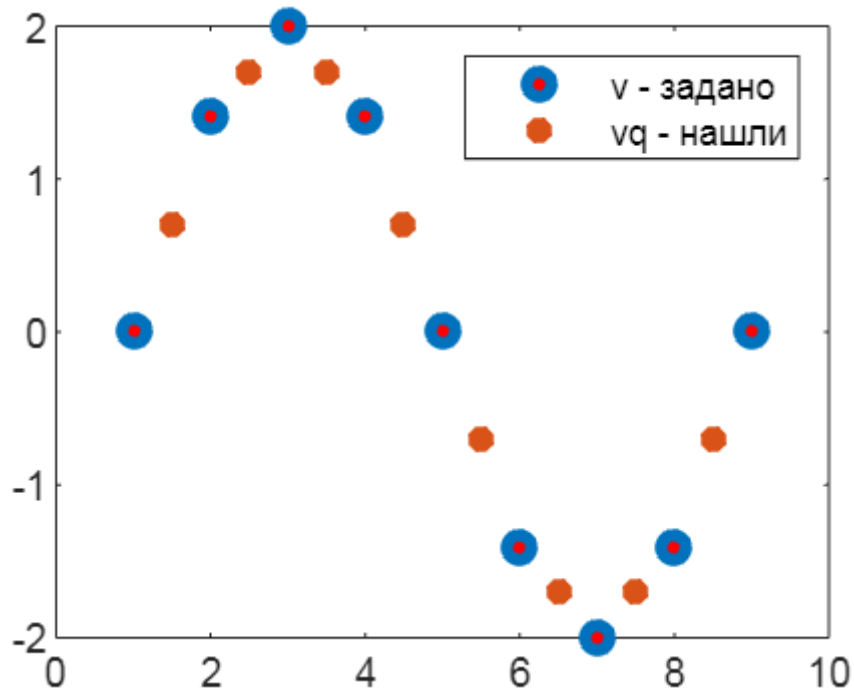
```
figure
```

```
h=plot((1:9),v, 'o',xq,vq, '*');
```

```
legend('v - задано', 'vq - нашли');
```

```
h.set('MarkerSize',6, 'MarkerFaceColor', 'r')
```

```
h.set('LineWidth',3)
```



```
% Интерполяция комплексных значений
```

```
x = 1:10;
```

```
v = (5*x)+(x.^2*1i);
```

```
xq = 1:0.25:10;
```

```
% интерполяция
```

```
vq = interp1(x,v,xq);
```

```
% графики мнимой и действительной частей
```

```
figure
```

```
h1=plot(x,real(v), '*r',xq,real(vq), '-r');
```

```
hold on
```

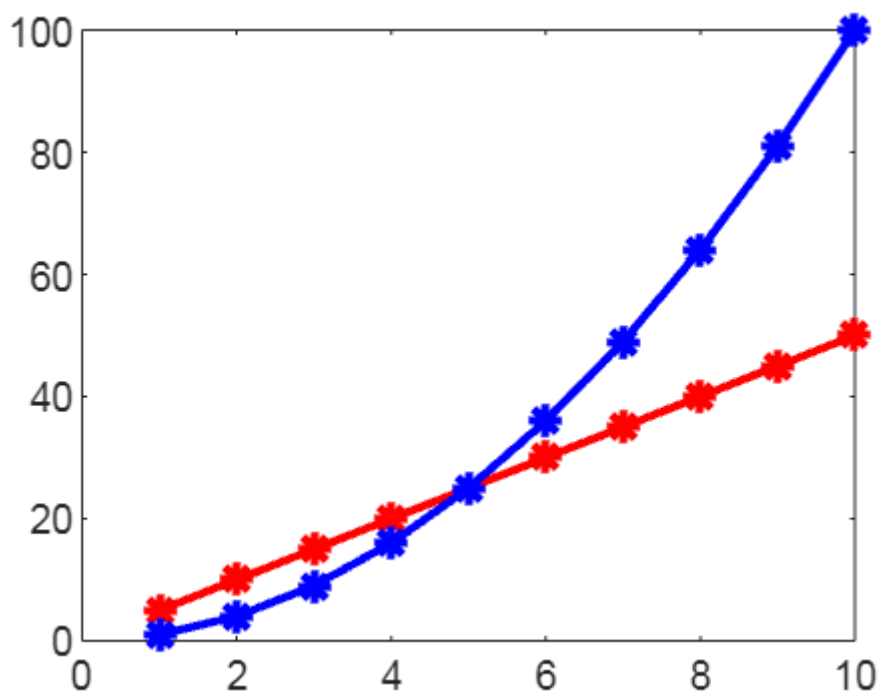
```
h2=plot(x,imag(v), '*b',xq,imag(vq), '-b');
```

```
h1.set('MarkerSize',8)
```

```
h1.set('LineWidth',2)
```

```
h2.set('MarkerSize',8)
```

```
h2.set('LineWidth',2)
```



```
%% Интерполяция с датами
```

```
x = (datetime(2024,1,1)...  
      :hours(4)...  
      :datetime(2024,1,2))';
```

```
x.Format = 'MMM dd, HH:mm';
```

```
T = [31 25 24 41 43 33 31]';
```

