Данные для справки и примеры

Аппроксимация (от лат. *approximare* — приближаться) — это математический метод, заключающийся в замене одних математических объектов (чисел, функций, графиков) другими, более простыми, известными или удобными для использования, но близкими к исходным в определённом смысле.

Цели аппроксимации

1. Упрощение вычислений. Замена сложной функции, например, тригонометрический ряд на простой полином, который легко дифференцировать или интегрировать.

2. Сжатие информации. Описание большого массива данных компактной формулой.

3. Восстановление зависимости. Найти закономерность там, где она неизвестна, например, по точкам на графике построить формулу.

Основные виды аппроксимации

Интерполяция. Жесткое условие, при котором аппроксимирующая функция $\varphi(x)$ обязана точно совпадать с исходными данными в заданных точках (x_k, y_k), называемых узлами интерполяции. Используется, когда исходные данные точны и их нужно сохранить.

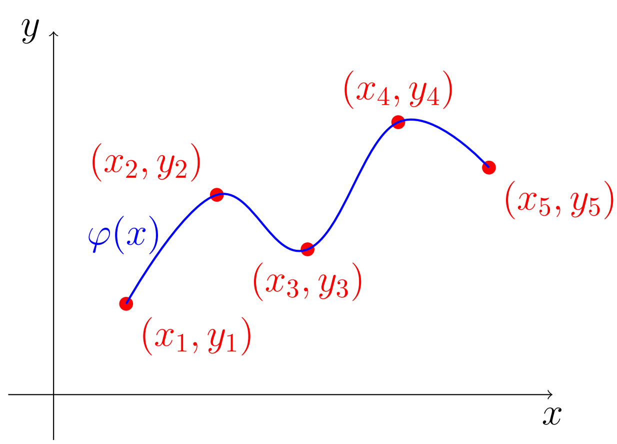

Рис. 1. Пример интерполяции. Гладкая кривая $\varphi(x)$ проходит через все точки данных

Экстраполяция. Процесс нахождения значений функции $\varphi(x)$ за пределами исходного интервала, на котором заданы данные. Результат сильно зависит от выбранной модели и может быть ненадежным при значительном удалении от известной области. Используется для прогнозирования, например, в экономике или физике.

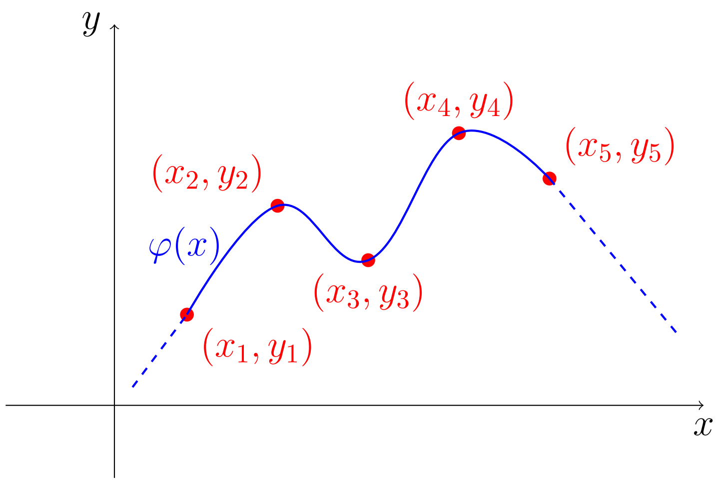

Рис. 2. Пример экстраполяции. Сплошная кривая $\varphi(x)$ проходит через точки данных; пунктир показывает продолжение функции за пределы области определения исходных данных.

Аппроксимация. Сглаживание методом наименьших квадратов. Функция $\varphi(x)$ не обязана проходить через точки. Она ищется так, чтобы сумма квадратов отклонений от этих точек была минимальной:

$$\sum_{i=1}^{n} \bigl(y_i - \varphi(x_i)\bigr)^2 \to \min .$$

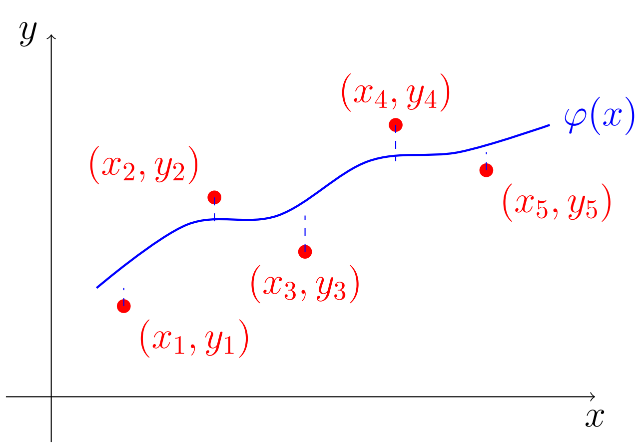

Рис. 3. Пример аппроксимации. Функция $\varphi(x)$ сглаживает точки данных, не проходя через них, и минимизирует сумму квадратов отклонений.

