

# Численными методы решения инженерных задач

***Численные методы являются одним из мощных математических средств решения инженерных задач.***

Современные численные методы и мощные ЭВМ дали возможность решать такие задачи, о которых полвека назад могли только мечтать. Но применять численные методы далеко не просто. Цифровые ЭВМ умеют выполнять только арифметические действия и логические операции. Поэтому помимо разработки математической модели, требуется еще разработка алгоритма, сводящего все вычисления к последовательности арифметических и логических действий.

Выбирать модель и алгоритм надо с учетом скорости и объема памяти ЭВМ: чересчур сложная модель может оказаться машине не под силу, а слишком простая – не даст физической точности.

Сам алгоритм и программа для ЭВМ должны быть тщательно проверены. Даже проверка программы нелегка, о чем свидетельствует популярное утверждение: «В любой сколь угодно малой программе есть по меньшей мере одна ошибка».

Однако численные методы не всемогущи. Они не отменяют все остальные математические методы. Начиная исследовать проблему, целесообразно использовать простейшие модели, аналитические методы и прикидки.

И только разобравшись в основных чертах явления, надо переходить к полной модели и сложным численным методам; даже в этом случае численные методы выгодно применять в комбинации с точными и приближенными аналитическими методами.

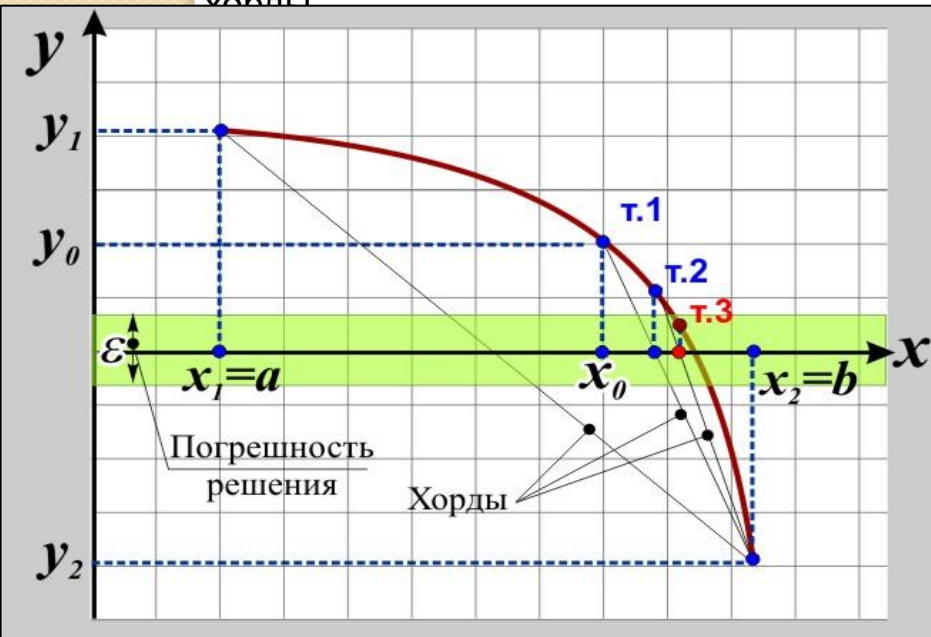
## Численные методы нахождения корней уравнения $y(x) = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1$

В инженерной практике часто возникает задача нахождения значения аргумента функции (аппроксимированной или нет) при котором эта функция равна определенному значению. Частный случай решения этой задачи – **это нахождение корней уравнения.**

### **Метод хорд**

- 1) Выбираем  $x_1 = a$  и  $x_2 = b$ , для которых функция имеет разные знаки (например,  $y(a) > 0$  и  $y(b) < 0$ );
- 2) Вычисляем значение функции в этих точках  $y_1 = y(x_1)$  и  $y_2 = y(x_2)$ ;
- 3) Между этими двумя точками  $(y_1, x_1)$  и  $(y_2, x_2)$  провести прямую – **хорду**. Уравнение этой хорды

$$y(x) = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1$$



котором хорда  $y(x_0) = 0$

в точке  $x_0$ ;

$\neq 0$ , то  $x_2 = x_0$ .

1.6 пока значение функции в этой точке  $y(x_0)$  не  
ости)  $\varepsilon$ .

**ней, или корней несколько, то время поиска решения  
к бесконечности.**

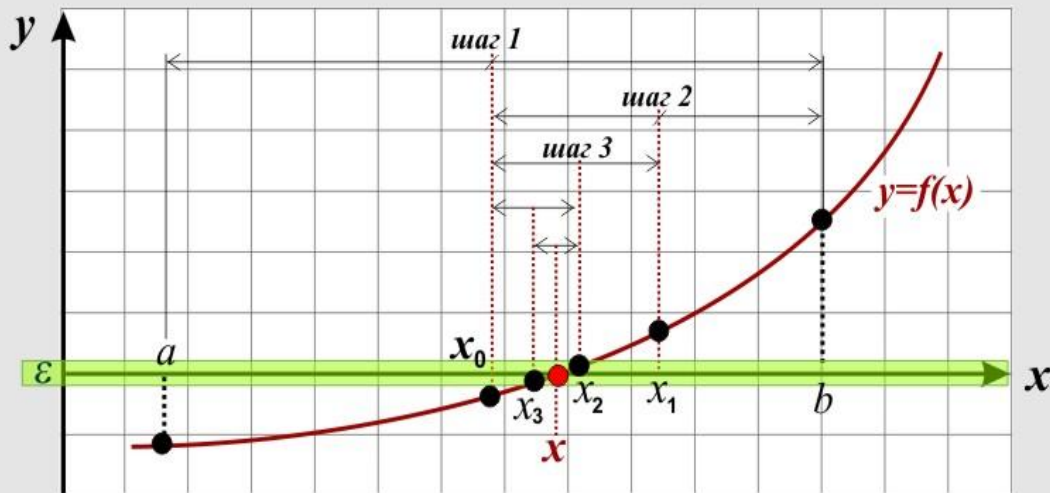
## Численные методы нахождения корней уравнения

### **Метод дихотомии (деление пополам)**

1) Пусть мы нашли такие точки  $a$  и  $b$  что  $f(a) \cdot f(b) < 0$ , т.е. на отрезке  $[a, b]$  лежит не менее одного корня 2) Найдем середину отрезка  $x_0 = (a+b)/2$  и вычислим значение функции на середине этого отрезка  $f(x_0)$  3) Из двух получившихся половинок отрезка выберем ту, для которой  $f(x_0) \cdot f(a)$  или  $f(b) \leq 0$ , т.е. отрезок на котором функция меняет знак.