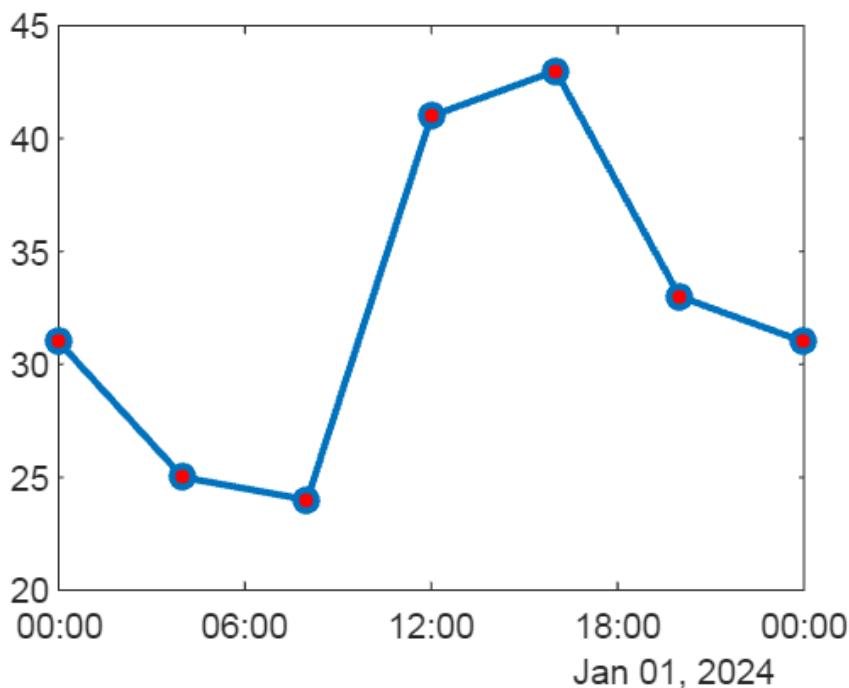
```
WeatherData = table(x,T,'VariableNames',...  
                    {'Time','Temperature'})
```

7x2 [table](#)

<u>Time</u>	<u>Temperature</u>
Jan 01, 00:00	31
Jan 01, 04:00	25
Jan 01, 08:00	24
Jan 01, 12:00	41
Jan 01, 16:00	43
Jan 01, 20:00	33
Jan 02, 00:00	31

```
h=plot(WeatherData.Time,  
WeatherData.Temperature,...  
      'o-')
```

```
h.set('MarkerSize',6,'MarkerFaceColor','r')  
h.set('LineWidth',2)
```