Функция `interp1`

Функция `interp1` выполняет одномерную интерполяцию, т.е. находит значения в произвольных точках x_q на основе заданных узлов (x_k, y_k).

Синтаксис:

```
yq = interp1(x, v, xq, method)
```

Параметры:

- `x` – вектор узлов (абсциссы). Должен быть монотонно возрастающим.
- `v` – вектор значений в узлах (ординаты).
- `xq` – точки, в которых требуется вычислить интерполяцию.
- `method` – метод интерполяции.

Таблица 1. Методы одномерной интерполяции `interp1`

Метод	**Описание**
`'linear'`	Кусочно-линейная интерполяция. Между узлами прямая линия. Простая, быстрая, но негладкая — разрыв производной в узлах.
`'spline'`	Кубический сплайн. Гладкая кривая, непрерывны первая и вторая производные. Может давать локальные выбросы при резких изменениях данных.
`'pchip'`	Кусочно-кубическая интерполяция Эрмита с сохранением монотонности. Не создаёт выбросов, гладкость ниже, чем у сплайна.
`'cubic'`	То же, что `'pchip'`, историческое название.
`'makima'`	Модифицированная интерполяция Акимы, доступна с R2019b. Компромисс между `'pchip'` (сохранение формы) и `'spline'` (гладкость).
`'nearest'`	Ближайший сосед. Значение равно значению в ближайшем узле. Ступенчатая функция.
`'next'`	Следующий сосед. Берётся значение из узла, следующего справа.
`'previous'`	Предыдущий сосед. Берётся значение из узла, слева от точки.
`'v5cubic'`	Устаревший кубический метод из MATLAB 5, не рекомендуется для новых проектов.

Примечание. Все методы требуют монотонно возрастающих узлов x. Экстраполяция — вычисление за пределами интервала узлов по умолчанию возвращает **`NaN`**. Для разрешения экстраполяции можно использовать аргумент `'extrap'` или указать скалярное значение.

Листинг 1. Пример. Интерполяция кубическая и сплайнами

```
% Данные
x = 0:4; %  вектор узлов. абсциссы
v = [1 2 1.5 3 2.5]; % вектор значений в узлах, ординаты
%  точки, в которых требуется вычислить интерполяцию
xq = linspace(0,4,100);

% Интерполяция
y_linear = interp1(x, v, xq, 'linear');
y_spline = interp1(x, v, xq, 'spline');
y_cubic = interp1(x, v, xq, 'cubic');
y_next = interp1(x, v, xq, 'next');

% Визуализация
figure(1)
plot(x, v, 'ko', xq,...
        y_spline, 'r--', xq, ...
        y_cubic, 'b-', 'LineWidth',2);
legend('Данные', 'Spline', 'Cubic');
```

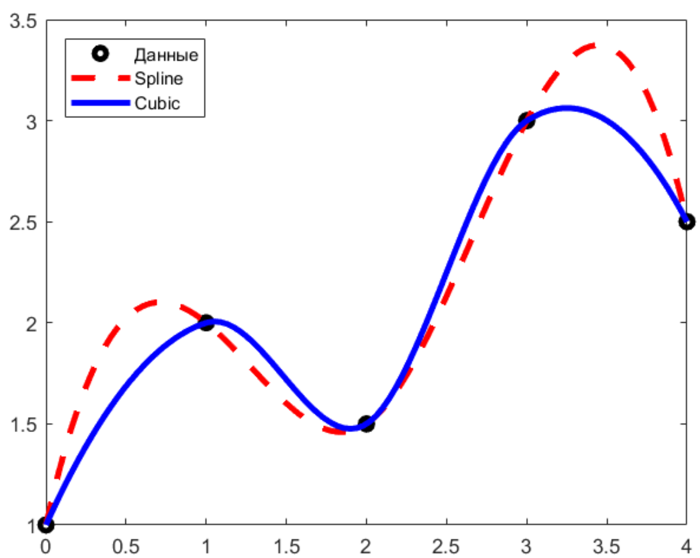

Рис. 4. Графические результаты для листинга 1

Функция `interp2`

Функция `interp2` выполняет двумерную интерполяцию на регулярной сетке, то есть находит значения в произвольных точках (x_q, y_q) на основе заданных узлов $(x_k, y_m, z_{k,m})$.

