

Тема 8

Решение обыкновенных дифференциальных уравнений в Maple

1. Аналитические решения обыкновенных дифференциальных уравнений.
 - 1.1. Общее решение.
 - 1.2. Фундаментальная система решений.
 - 1.3. Решение задачи Коши.
 - 1.4. Решение систем дифференциальных уравнений.
 - 1.5. Проверка найденного решения.
 - 1.6. Интегральные преобразования для решения дифференциальных уравнений.
2. Приближенные решения обыкновенных дифференциальных уравнений.
3. Численные решения обыкновенных дифференциальных уравнений, построение графиков численных решений.
4. Графическое представление решений дифференциальных уравнений, построение фазовых портретов
 - 4.1. Двумерные графики решений ДУ или системы ДУ
 - 4.2. Трехмерные графики решений ДУ или системы ДУ
 - 4.3. Построение фазового портрета.
 - 4.4. Изображение поля направлений.

Всюду далее примеры работы пакета Maple будут приведены в режиме интерфейса Worksheet Mode с текстовым режимом ввода команд и выражений Text Mode или математическим режимом ввода Math Mode.

§1. Аналитическое решение дифференциальных уравнений

1.1 Общее решение дифференциальных уравнений.

Для нахождения аналитических решений дифференциальных уравнений в Maple применяется команда **dsolve(eq, var, options)**, где **eq** – дифференциальное уравнение, **var** – неизвестные функции, **options** – параметры. Параметры могут указывать метод решения задачи, например, по умолчанию ищется аналитическое решение: **type=exact**. По умолчанию Maple старается найти явное выражение для неизвестной функции. При невозможности выделить искомую функцию решение выводится в неявном виде. В этом случае для вывода решения в явном виде следует указать опцию **type=explicit** либо задать **_EnvExplicit:=true** перед выполнением команды **dsolve**

При составлении дифференциальных уравнений для обозначения производной применяется команда **diff**, например, дифференциальное уравнение $y''+y=x$ записывается в виде: **diff(y(x), x\$2)+y(x)=x**.

В режиме ввода Math Mode можно также использовать стандартную нотацию со штрихами, например:

> $y' + y \cdot \cos(x) = \sin(x) \cdot \cos(x)$

Общее решение дифференциального уравнения зависит от произвольных постоянных, число которых равно порядку дифференциального уравнения. В Maple такие постоянные, как правило, обозначаются как **_C1**, **_C2**, и т.д.

Общее решение неоднородного линейного дифференциального уравнения всегда выводится так, чтобы была четко видна, структура этого решения. Как известно, общее решение неоднородного линейного дифференциального уравнения равно сумме общего

решения соответствующего однородного дифференциального уравнения и частного решения этого же неоднородного дифференциального уравнения. Поэтому в строке вывода решение неоднородного линейного дифференциального уравнения всегда состоит из слагаемых, которые содержат произвольные постоянные (это общее решения соответствующего однородного дифференциального уравнения), и слагаемых без произвольных постоянных (это частное решения этого же неоднородного дифференциального уравнения).

Команда **dsolve** выдает решение дифференциального уравнения в невычисляемом формате. Для того, чтобы с решением можно было бы работать далее (например, построить график решения) следует отделить правую часть полученного решения командой **rhs (%)**.

Задание 1.1.

1.1.1. Найти общее решение дифференциального уравнения $y'+y\cos x=\sin x\cos x$.

> **restart;**

> **de:=diff(y(x),x)+y(x)*cos(x)=sin(x)*cos(x);**

$$de := \frac{d}{dx} y(x) + y(x) \cos(x) = \sin(x) \cos(x)$$

> **dsolve(de,y(x));**

$$y(x) = \sin(x) - 1 + e^{-\sin(x)} _C1$$

Итак, решение искомого уравнения есть функция $y(x) = \sin(x) - 1 + e^{-\sin(x)} _C1$

Замечание: при записи решения дифференциального уравнения в *Maple* в строке вывода произвольная постоянная обозначена как $_C1$.

1.1.2. Найти общее решение дифференциального уравнения второго порядка $y''-2y'+y=\sin x+e^{-x}$.

> **restart;**

> **deq:=diff(y(x),x\$2)-2*diff(y(x),x)+y(x)=sin(x)+exp(-x);**

$$deq := \frac{d^2}{dx^2} y(x) - 2 \left(\frac{d}{dx} y(x) \right) + y(x) = \sin(x) + e^{-x}$$

> **dsolve(deq,y(x));**

$$y(x) = e^x _C2 + e^x x _C1 + \frac{1}{4} e^{-x} (2 \cos(x) e^x + 1)$$

Замечание: так как исходное уравнение было второго порядка, то полученное решение содержит две произвольные константы, которые в *Maple* обычно обозначаются как $_C1$ и $_C2$. Первые два слагаемых представляют собой общее решение соответствующего однородного дифференциального уравнения, а последнее слагаемое – частное решение неоднородного дифференциального уравнения.