4) новый отрезок опять делим пополам и выберем ту половину, на концах которой функция имеет разные знаки, и т. д.

Если требуется найти корень с точностью  $\varepsilon$ , то продолжаем деление пополам до тех пор, пока значение функции в средней точке не будет меньше  $\varepsilon$ . **Тогда середина последнего отрезка даст значение корня с требуемой точностью.**



**ых непрерывных функций, в том числе**

отором функция меняет знак. Если в этом к какому из них сойдется процесс (хотя к

надежность счета, а скорость сходимости

# Численные методы нахождения корней уравнения

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)};$$

## Метод касательных (метод Ньютона), (метод линеаризации)

1) Дана функция  $f(x)$  и отрезок  $[a, b]$ . Построится касательная

к функции в точке  $x_1$  – это точка пересечения, полученной касательной с

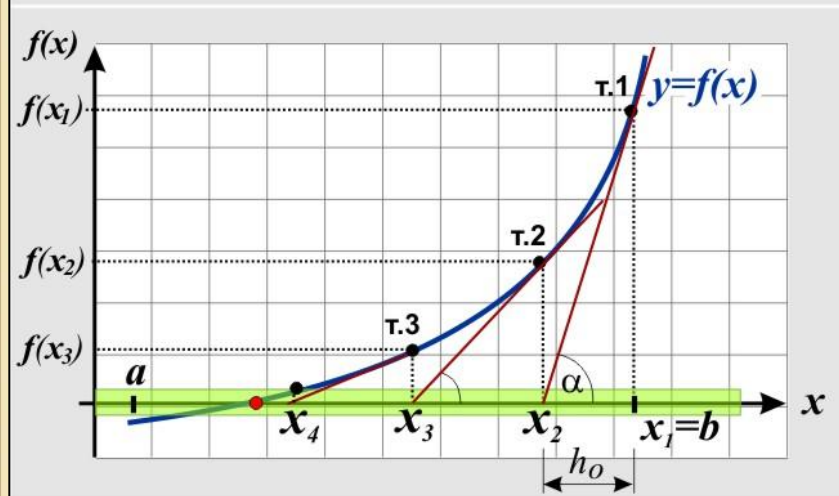
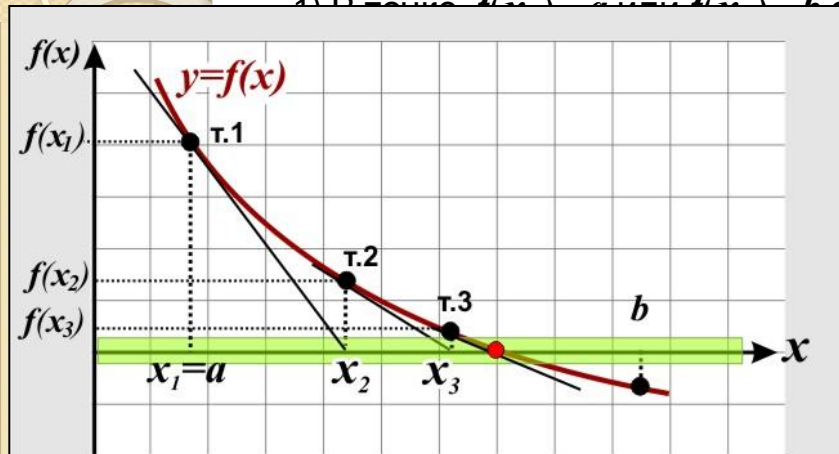
оси  $x$  – до достижения заданной точности  $\varepsilon$

получить итерационную формулу метода, используя формулу Ньютона. Если  $f(x)$  имеет непрерывную производную, тогда

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)};$$

$x_3, \dots$  Таким образом, можно записать общую формулу:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)};$$



## Численные методы нахождения корней уравнения

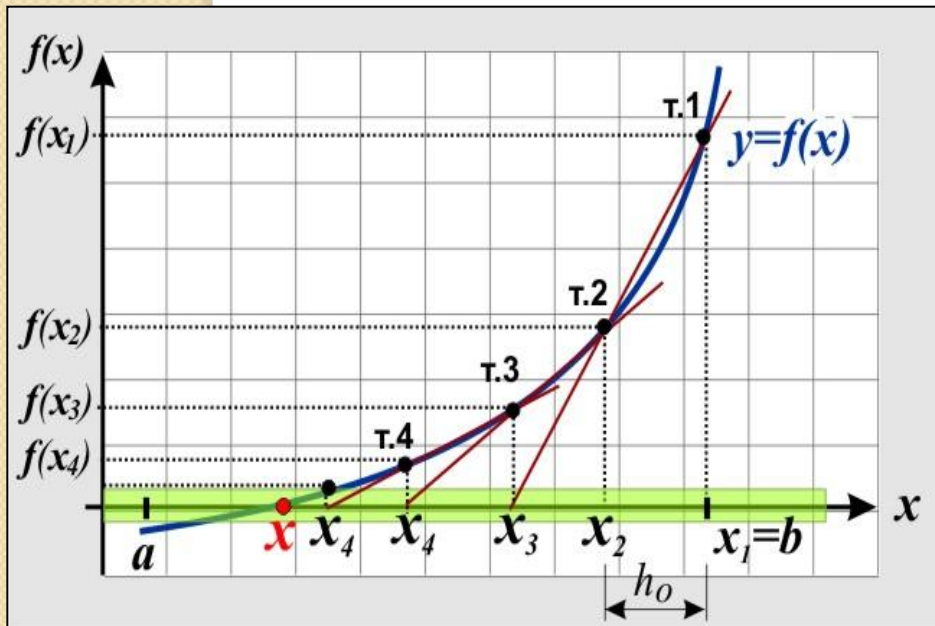
$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})f(x_n)}{f(x_n) - f(x_{n-1})}$$

### Метод секущих

В данном методе, в отличие от метода Ньютона, проводятся не касательные, а секущие. Из рисунка легко получить итерационную формулу:

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})f(x_n)}{f(x_n) - f(x_{n-1})};$$

- 1) задаться первоначальными значениями  $x_1$  и  $x_2$
- 2) вычислить  $x_3$  по формуле, представленной выше
- 3) вычислить значение функции  $f(x_3)$  в этой точке, сравнить с заданной погрешностью  $\varepsilon$
- 5) если  $f(x_3) > \varepsilon$  заменить  $x_2$  на  $x_3$ , если  $f(x_3) \leq \varepsilon$  прекратить расчет
- 6) повторить действия с п.2 по п.3, пока заданная точность не будет достигнута.