Синтаксис:

```
Zq = interp2(X, Y, Z, Xq, Yq, method)

% если X = 1:size(Z,2), Y = 1:size(Z,1)
Zq = interp2(Z, Xq, Yq, method)
```

Параметры:

- `X`, `Y` – векторы или матрицы, задающие координаты узлов сетки. Если заданы векторы, они должны быть монотонно возрастающими.
- `Z` – матрица значений размера `length(Y)` $\times$ `length(X)`.
- `Xq`, `Yq` – координаты точек, в которых требуется вычислить интерполяцию, могут быть скалярами, векторами или матрицами.
- `method` – метод интерполяции.

Таблица 2. Методы двумерной интерполяции `interp2`

Метод	**Описание**
`'linear'`	Кусочно-линейная интерполяция (билинейная). Значение в точке вычисляется по линейной интерполяции сначала по x, затем по y. Быстрая, негладкая — разрыв градиента.
`'spline'`	Интерполяция кубическим сплайном (бикубическая). Гладкая кривая, непрерывны первые и вторые частные производные. Может давать выбросы при резких изменениях данных.
`'cubic'`	Бикубическая интерполяция, аналогичная `'spline'`, но с несколько отличающейся реализацией, сохраняет форму.
`'makima'`	Модифицированная интерполяция Акимы в 2D, доступна с R2021b. Сочетает гладкость `'spline'` и сохранение формы `'pchip'`.
`'nearest'`	Ближайший сосед. Значение равно значению в ближайшем узле сетки. Даёт ступенчатую поверхность.
`'pchip'`	Кусочно-кубическая интерполяция Эрмита с сохранением монотонности. Не создаёт выбросов, гладкость ниже, чем у `'spline'`.

Примечание. Все методы требуют, чтобы сетка была прямоугольной и монотонной по каждому направлению. Для неравномерных сеток можно использовать `griddata` или `scatteredInterpolant`.

Листинг 2. Пример. Двумерная интерполяция вещественной функции

```
% Создаем исходную грубую сетку
[X, Y] = meshgrid(-2:0.5:2);

% Определяем вещественную функцию: F = X * exp(-X.^2 - Y.^2)
F = X .* exp(-X.^2 - Y.^2);

% Создаем более мелкую сетку для интерполяции
[Xq, Yq] = meshgrid(-2:0.1:2);

% Выполняем кубическую интерполяцию вещественного поля
Fq = interp2(X, Y, F, Xq, Yq, 'cubic');

% Визуализация исходных точек, грубая сетка
figure;
surf(X, Y, F, 'FaceAlpha', 0.5, 'EdgeColor', 'k', ...
              'Marker', 'o', 'MarkerFaceColor', 'r');
hold on;
% Визуализация интерполированной поверхности
surf(Xq, Yq, Fq, 'EdgeColor', 'none', 'FaceAlpha', 0.8);
title('Интерполяция вещественной функции кубический( метод)');
xlabel('X'); ylabel('Y'); zlabel('F(X,Y)');
legend('Исходные точки', 'Интерполированная поверхность');
colorbar;
```

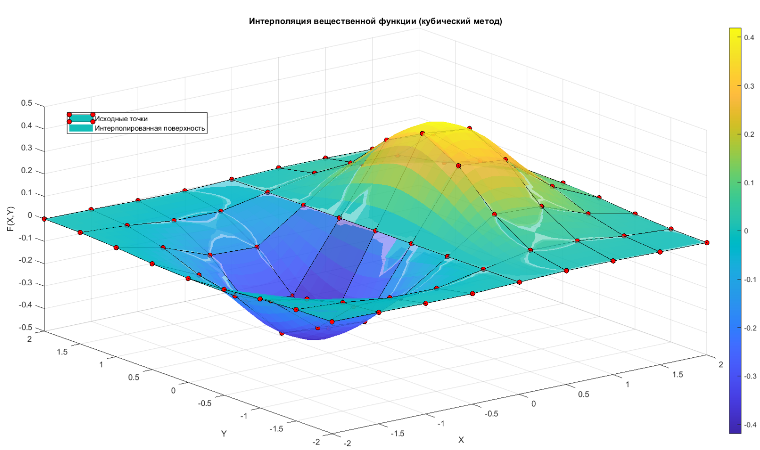

Рис. 5. Графические результаты для листинга 2

Функция `polyfit`

Функция `polyfit` выполняет аппроксимацию данных полиномом заданной степени по методу наименьших квадратов.