1.1.3. Найти общее решение дифференциального уравнения второго порядка $y''+k^2y=\sin(qx)$ в двух случаях: $q \neq k$ и $q=k$ (резонанс).

> **restart; de:=diff(y(x),x\$2)+k^2*y(x)=sin(q*x);**

$$de := \frac{d^2}{dx^2} y(x) + k^2 y(x) = \sin(q x)$$

> **dsolve(de,y(x));**

$$y(x) = \sin(k x) _C2 + \cos(k x) _C1 + \frac{\sin(q x)}{k^2 - q^2}$$

Теперь найдем решение в случае резонанса. Для этого перед вызовом команды **dsolve** следует приравнять $q=k$.

> **q:=k: dsolve(de, y(x));**

$$y(x) = \sin(kx) _C2 + \cos(kx) _C1 - \frac{1}{2} \frac{\cos(kx)x}{k}$$

Замечание: в обоих случаях частное решение неоднородного уравнения и общее решение, содержащее произвольные постоянные, выводятся отдельными слагаемыми.

1.2 Фундаментальная (базисная) система решений.

Команда **dsolve** представляет возможность найти фундаментальную систему решений (базисные функции) дифференциального уравнения. Для этого в параметрах команды **dsolve** следует указать **output=basis**.

Задание 1.2.

Найти фундаментальную систему решений дифференциального уравнения: $y^{(4)}+2y''+y=0$.

> **de:=diff(y(x), x\$4)+2*diff(y(x), x\$2)+y(x)=0;**

$$de := \frac{d^4}{dx^4} y(x) + 2 \left(\frac{d^2}{dx^2} y(x) \right) + y(x) = 0$$

> **dsolve(de, y(x), output=basis);**

[sin(x), cos(x), sin(x) x, cos(x) x]

В режиме ввода Math Mode:

> **dsolve(y''''+2*y''+y=0, y);**

$$y(x) = _C1 \sin(x) + _C2 \cos(x) + _C3 \sin(x) x + _C4 \cos(x) x$$

> **dsolve(y''''+2*y''+y=0, y, output=basis);**

[sin(x), cos(x), sin(x) x, cos(x) x]

1.3 Решение задачи Коши или краевой задачи

Команда **dsolve** может найти решение задачи Коши или краевой задачи, если помимо дифференциального уравнения задать начальные или краевые условия для неизвестной функции. В режиме ввода Text Mode для обозначения производных в начальных или краевых условиях используется дифференциальный оператор D , например, условие $y''(0)=2$ следует записать в виде $(D@@2)(y)(0)=2$, или условие $y'(1)=0$: $D(y)(1)=0$. Напомним, что производная n -го порядка записывается в виде $(D@@n)(y)$.

В режиме ввода Math Mode (2D-Math) можно использовать стандартную нотацию со штрихами (см. пример ниже).

Задание 1.3.

1.3.1. Найти решение задачи Коши: $y^{(4)}+y''=2\cos x$, $y(0)=-2$, $y'(0)=1$, $y''(0)=0$, $y'''(0)=0$.

Первый способ (Math Mode):

> **dsys := {y''''+y''=2*cos(x), y(0)=-2, y'(0)=1, y''(0)=0, y'''(0)=0};**

> **dsolve(dsys, y(x));**

$$y(x) = -2 \cos(x) - \sin(x) x + x$$

Второй способ (Text Mode):

```
> de:=diff(y(x),x$4)+diff(y(x),x$2)=2*cos(x);
```

$$de := \frac{d^4}{dx^4} y(x) + \frac{d^2}{dx^2} y(x) = 2 \cos(x)$$

```
> cond:=y(0)=-2, D(y)(0)=1, (D@@2)(y)(0)=0, (D@@3)(y)(0)=0;
```

$$cond := y(0) = -2, D(y)(0) = 1, \\ D^{(2)}(y)(0) = 0, D^{(3)}(y)(0) = 0$$

```
> dsolve({de,cond},y(x));
```

$$y(x) = -2 \cos(x) - \sin(x) x + x$$

1.3.2. Найти решение краевой задачи: $y''+y=2x-\pi$, $y(0)=0$, $y\left(\frac{\pi}{2}\right)=0$. Построить график решения.

```
> restart; de:=diff(y(x),x$2)+y(x)=2*x-Pi;
```

$$de := \frac{d^2}{dx^2} y(x) + y(x) = 2x - \pi$$

```
> cond:=y(0)=0,y(Pi/2)=0;
```