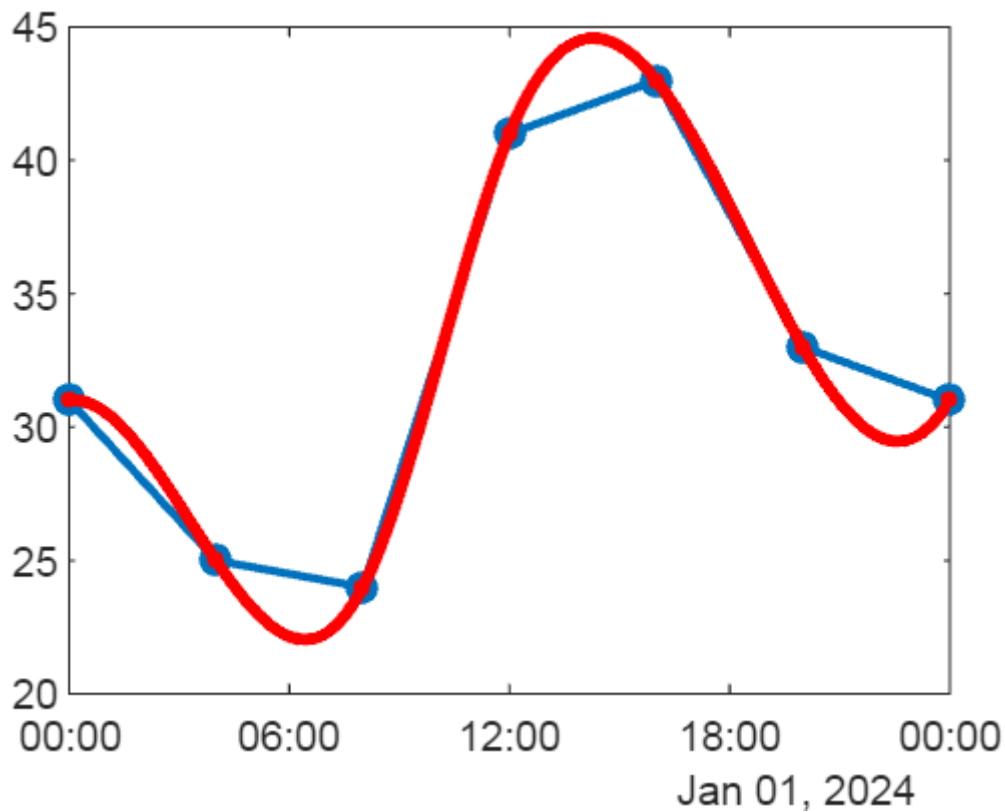
```
% Интерполируйте набор данных,
% чтобы предсказать температуру в течение
каждой минуты дня.
% Используйте 'spline' метод интерполяции.
```

```
xq = (datetime(2024,1,1)...
      :minutes(1):...
      datetime(2024,1,2))';
```

```
V = interp1(WeatherData.Time,...
            WeatherData.Temperature,...
            xq, 'spline');
```

```
% Постройте интерполированные точки.
```

```
hold on
hh=plot(xq,V,'r')
hh.set('LineWidth',3)
```



% Экстраполяция

% Выборка

```
x = [1 2 3 4 5];
```

```
v = [12 16 31 10 6];
```

% точки запроса, xq - расширяет область x

```
xq = [0 0.5 1.5 5.5 6];
```

% экстраполяция pchip

```
vq_pchip = interp1(x,v,xq,'pchip')
```

% экстраполяция linear

```
vq_linear = interp1(x,v,xq,'linear')
```

Метод	Описание	Непрерывность	Комментарии
'linear'	Линейная интерполяция. Интерполированное значение в точке запроса основано на линейной интерполяции значений в соседних узлах решетки в каждой соответствующей размерности. Это - метод интерполяции по умолчанию.	C^0	Требует по крайней мере 2 точек Требует большей памяти и время вычисления, чем самый близкий сосед
'nearest'	Интерполяция по ближайшему соседу. Интерполированное значение в точке запроса является значением в самом близком демонстрационном узле решетки.	Прерывистый	Требует по крайней мере 2 точек Скромные требования к памяти Самое быстрое время вычисления
'next'	Следующая соседняя интерполяция. Интерполированное значение в точке запроса является значением в следующем демонстрационном узле решетки.	Прерывистый	Требует по крайней мере 2 точек Те же требования к памяти и время вычисления как 'nearest'
'previous'	Предыдущая соседняя интерполяция. Интерполированное значение в точке запроса является значением в предыдущем демонстрационном узле решетки.	Прерывистый	Требует по крайней мере 2 точек Те же требования к памяти и время вычисления как 'nearest'
'pchip'	Сохраняющая форму кусочная кубичная интерполяция. Интерполированное значение в точке запроса основано на сохраняющей форму кусочной кубичной интерполяции значений в соседних узлах решетки.	C^1	Требует по крайней мере 4 точек Требует большей памяти и время вычисления, чем 'linear'
'cubic'	Кубическая свертка используется в MATLAB® 5.	C^1	Требует по крайней мере 3 точек
'v5cubic'	То же самое как 'cubic'.	C^1	Точки должны быть расположены равными интервалами Этот метод отстает к 'spline' интерполяция для нерегулярно распределенных данных Подобные требования к памяти и время вычисления как 'pchip'
'makima'	Модифицированный Акима кубическая интерполяция Эрмита. Интерполированное значение в точке запроса основано на кусочно-линейной функции полиномов со степенью самое большее три. Формула Акима изменяется, чтобы избежать перерегулирований.	C^1	Требует по крайней мере 2 точек Производит меньше волнистостей, чем 'spline', но не сглаживается так же настойчиво как 'pchip'