Синтаксис:

```
p = polyfit(x, y, n)
[p, S] = polyfit(x, y, n)
[p, S, mu] = polyfit(x, y, n)
```

Параметры:

- `x`, `y` — векторы с координатами точек данных.
- `n` — степень аппроксимирующего полинома.
- `p` — вектор коэффициентов полинома, упорядоченный от старшей степени к свободному члену.
- `S` — структура, содержащая информацию для оценки погрешности, поле `normr` — невязка, `df` — число степеней свободы.
- `mu` — вектор $[\mu_x, \sigma_x]$ для центрирования и масштабирования, применяется, если необходимо улучшить численную устойчивость.

```
x = 0:0.1:2;
y = sin(x) + 0.1*randn(size(x)); % зашумленные данные
p = polyfit(x, y, 3); % интерполяционный полином третьей степени
y_fit = polyval(p, x); % вычисление значений полинома
plot(x, y, 'o', x, y_fit, '-'); % визуализация
```

Примечания

- Для вычисления значений полинома в произвольных точках используется функция `polyval`.
- При использовании трёх выходных аргументов `[p, S, mu]` данные автоматически центрируются и масштабируются, что повышает устойчивость для высоких степеней.
- Для полиномов высокой степени рекомендуется использовать альтернативные методы, например, сплайны или кусочно-полиномиальную аппроксимацию.

Функция `fit`

Функция `fit` (требует наличия Curve Fitting Toolbox) позволяет строить аппроксимирующие модели различных типов: полиномиальные, экспоненциальные, сплайны и др.

Синтаксис:

```
fitobject = fit(x, y, fitType)
fitobject = fit([x, y], z, fitType)
fitobject = fit(x, y, fitType, fitOptions)
```

Параметры:

- `x`, `y`, `z` — векторы с данными для аппроксимации (для 2D и 3D).
- `fitType` — строка или объект `fittype`, определяющий модель.
- `fitOptions` — объект `fitoptions`, задающий дополнительные параметры: начальные приближения, веса, границы и т.д.
- `fitobject` — объект, содержащий коэффициенты модели, доверительные интервалы и методы для визуализации.

Таблица 3. Наиболее часто используемые типы моделей `fitType`

Тип модели	**Описание и пример задания**
`'poly1'` – `'poly9'`	Полином степени от 1 до 9.
`'smoothingspline'`	Сплайн с автоматическим подбором параметра сглаживания.
`'cubicinterp'`	Кубическая сплайн-интерполяция (проходит через все точки).
`'pchipinterp'`	Интерполяция Эрмита с сохранением формы.
`'exp1'`, `'exp2'`	Экспоненциальные модели: ae^{bx} и $ae^{bx} + ce^{dx}$.
`'power1'`, `'power2'`	Степенные модели: ax^b и $ax^b + c$.
`'fourier1'` – `'fourier8'`	Ряды Фурье (синусы и косинусы).
`'gauss1'` – `'gauss8'`	Суммы гауссианов.

Примечания

- Для сложных пользовательских моделей используется функция `fittype`, позволяющая задавать выражение в виде строки или анонимной функции.
- Объект `fitobject` содержит методы `plot`, `predict`, `coeffvalues`, `confint` и др.

Листинг 3. Пример. Аппроксимации полиномами

```
%% Исходные данные - зашумленная синусоида
x = (0:0.5:5)';
y = sin(x) + 0.2*randn(size(x));
% Аппроксимация полиномами
p1 = polyfit(x, y, 1); % Линейная аппроксимация
p2 = polyfit(x, y, 2); % Квадратичная
p3 = polyfit(x, y, 3); % Кубическая
% Создаем мелкую сетку для построения гладких графиков
x_fine = (0:0.05:5)';
% Вычисляем значения полиномов
y_fit1 = polyval(p1, x_fine);
y_fit2 = polyval(p2, x_fine);
y_fit3 = polyval(p3, x_fine);

% Визуализация
plot(x, y, 'ko', 'MarkerFaceColor', 'k','MarkerSize', 8); hold on;
plot(x_fine, y_fit1, 'r--', 'LineWidth', 3.5);
plot(x_fine, y_fit2, 'g-.', 'LineWidth', 3.5);
plot(x_fine, y_fit3, 'b-', 'LineWidth', 3.5);
legend('Данные', 'Линейная', 'Квадратичная', 'Кубическая');
title('Полиномиальная аппроксимация');
xlabel('x'); ylabel('y');
grid on;
hold off;
```