необходимо задать не только  $x_1$ , но и  $x_2$ . Метод сед методом Ньютона – здесь не нужно вычислять

**важные недостатки.** Сходимость итераций может быть в этой окрестности корня. В знаменателе формулы стоит  $f(x_n) - f(x_{n-1})$ , что не существенно, но **вблизи корня значения** **делительная проблема деления на очень малые числа или** **цифр, т.е. точности, или, как говорят к «разболтке» счёта.**

## Численные методы нахождения корней уравнения

### *Решение уравнений, функции `fzero` и `roots`*

Для уточнения корня в MATLAB предназначена функция **fzero**. В самом простом варианте существует два способа обращения к ней:

- с указанием начального приближения к корню  $x_0$   $x = fzero(fun, x_0)$
- с указанием отрезка  $[a, b]$ , на котором отделен корень  $x = fzero(fun, [a, b])$

Если работа алгоритма **fzero** закончится успешно, то в выходном аргументе  $x$  возвращается найденное приближенное к корню значение.

По умолчанию, корень находится с точностью, близкой к машинной, т.е. порядка  $10^{-16}$ .

Входной аргумент **fun** задает функцию. Он может быть строкой с выражением для функции, заключенной в апострофы, **inline- функцией** или **m-функцией**.

Перед вызовом **fzero** полезно установить формат **long e** для того, чтобы видеть все значащие цифры. Для этого выполняем команду

```
format long e
```

(или в меню **File** выбираем пункт **Preferences**, в левой части окна **Preferences** переходим к разделу **Command Window**, а в правой части в списке **Numeric format** выбираем **long e**).

## Численные методы нахождения корней уравнения

### Решение уравнений, функции `fzero` и `roots`

Рассмотрим простой пример. Требуется решить уравнение  $\cos(x) - x = 0$

1. Для этого сначала отделяем корень.

В нашем примере он будет лежать на отрезке  $[0, \pi/2]$ , т.к. корень — абсцисса точки пересечения графиков функций  $g(x)=\cos(x)$  и  $h(x)=x$ . На границах этого отрезка функция принимает значения разных знаков и это существенно для начала работы `fzero`.

2. Определим inline - функцию

```
fun = inline('cos(x)-x');
```

3. Вызовем `fzero` для нахождения корня

```
x = fzero(fun, [0 pi/2])
```

```
x = 7.390851332151607e-001
```

Для проверки того, что корень найден с точностью  $10^{-16}$  можно воспользоваться простым, но эффективным способом, а именно вычислить значения функции в точках  $(x + 10^{-16})$  и  $(x - 10^{-16})$ . Если знаки функции в этих точках различны, то точный корень находится на расстоянии не больше  $10^{-16}$  от найденного значения  $x$ .

```
fun(x-1e-16)
```

```
ans = 1.110223024625157e-016
```

```
fun(x+1e-16)
```

```
ans = -2.220446049250313e-016
```

Проверка показывает, что знаки разные, следовательно корень найден с точностью  $10^{-16}$

## Численные методы вычисления определённых интегралов

$$J = \int_a^b f(x) dx$$

В общем виде определенный интеграл записывается в виде:

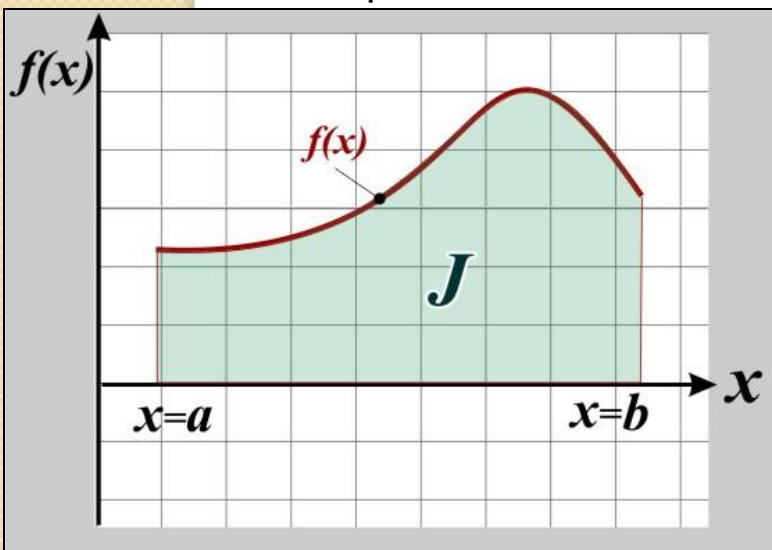
$$J = \int_a^b f(x) dx$$

Вычислить значение интеграла – определить площадь фигуры, ограниченной отрезками прямых  $x=a$  и  $x=b$ , осью абсцисс, и графиком подынтегральной функции  $f(x)$

Для некоторых подынтегральных функций  $f(x)$  интеграл можно вычислить аналитически, найти в справочниках или оценить с помощью асимптотических рядов.