Метод	Описание	Непрерывность	Комментарии
			Расчет является более дорогим, чем 'pchip', но обычно меньше, чем 'spline' Требования к памяти похожи на те из 'spline'
'spline'	Интерполяция сплайна с помощью граничных условий и условий отсутствия узла. Интерполированное значение в точке запроса основано на кубичной интерполяции значений в соседних узлах решетки в каждой соответствующей размерности.	C^2	Требует по крайней мере 4 точек Требует большей памяти и время вычисления, чем 'pchip'

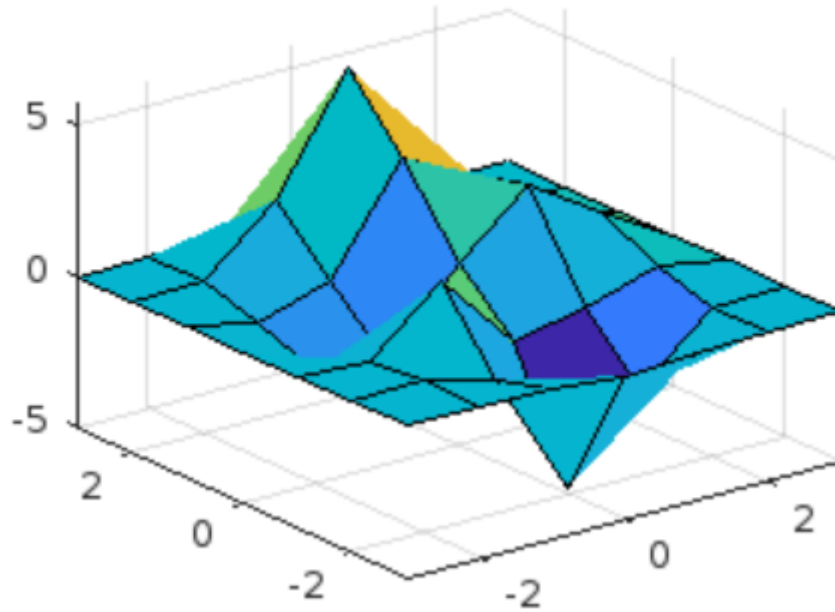
interp2

Интерполяция для 2D данных с координатной сеткой в meshgrid формате

```
%%
% Vq = interp2(X,Y,V,Xq,Yq)
% Vq = interp2(V,Xq,Yq)
% Vq = interp2(V)
% Vq = interp2(V,k)
% Vq = interp2(___,method)
% Vq = interp2(___,method,extrapval)

[X,Y] = meshgrid(-3:3);
V = peaks(X,Y);
figure
surf(X,Y,V)
title('Original Sampling');
```

Original Sampling



```
%% мелкая сетка
```

```
[Xq,Yq] = meshgrid(-3:0.25:3);
```

```
% интерполяция
```

```
Vq = interp2(X,Y,V,Xq,Yq);
```

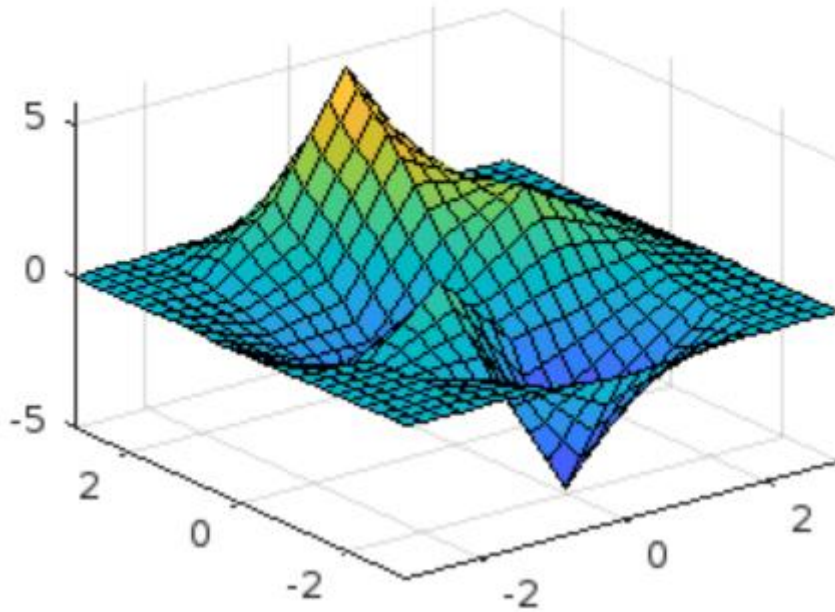
```
% результат
```

```
figure
```

```
surf(Xq,Yq,Vq);
```

```
title('Linear Interpolation Using Finer Grid');
```