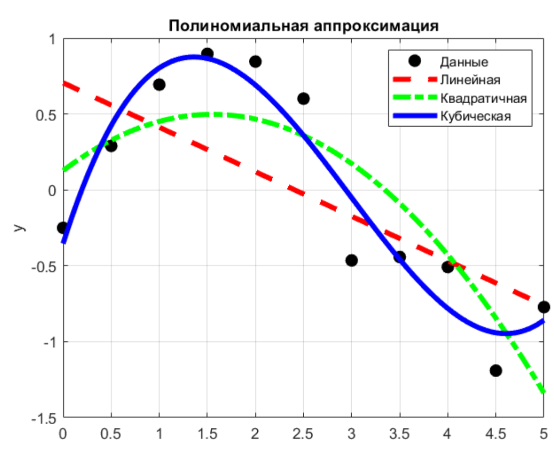

Рис. 6. Графические результаты для листинга 3

Листинг 4. Пример. Аппроксимация экспоненциальной моделью

```
% Данные
x = (0:0.5:10)';
y = 3 * exp(−0.5 * x) + 0.5*randn(size(x));

% Аппроксимация
f = fit(x, y, 'exp1');
disp(f);

% Тонкая сетка для гладкой кривой
x_fine = linspace(min(x), max(x), 100);
y_fit = f(x_fine);

% Построение
figure;
plot(x, y, 'ko', 'MarkerSize', 12, 'MarkerFaceColor', 'k');
hold on;
plot(x_fine, y_fit, 'r−', 'LineWidth', 4);
hold off;

legend('Данные', 'Экспоненциальная аппроксимация');
title('Аппроксимация экспонентой');
xlabel('x'); ylabel('y');
grid on;
```

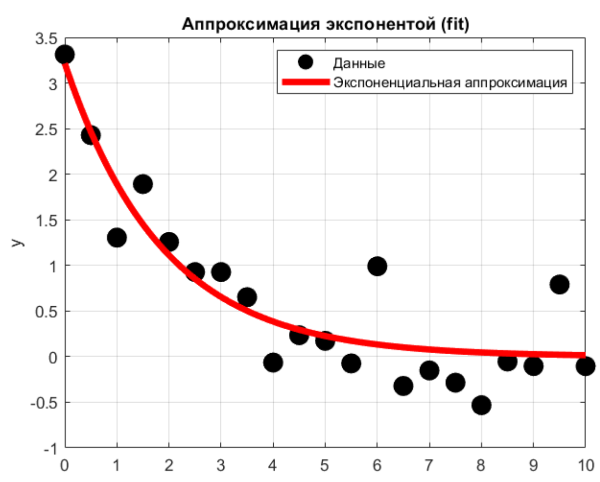

Рис. 7. Графические результаты для листинга 4

Листинг 5. Пример. Аппроксимация пользовательской моделью

```
% Данные
x = (0:0.5:10)';
y = 3 * exp(-0.5 * x) + 0.5*randn(size(x));

% Аппроксимация
my_model = fittype('a*exp(-b*x)*sin(c*x)');
f = fit(x, y, my_model);

% Тонкая сетка для гладкой кривой
x_fine = linspace(min(x), max(x), 100);
y_fit = f(x_fine);

% Построение
figure;
plot(x, y, 'ko', 'MarkerSize', 12, 'MarkerFaceColor', 'k');
hold on;
plot(x_fine, y_fit, 'r-', 'LineWidth', 4);
hold off;

legend('Данные', 'Экспоненциальная аппроксимация');
title('Аппроксимация пользовательской моделью');
xlabel('x'); ylabel('y');
grid on;
```

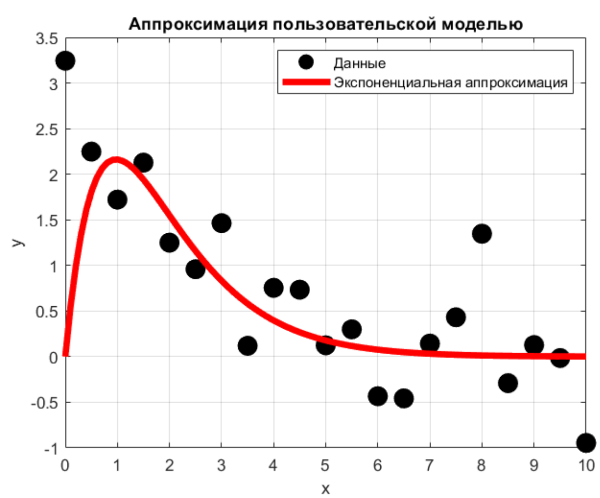

Рис. 8. Графические результаты для листинга 5

Задание 1. Интерполяция

Выполнить интерполяцию функции (данные в таблице 4), заданной на равномерной сетке в указанном диапазоне с заданным количеством узлов, различными методами:

- `'nearest'` — ближайший сосед;
- `'next'` — следующее значение;
- `'previous'` — предыдущее значение;
- `'pchip'` — кусочно-кубическая интерполяция Эрмита;
- `'cubic'` — кубическая интерполяция (требует равномерной сетки);
- `'spline'` — сплайн-интерполяция.

Построить графики восстановленных данных для каждого метода на одной координатной плоскости, сравнить с исходной функцией.