**Однако в общем случае первообразная  $F(x)$  может быть не определена: либо первообразные не выражаются через элементарные функции, либо сами подынтегральные функции не являются элементарными.**

Это приводит к необходимости разработки приближенных методов вычисления



тельными приближенными методами вычисления интегралов являются так называемые «классические» **интегрирования по квадратурным формулам: метод трапеций, метод парабол** основанные на **разбиении площади, на которые разбивается вся**



# Численные методы вычисления определённых интегралов

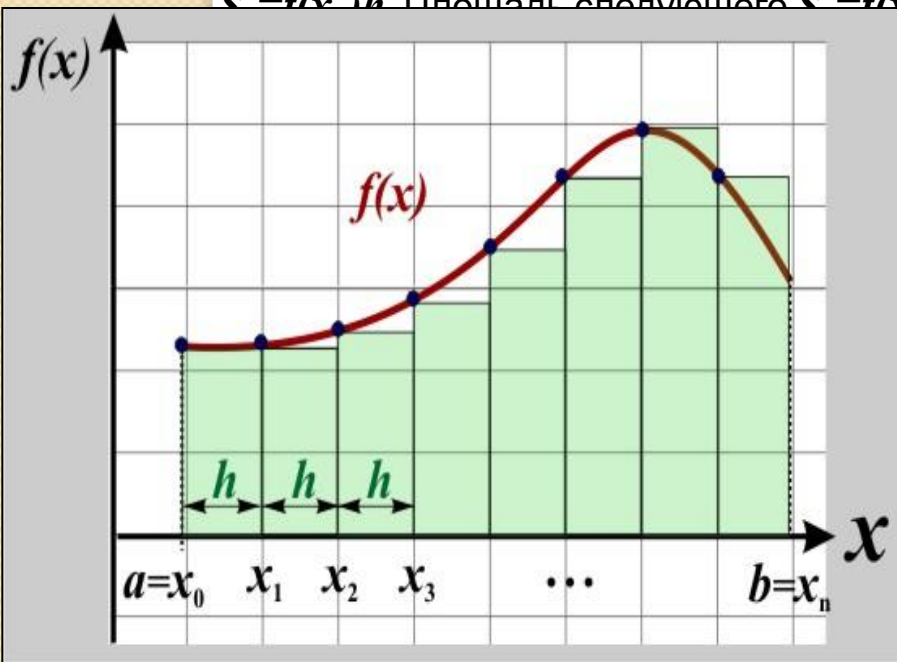
$$J = h \cdot \sum_{i=0}^{n-1} f(x_i)$$

## Метод левых прямоугольников

Для вычисления интеграла отрезок  $[a, b]$  разбивают на  $n$  отрезков длиной  $h$  (шаг интегрирования). На каждом из этих отрезков криволинейную трапецию приближают прямоугольником, так как его площадь можно легко вычислить. Затем суммируют все полученные площади, получая тем самым приближённое значение интеграла.

Название объясняется тем, что высота прямоугольника  $f(x)$  вычисляется в левой границе отрезка  $h$ .

Как видно из рисунка площадь первого слева прямоугольника равна произведению  $S = f(x_0) \cdot h$ . Площадь следующего  $S = f(x_1) \cdot h$ .



прямоугольника  $S_i = f(x_i) \cdot h$ . Всего таких прямоугольников  $n$ ,  
 Таким образом, приближённое значение интеграла,  
 по формуле:

$$J = h \cdot \sum_{i=0}^{n-1} f(x_i)$$

определяется как половина площади одного  
 ть меньше заданной погрешности. Это достигается  
 ольшого числа точек  $n$  на отрезке интегрирования.

# Численные методы вычисления определённых интегралов $J = h \cdot \sum_{i=0}^{n-1} f(x_{i+1})$

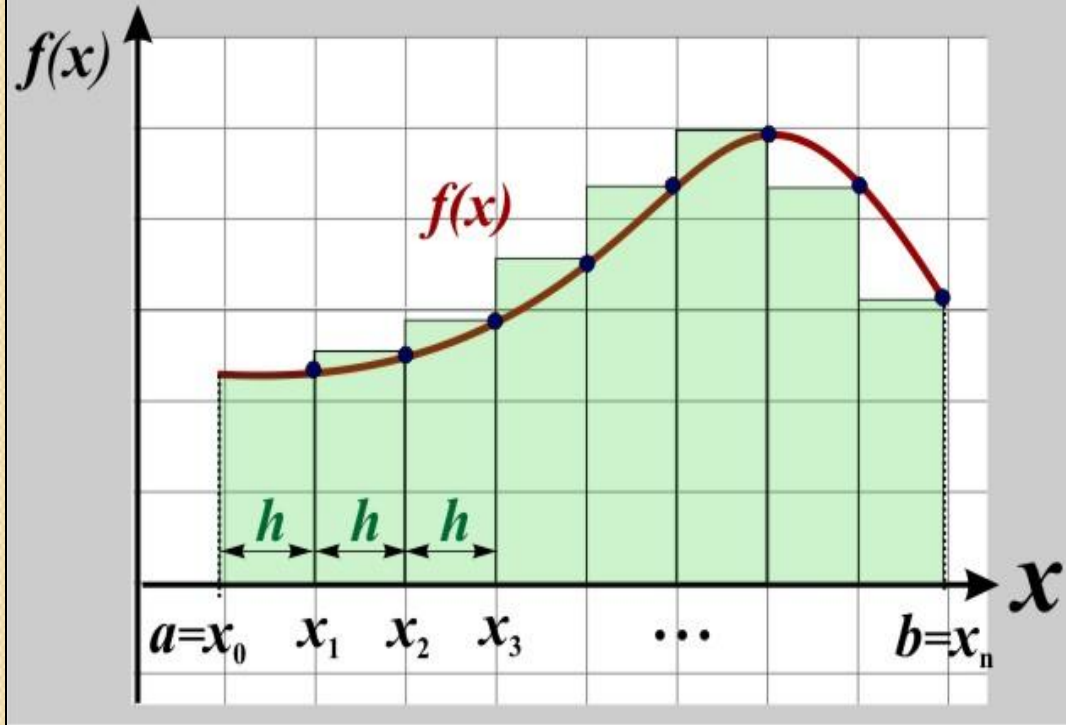
## Метод правых прямоугольников

По аналогии с методом левых прямоугольников существует и метод правых прямоугольников. Отличие в том, что высота прямоугольников вычисляется по правой границе.

Выводы формул для данного метода аналогичны предыдущему.