Linear Interpolation Using Finer Grid



```
% cubic
```

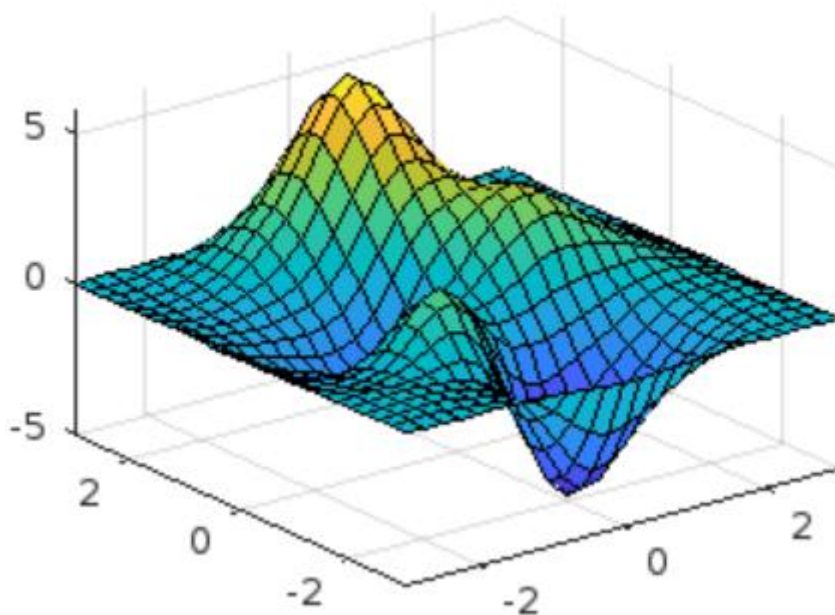
```
Vq = interp2(X,Y,V,Xq,Yq,'cubic');
```

```
figure
```

```
surf(Xq,Yq,Vq);
```

```
title('Cubic Interpolation Over Finer Grid');
```

Cubic Interpolation Over Finer Grid



```
% Аппроксимация
```

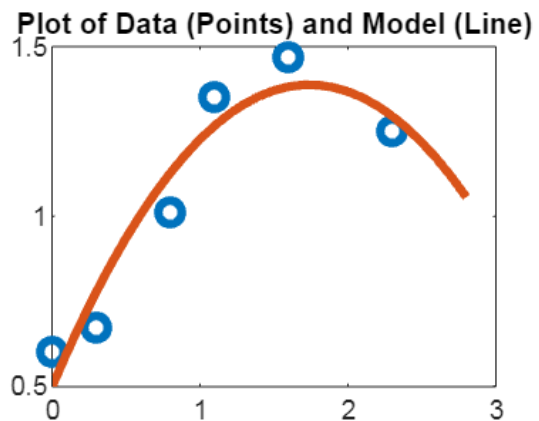
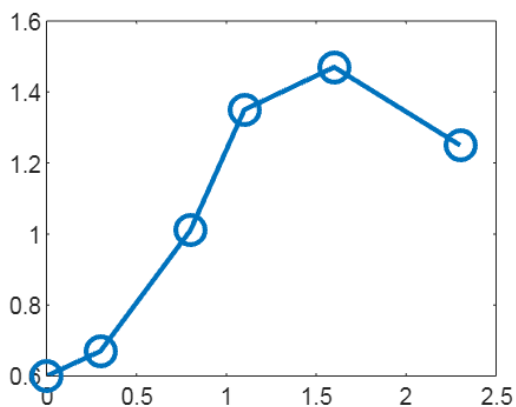
```
% набор данных
```

```
t = [0 0.3 0.8 1.1 1.6 2.3];
```

```
y = [0.6 0.67 1.01 1.35 1.47 1.25];
```

```
h=plot(t,y,'o-')
```

```
h.set('MarkerSize',12,'LineWidth',2)
```



```
% возвращает коэффициенты для полинома
```

```
t2 = 0:0.1:2.8;
```

```
p = polyfit(t,y,2)
```

```
p = -0.2942    1.0231    0.4981
```

```
% возвращает данные для полинома
```

```
y2 = polyval(p,t2);
```

```
figure
```

```
h=plot(t,y,'o',t2,y2)
```

```
h.set('MarkerSize',8,'LineWidth',3)
```

```
title('Plot of Data (Points) and Model (Line)')
```