Таблица 4. Варианты к заданию 1 (интерполяция)

№	Функция	Диапазон	Кол-во точек	№	Функция	Диапазон	Кол-во точек
1	$\sin x$	$[0, 2\pi]$	9	**26**	$\cos x$	$[0, \pi]$	11
2	e^{-x}	$[0, 3]$	7	**27**	x^2	$[-2, 2]$	10
3	$\sqrt{x}$	$[0, 4]$	6	**28**	$1/(1+x^2)$	$[-5, 5]$	15
4	$\|x\|$	$[-3, 3]$	13	**29**	$1/(1+x^4)$	$[-10, 10]$	12
5	$\tanh x$	$[-3, 3]$	12	**30**	$\ln(1+x)$	$[0, 5]$	8
6	$x^3 - x$	$[-2, 2]$	9	**31**	$x^3 - 3x$	$[-3, 3]$	13
7	$\sin(x^2)$	$[0, \sqrt{2\pi}]$	10	**32**	e^{-x^2}	$[-3, 3]$	12
8	$1/x$	$[1, 10]$	10	**33**	$x \sin x$	$[0, 4\pi]$	10
9	$x \cos x$	$[0, 4\pi]$	14	**34**	$\sin(x)/x$	$[-4\pi, 4\pi]$	9
10	e^x	$[0, 2]$	6	**35**	xe^{-x}	$[0, 5]$	10
11	$\sinh x$	$[-2, 2]$	9	**36**	$\cosh x$	$[-2, 2]$	9
12	$\arctan x$	$[-5, 5]$	15	**37**	$1/(1+x^2)$	$[-2, 2]$	10
13	$\sin(2x)$	$[0, 10]$	15	**38**	$\cos(2x)$	$[0.5, 10]$	15
14	$\ln x$	$[1, 5]$	9	**39**	$x \ln x$	$[1, 5]$	9
15	$x \sin(1/x)$	$[0.1, 1]$	10	**40**	$\|x\| \sin x$	$[-2\pi, 2\pi]$	8
16	$\mathrm{sign}(x)\, x^2$	$[-2, 2]$	9	**41**	$1/(1+e^{-x})$	$[-5, 5]$	12
17	$\sin^3 x$	$[0, 2\pi]$	10	**42**	$\cos^3 x$	$[0, 2\pi]$	10
18	$\sqrt{1-x^2}$	$[-1, 1]$	8	**43**	$x\sqrt{1-x^2}$	$[-1, 1]$	8
19	$\ln(1+x^2)$	$[-5, 5]$	15	**44**	$\sin(x)/(1+x^2)$	$[-5, 5]$	15
20	$\tan x$	$[-\pi/3, \pi/3]$	10	**45**	$\cot x$	$[\pi/6, \pi/2]$	8
21	$\sec x$	$[-\pi/3, \pi/3]$	10	**46**	$\csc x$	$[\pi/6, \pi/2]$	12
22	$x^3 + x$	$[-3, 3]$	12	**47**	$1/(1+(x-2)^2)$	$[1, 4]$	8
23	$x/(1-x^2)$	$[-0.9, 0.9]$	10	**48**	$1/(1+x^2)$	$[-2, 2]$	10
24	x^3	$[-0.5, 0.5]$	8	**49**	$1/x$	$[0.5, 1.5]$	8
25	$\sin(x)/(1+x)$	$[0, 10]$	15	**50**	$\cos(x)/(1+x)$	$[0.5, 10]$	15

Задание 2. Двумерная интерполяция

Выполнить двумерную интерполяцию функции (данные в таблице 5), заданной на равномерной прямоугольной сетке в указанных диапазонах с заданным количеством узлов по каждому направлению, различными методами:

- `'nearest'` — ближайший сосед;
- `'linear'` — билинейная интерполяция;
- `'cubic'` — бикубическая интерполяция;
- `'spline'` — сплайн-интерполяция.

Для каждого метода построить график поверхности (например, с помощью `mesh` или `surf`), сравнить с исходной функцией.