Основные отличия заключаются в нумерации. Формула метода правых прямоугольников выглядит следующим образом:

$$J = h \cdot \sum_{i=0}^{n-1} f(x_{i+1})$$



Связанные с определением погрешности для метода правых прямоугольников, аналогичны тому, что

# Численные методы вычисления определённых интегралов

$$h \leq \sqrt{\frac{24\varepsilon}{(b-a) \max_{[a,b]} |f''(x)|}}$$

## Метод средних прямоугольников

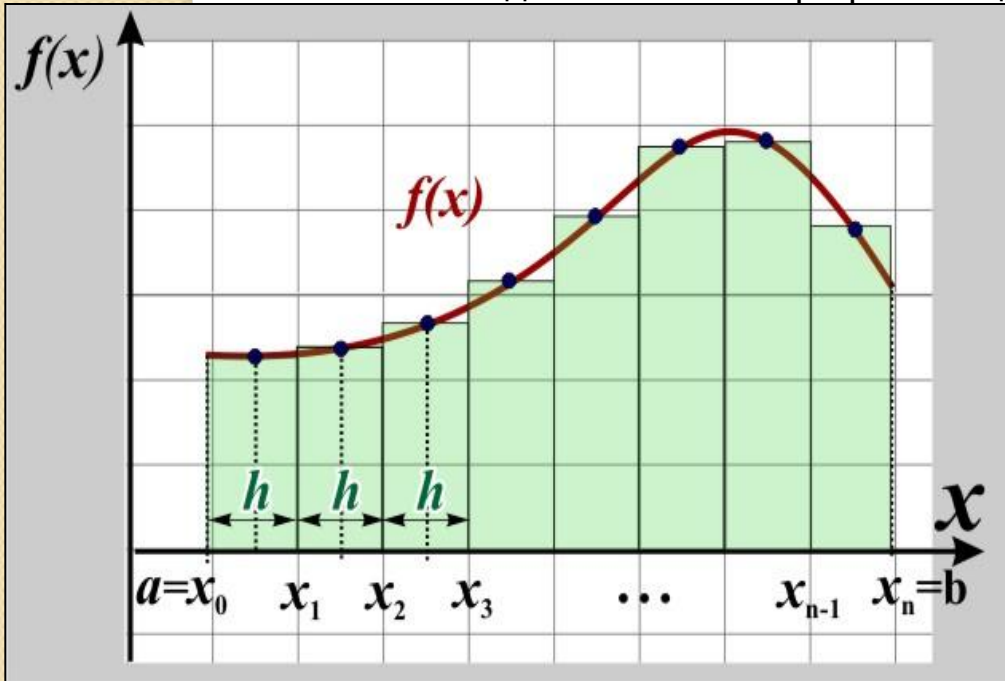
Чтобы уменьшить погрешность методов левых и правых прямоугольников был предложен метод средних, т.е. метод в котором высота прямоугольника вычисляется в середине отрезка  $h$ .

Площади прямоугольников вычисляются по следующей формуле:

$$J = h \cdot \sum_{i=1}^{n-1} f\left(x_i + \frac{h}{2}\right)$$

Оценим погрешность формулы на отрезке  $[x_i, x_i + h]$ .

Вычислим необходимый шаг интегрирования для получения заданной точности по формуле:



$$h \leq \sqrt{\frac{24\varepsilon}{(b-a) \max_{[a,b]} |f''(x)|}}$$

функции интеграл которой необходимо найти. (левых, средних прямоугольников) приходим к **во много раз ниже погрешности метода** метод средних во много раз точнее и для экономии машинного времени.

# Численные методы вычисления определённых интегралов $h \leq \sqrt{\frac{12\varepsilon}{(b-a)\max_{[a,b]}|f''(x)|}}$

## Метод трапеций

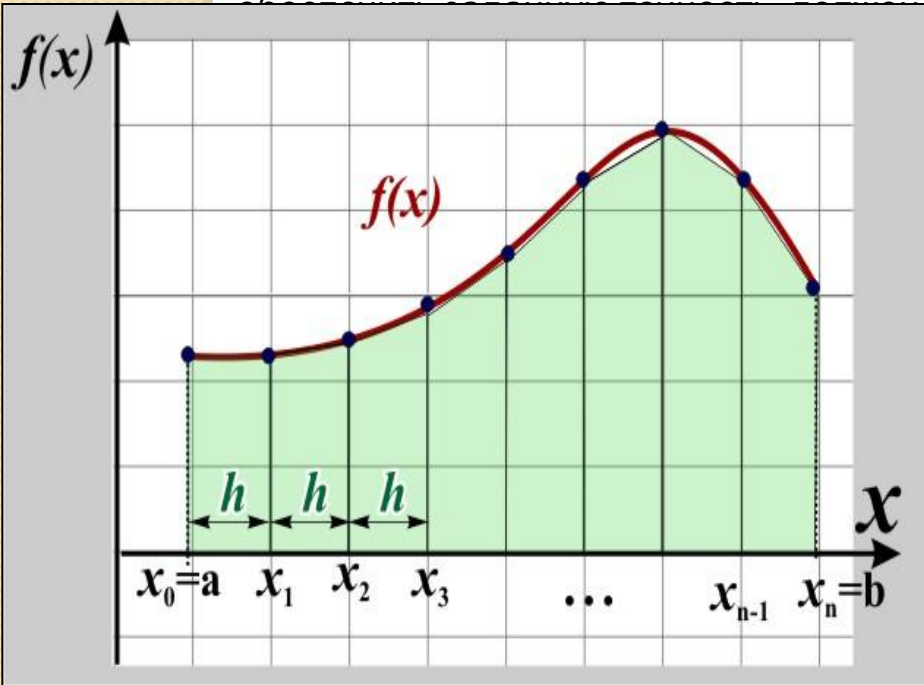
Метод трапеций основан на том, что криволинейная трапеция приближается прямолинейной. Т.е. площадь элементарного прямоугольника вычисляются по следующей формуле:

$$S_i = h \frac{f(x_i) + f(x_{i-1})}{2}$$

Общая формула метода трапеций

$$J_{mp.} = \frac{h}{2} \cdot \sum_{i=1}^n f(x_i) + f(x_{i-1})$$

Оценим погрешность метода. Для кусочно-непрерывной функции шаг интегрирования, чтобы быть меньше величины:



$$h \leq \sqrt{\frac{12\varepsilon}{\max_{[a,b]}|f''(x)|}}$$

Погрешность метода (трапеций) примерно вдвое меньше величины  $\sqrt{\frac{12\varepsilon}{\max_{[a,b]}|f''(x)|}}$  для функций определённых только в узлах