Таблица 5. Варианты к заданию 2 (двумерная интерполяция)

№	Функция $f(x,y)$	Диапазон x и y	Сетка	№	Функция $f(x,y)$	Диапазон x и y	Сетка
1	$\sin x \cos y$	$[-2\pi, 2\pi]$	9×9	**26**	$xe^{-x^2-y^2}$	$[-2, 2]$	9×9
2	$e^{-x^2-y^2}$	$[-3, 3]$	11×11	**27**	$\cos(x+y)$	$[-\pi, \pi]$	9×9
3	$x^2 - y^2$	$[-2, 2]$	9×9	**28**	$1/(1+x^2+y^2)$	$[-5, 5]$	15×15
4	$\sin(x)\sin(y)$	$[-\pi, \pi]$	10×10	**29**	$\sqrt{\lvert xy\rvert}$	$[0, 4]$	9×9
5	xy	$[-3, 3]$	9×9	**30**	$\cos(x)e^{-y}$	$[0, 4]$	10×10
6	$\sin(x^2+y^2)$	$[-\pi, \pi]$	11×11	**31**	$\arctan(x+y)$	$[-5, 5]$	13×13
7	$e^{-x}\cos y$	$[0, 4]$	9×9	**32**	x^2+y^2	$[-2, 2]$	9×9
8	$x \sin y$	$[-3, 3]$	10×10	**33**	$\sin(x)/(1+y^2)$	$[-\pi, \pi]$	12×12
9	$\ln(1+x^2+y^2)$	$[-5, 5]$	13×13	**34**	$\tanh(x)\cosh(y)$	$[-3, 3]$	9×9
10	$\cos(x^2)\sin(y)$	$[-\pi, \pi]$	10×10	**35**	$x^3 - 3xy^2$	$[-2, 2]$	9×9
11	$\mathrm{sinc}(x)\,\mathrm{sinc}(y)$	$[-10, 10]$	15×15	**36**	$e^{-x^2}\cos y$	$[-3, 3]$	11×11
12	$\lvert x\rvert + \lvert y\rvert$	$[-3, 3]$	9×9	**37**	$1/(y^2+(x+1)^2)$	$[-5, 5]$	12×12
13	$\sin(x)\cosh(y)$	$[-\pi, \pi]$	10×10	**38**	$xy^2 - x^2y$	$[-2, 2]$	9×9
14	$\exp(-x^2)\sin(y)$	$[-4, 4]$	9×9	**39**	$\sin(x)e^{-y^2}$	$[-\pi, \pi]$	10×10
15	$\ln(1+\lvert x\rvert)\cos y$	$[-5, 5]$	11×11	**40**	$\sqrt{x^2+y^2}$	$[0, 5]$	10×10
16	$\sin(xy)$	$[-2, 2]$	9×9	**41**	$1/(1+e^{-x-y})$	$[-5, 5]$	13×13
17	$\cos(x)\cos(y)$	$[-\pi, \pi]$	9×9	**42**	$x^2 \sin y$	$[-3, 3]$	10×10
18	$\tanh(x+y)$	$[-4, 4]$	11×11	**43**	$\mathrm{sign}(x)y^2$	$[-2, 2]$	9×9
19	$\sqrt{1-x^2-y^2}$	$[-0.9, 0.9]$	8×8	**44**	$\sin^2 x \cos^2 y$	$[-\pi, \pi]$	10×10
20	xe^{-y}	$[-2, 2]$	9×9	**45**	$\cos(x^2)e^{-y}$	$[0, 3]$	9×9
21	$\mathrm{erf}(x)\cos y$	$[-2, 2]$	10×10	**46**	$1/(1+x^2)\sin y$	$[-5, 5]$	11×11
22	$x^2y - y^3$	$[-3, 3]$	9×9	**47**	$\cosh(x)\sinh(y)$	$[-2, 2]$	9×9
23	$\ln(1+x^2)\cos y$	$[-5, 5]$	11×11	**48**	$\sin(x)/(1+y^2)$	$[-\pi, \pi]$	10×10
24	$\exp(-x^2)\exp(-y^2)$	$[-3, 3]$	9×9	**49**	$\cos(x)\ln(1+y^2)$	$[-5, 5]$	11×11
25	$\sin(x)\cos(y)e^{-xy}$	$[-2, 2]$	9×9	**50**	$\sqrt{\lvert x\rvert}\sin y$	$[0, 4]$	9×9

Задание 3. Аппроксимация данных

Выполнить аппроксимацию функции (данные в таблице 6), заданной на равномерной сетке в указанном диапазоне с заданным количеством узлов, следующими методами:

- экспоненциальная аппроксимация `exp1`;
- полиномиальная аппроксимация 3-й степени;
- аппроксимация пользовательской функцией.

Для каждого метода построить график исходных точек и аппроксимирующей кривой, сравнить результаты и сделать выводы о точности и применимости методов для данной функции.