на всём интервале, лучше пользоваться методом

трапеций для функций определённых только в узлах

## Численные методы вычисления определённых интегралов

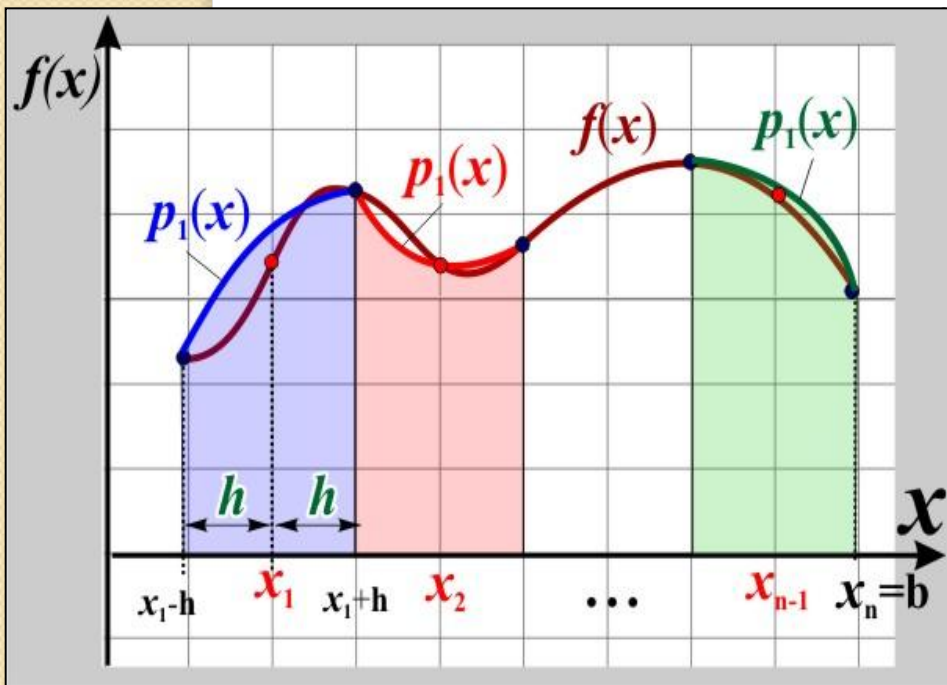
### **Метод Симпсона (метод парабол)**

Более высокую точность расчетов обеспечивает использование параболической (квадратичной) аппроксимации по трем соседним точкам отрезка.

Уравнение полинома второй степени, проходящего через три точки:  $(x_0, y_0)$ ;  $(x_1, y_1)$ ;  $(x_2, y_2)$  можно записать в виде (аппроксимация методом Лагранжа)

$$y(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

$$y(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$



приближается на отрезке  $[x_i - h, x_i + h]$  параболой

вид  $p(x) = ax^2 + bx + c$ , т.е. чтобы определить коэффициенты, необходимо решить систему из трёх уравнений, а для этого достаточно знать три точки, через которые проходит парабола  $p$

**Вместо использования формул Симпсона, для вычисления площади трапеции понадобится не две, а**

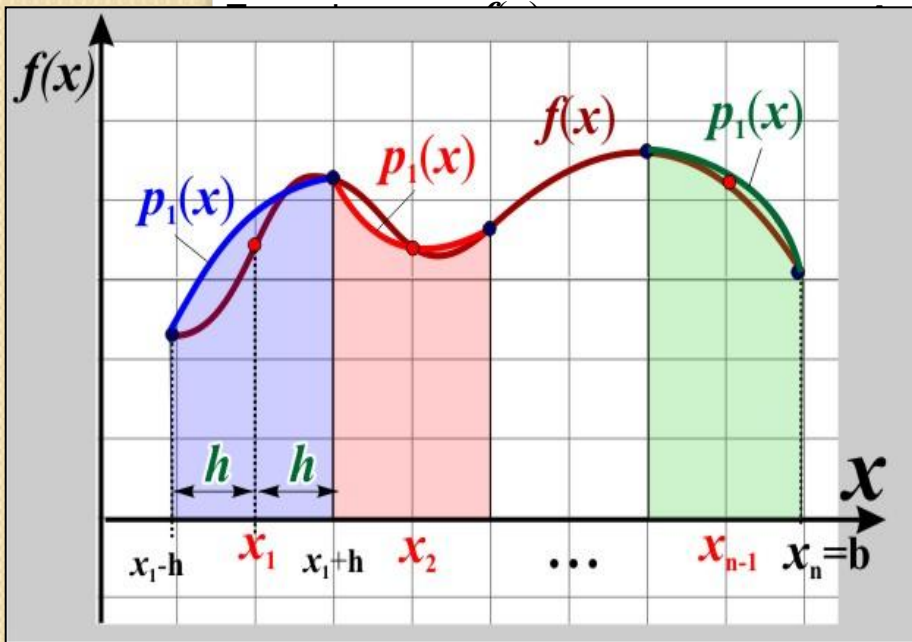
# Численные методы вычисления определённых интегралов

## Метод Симпсона (метод парабол)

Окончательная формула определения интеграла.

$$J = \frac{h}{3} (f(a) + f(b) + 4 \sum_{i=1}^n f(x_i) + 2 \sum_{i=1}^{n-1} f(x_i + h))$$

В этой формуле  $a$  и  $b$  – границы отрезка интегрирования,  $h$  – шаг интегрирования,  $n$  – количество отрезков длиной  $2h$ .



а) непрерывную производную четвертого порядка, то погрешность формулы Симпсона не больше чем:

$$\Delta = M \frac{(b-a)^5}{2880n^4}$$

где  $M$  – максимум по абсолютной величине четвертой производной  $f^{(4)}(x)$ , функции  $f(x)$  на отрезке  $[a, b]$ .  
 то погрешность формулы Симпсона с ростом  $n$  уменьшается быстрее, чем погрешность формулы трапеций.

## Численные методы вычисления определённых интегралов

### **Метод Симпсона (метод парабол)**

Для вычисления интегралов методом трапеций в системе MATLAB предусмотрена функция **trapz**:

$$\text{Integ} = \text{trapz}(x,y);$$

Одномерный массив  $x$  (вектор) содержит дискретные значения аргументов подынтегральной функции. Значения подынтегральной функции в этих точках сосредоточены в одномерном массиве  $y$ .

Чаще всего для интегрирования выбирают равномерную сетку, то есть значения элементов массива  $x$  отстоят друг от друга на одну и ту же величину (**шаг интегрирования**). Точность вычисления интеграла зависит от величины шага интегрирования: чем меньше этот шаг, тем больше точность.

$$I = \int_0^{\pi} \cos(x) dx$$

Вычислим простой интеграл методом трапеций с шагом интегрирования  $\pi/10$ .

```
x = 0:pi/10:pi;
```

```
y=cos(x);
```

```
J1 = trapz(x,y)
```

```
J1 = 1.387778780781446e-16
```

В системе Matlab методы интегрирования более высоких порядков точности реализуются функцией: **quad** (метод квадратур)

Как и многие другие функции системы MATLAB, функция **quad** может принимать различное количество параметров. Полный формат вызова этой функции

```
Q = quad('Fun',a,b,Tol,Trace)
```

## Численные методы решения дифференциальных уравнений

### Основные понятия и определения

Дифференциальным уравнением называется уравнение, связывающее независимую переменную  $x$ ,

искомую функцию  $y=f(x)$  и её производные  $y', y'', \dots, y^{(n)}$ , т. е. уравнение вида

$$F(x, y, y', y'', \dots, y^{(n)}) = 0$$

Если искомая функция  $y=f(x)$  есть функция одной независимой переменной  $x$ , дифференциальное уравнение называется **обыкновенным**

$$dy/dx + xy = 0 \quad y'' + y' + x = \cos x \quad (x^2 - y^2)dx - (x + y)dy = 0$$

Когда искомая функция  $y$  есть функция двух и более независимых переменных, например, если  $y=y(x, t)$ , то уравнение вида

$$F(x, t, y, dy/dx, dy/dt, \dots, d^m y/dx^k \partial t^l) = 0$$

называется **уравнением в частных производных**. Здесь  $k, l$  — неотрицательные целые числа, такие, что  $k + l = m$ ; например

$$(dy/dt - dy/dx) = 0, \quad (dy/dt) = (d^2 y/dx^2)$$



# Численные методы решения дифференциальных уравнений

## Основные понятия и определения

**Порядком дифференциального уравнения** называется порядок **наивысшей производной**, входящей в уравнение.

Например:

$y' + xy = e^x$  - уравнение первого порядка;

$y'' + p(x)y = 0$ , где  $p(x)$  - известная функция, - уравнение второго порядка;

$y^{(9)} - xy'' = x^2$  - уравнение 9-го порядка.

**Решением дифференциального уравнения  $n$ -го порядка** на интервале  $(a, b)$  называется функция  $y = \phi(x)$ , определенная на интервале  $(a, b)$  вместе со своими производными до  $n$ -го порядка включительно, и такая, что подстановка функции  $y = \phi(x)$  в дифференциальное уравнение превращает его в тождество по  $x$  на  $(a, b)$

Например, функция  $y = \sin x + \cos x$  является решением уравнения  $y'' + y = 0$  на интервале  $(-\infty, +\infty)$ .

В самом деле, дифференцируя функцию дважды, будем иметь

$$y' = \cos x - \sin x, \quad y'' = -\sin x - \cos x$$

Подставляя выражения  $y''$  и  $y$  в дифференциальное уравнение, получим тождество

$$-\sin x - \cos x + \sin x + \cos x \equiv 0$$

График решения дифференциального уравнения называется **интегральной кривой** этого уравнения.

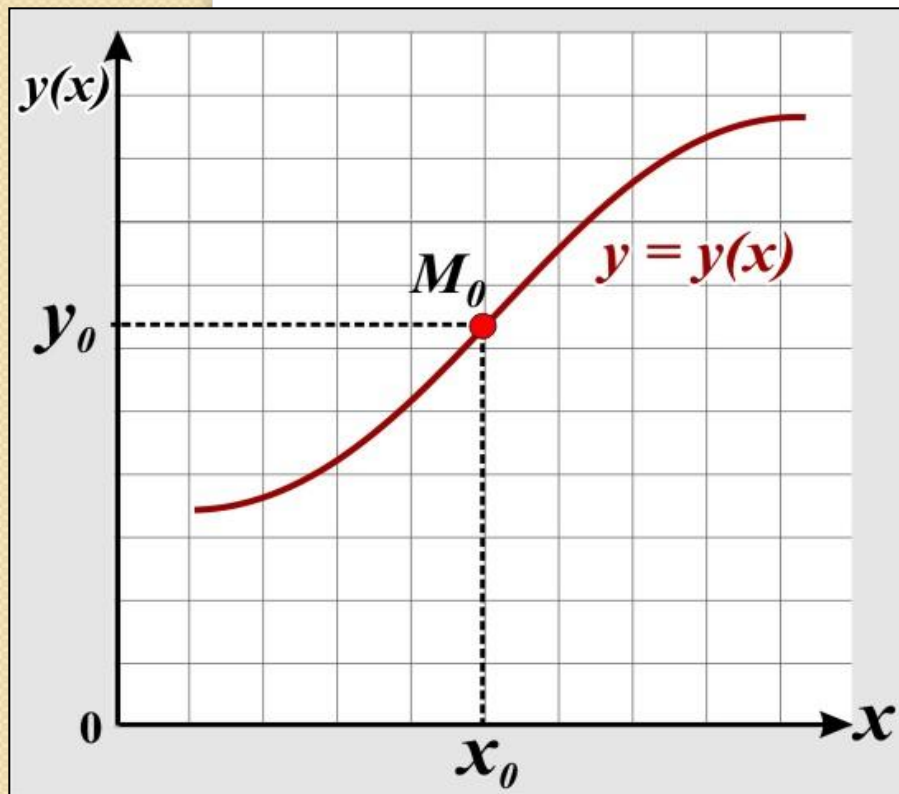
# Численные методы решения дифференциальных уравнений