Таблица 6. Варианты к заданию 3 (аппроксимация)

№	Функция $f(x)$	Диапазон x	Кол-во точек	№	Функция $f(x)$	Диапазон x	Кол-во точек
1	$\sin x$	$[0, 2\pi]$	9	**26**	$\ln x$	$[1, 5]$	9
2	$\cos x$	$[0, \pi]$	11	**27**	$x \ln x$	$[1, 5]$	9
3	e^{-x}	$[0, 3]$	7	**28**	$x \sin(1/x)$	$[0.1, 1]$	10
4	x^2	$[-2, 2]$	10	**29**	$\|x\| \sin x$	$[-2\pi, 2\pi]$	20
5	$\sqrt{x}$	$[0, 4]$	6	**30**	$\operatorname{sign}(x)\, x^2$	$[-2, 2]$	9
6	$1/(1 + x^2)$	$[-5, 5]$	15	**31**	$1/(1 + e^{-x})$	$[-5, 5]$	12
7	$\|x\|$	$[-3, 3]$	13	**32**	$\sin^3 x$	$[0, 2\pi]$	10
8	$1/(1 + x^4)$	$[-10, 10]$	20	**33**	$\cos^3 x$	$[0, 2\pi]$	10
9	$\tanh x$	$[-3, 3]$	12	**34**	$\sqrt{1 - x^2}$	$[-1, 1]$	8
10	$\ln(1 + x)$	$[0, 5]$	8	**35**	$x\sqrt{1 - x^2}$	$[-1, 1]$	8
11	$x^3 - x$	$[-2, 2]$	9	**36**	$\ln(1 + x^2)$	$[-5, 5]$	15
12	$x^3 - 3x$	$[-3, 3]$	13	**37**	$\sin(x)/(1 + x^2)$	$[-5, 5]$	15
13	$\sin(x^2)$	$[0, \sqrt{2\pi}]$	10	**38**	$\tan x$	$[-\pi/3, \pi/3]$	10
14	e^{-x^2}	$[-3, 3]$	15	**39**	$\cot x$	$[\pi/6, \pi/2]$	8
15	$1/x$	$[1, 10]$	10	**40**	$\sec x$	$[-\pi/3, \pi/3]$	10
16	$x \sin x$	$[0, 4\pi]$	20	**41**	$\csc x$	$[\pi/6, \pi/2]$	8
17	$x \cos x$	$[0, 4\pi]$	20	**42**	$x^3 + x$	$[-3, 3]$	12
18	$\sin(x)/x$	$[-4\pi, 4\pi]$	25	**43**	$1/(1 + (x - 2)^2)$	$[1, 4]$	8
19	e^x	$[0, 2]$	6	**44**	$x/(1 - x^2)$	$[-0.9, 0.9]$	10
20	xe^{-x}	$[0, 5]$	10	**45**	$1/(1 + x^2)$	$[-2, 2]$	10
21	$\sinh x$	$[-2, 2]$	9	**46**	x^3	$[-0.5, 0.5]$	8
22	$\cosh x$	$[-2, 2]$	9	**47**	$1/x$	$[0.5, 1.5]$	8
23	$\arctan x$	$[-5, 5]$	15	**48**	$\sin(x)/(1 + x)$	$[0, 10]$	15
24	$\sin(2x)$	$[0, 10]$	15	**49**	$\cos(x)/(1 + x)$	$[0.5, 10]$	15
25	$\cos(2x)$	$[0.5, 10]$	15	**50**	$\sqrt{1 + x^2}$	$[-3, 3]$	11

Задание 4. Экстраполяция

Для вариантов из таблиц 4 и 5 выполните экстраполяцию, увеличив диапазон изменения аргументов функций на 5% от заданного значения.

1 Вопросы

1. Дайте определение интерполяции, аппроксимации и экстраполяции.
2. В чём принципиальное различие между понятиями интерполяции и аппроксимации?
3. В каких случаях целесообразно применять интерполяцию?
4. В каких случаях целесообразно применять аппроксимацию?
5. В чём отличие интерполяции от экстраполяции?
6. Почему экстраполяция считается менее надёжной?
7. Какая функция в MATLAB отвечает за одномерную интерполяцию? Опишите её основные аргументы.
8. В чём отличие методов `'cubic'` и `'spline'`? Какой метод даёт более гладкую кривую?
9. Для каких целей используется метод `'pchip'`? Каковы его особенности?
10. Поддерживает ли функция `interp1` интерполяцию комплексных чисел? Если да, то как это реализовано?
11. Почему метод `'pchip'` не работает с комплексными данными?
12. Как выполнить двумерную интерполяцию в MATLAB? Какие функции для этого существуют?
13. Какая функция используется для полиномиальной аппроксимации методом наименьших квадратов?
14. Как вычислить значения полученного полинома в заданных точках?
15. Что возвращает функция `polyfit` помимо коэффициентов полинома? Для чего нужны дополнительные выходные аргументы?
16. Объясните, как правильно выбрать степень аппроксимирующего полинома. Какие признаки указывают на неправильный выбор степени?
17. Как выполнить экстраполяцию с помощью функции `interp1`?