## Основные понятия и определения

Общий вид уравнения первого порядка  $F(x, y, y') = 0$

Если уравнение  $F(x, y, y') = 0$  удастся разрешить относительно  $y'$ , то получится уравнение первого порядка, разрешенное относительно производной.

$$y' = f(x, y)$$



задачу нахождения решения  $y = y(x)$   
соответствующего начально-му условию  $y(x_0)$

получается интегральная кривая, проходящая через  
точку  $M_0$  в  $xOy$

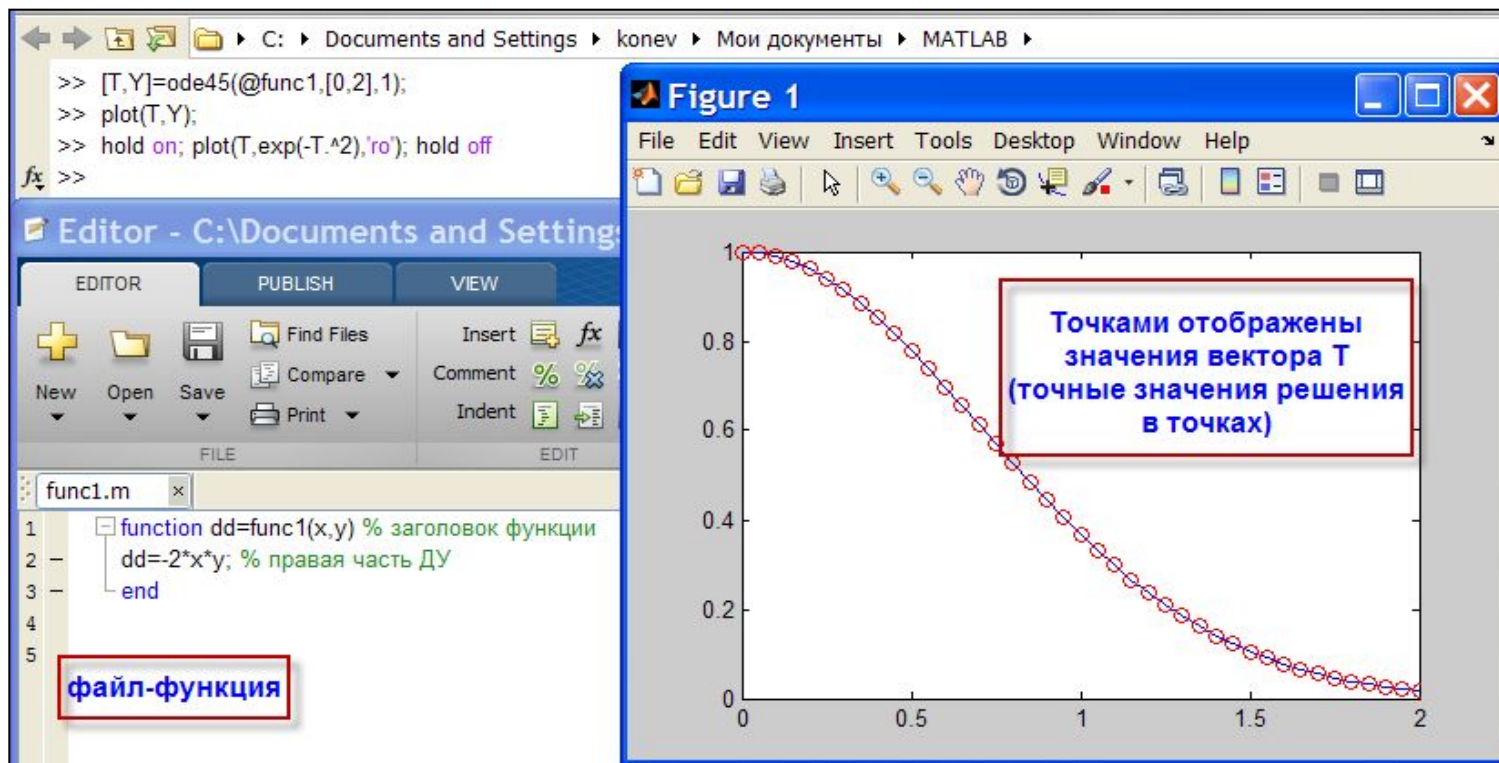
# Численные методы решения дифференциальных уравнений

## Решение дифференциальных уравнений в Matlab

В качестве самого простого примера приведем решение следующего уравнения  $dy/xy dx = -2xy$  с начальным условием  $y(0)=1$  и аналитическим решением  $y(x)=e^{-x^2}$

**Возможный формат вызова процедуры решателя в MatLab:**

`[T,Y]=ode45(@DiffEquatFunc,[Tstart,Tfinal],StartVector)`



## Численные методы решения систем дифференциальных уравнений

*Все методы решения системы дифференциальных уравнений (ДУ) можно грубо разделить на:*

- явные, которые используют для нахождения приближенной будущей точки только прошлые значения;*
- неявные, которые используют пробные шаги вперед*

Большинство численных методов решения начинается с представления исходных ДУ в форме Коши: *слева только одна производная в одном уравнении, справа – только переменные (координаты) функции и аргументы.*

Формулировка задачи Коши – *требуется решить с заданной точностью  $\varepsilon$  на отрезке  $[x_0, x_0 + b]$  уравнение вида:  $y' = F(x, y)$  при  $y(x_0) = y_0$*

Наиболее популярным методом ее численного решения ДУ считается обычный метод Рунге-Кутты, который дает высокую степень наглядности, и обладает такими положительными чертами, как возможность применения переменного шага, и достаточно малой погрешностью получаемого решения (порядка  $h^5$ , где  $h$  – шаг).

*Эти обстоятельства делают данный метод приемлемым для инженерных расчетов.*