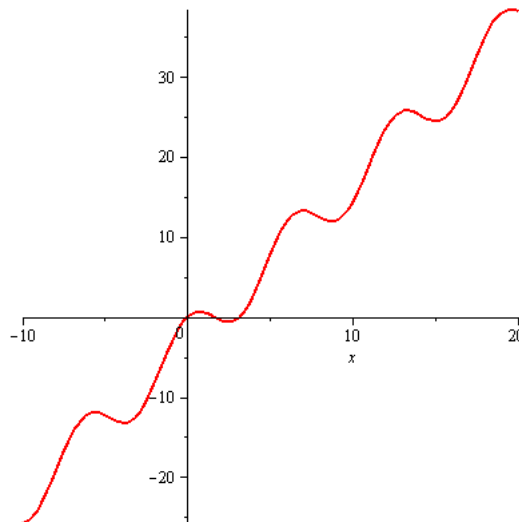
$$cond := y(0) = 0, y\left(\frac{1}{2} \pi\right) = 0$$

```
> dsolve({de,cond},y(x));
```

$$y(x) = \cos(x) \pi + 2x - \pi$$

Замечание: для построения графика решения предварительно следует отделить правую часть полученного выражения.

```
> y1:=rhs(%):plot(y1,x=-10..20,thickness=2);
```



1.4 Решение систем дифференциальных уравнений

Команда **dsolve** может найти решение системы дифференциальных уравнений (или задачи Коши), если в ней указать: **dsolve({sys},{x(t),y(t),...})**, где **sys** – система дифференциальных уравнений, **x(t), y(t), ...** – набор неизвестных функций.

Задание 1.4.

Найти решение системы дифференциальных уравнений:

$$\begin{cases} \dot{x} = -4x - 2y + \frac{2}{e^t - 1}, \\ \dot{y} = 6x + 3y - \frac{3}{e^t - 1} \end{cases}$$

```
> sys:=diff(x(t),t)=-4*x(t)-2*y(t)+2/(exp(t)-1),
diff(y(t),t)=6*x(t)+3*y(t)-3/(exp(t)-1);
```

$$sys := \frac{d}{dt} x(t) = -4x(t) - 2y(t) + \frac{2}{e^t - 1}, \frac{d}{dt} y(t) = 6x(t) + 3y(t) - \frac{3}{e^t - 1}$$

```
> dsolve({sys},{x(t),y(t)});
```

$$\left\{ x(t) = \frac{2 + 2 \ln(e^t - 1) - C1}{e^t} + C2, y(t) = -2C2 - 3e^{-t} - 3e^{-t} \ln(e^t - 1) + \frac{3}{2} e^{-t} C1 \right\}$$

Найдены две функции $x(t)$ и $y(t)$, которые зависят от двух произвольных постоянных $C1$ и $C2$.

1.5 Проверка найденного решения

Команда `odetest(sol,de)` позволяет проверить, что найденное решение `sol` удовлетворяет уравнению или системе `de`. Может потребоваться выполнить дополнительные преобразования с помощью команд `simplify`, `combine`, `expand`, чтобы получить ноль.

Проверим решение дифференциального уравнения из задания 1.1.1.

```
> restart; de:=diff(y(x),x)+y(x)*cos(x)=sin(x)*cos(x):
```

```
> sol:=dsolve(de,y(x));
```

$$sol := y(x) = \sin(x) - 1 + e^{-\sin(x)} C1$$

```
> odetest(sol,de);
```

0

Проверим решение краевой задачи из задания 1.3.2:

```
> restart; de:=diff(y(x),x$2)+y(x)=2*x-Pi:
```

```
> cond:=y(0)=0,y(Pi/2)=0:
```

```
> sol:=dsolve({de,cond},y(x));
```

$$sol := y(x) = \cos(x) \pi + 2x - \pi$$

```
> odetest(sol,{de,cond});
```

{0}

1.6 Интегральные преобразования для решения дифференциальных уравнений

Для решения дифференциальных уравнений и задач с начальными данными можно применять интегральные преобразования. Для этого в качестве опций команды `dsolve` нужно задать метод интегральных преобразований в `method=transform`, где в качестве `transform` можно указать: `laplace`, `fourier`, `fouriercos`, `fouriersin`. Для использования метода `laplace` должен быть задан весь набор начальных условий.

```
> ode := diff(y(x), x, x) = 2*y(x)+2; icond:=y(0) = 1, (D(y))(0) = 0;
```

$$ode := \frac{d^2}{dx^2} y(x) = 2y(x) + 2$$

$$icond := y(0) = 1, D(y)(0) = 0$$

```
> dsolve({ode,icond}, y(x));
```

$$y(x) = e^{\sqrt{2}x} + e^{-\sqrt{2}x} - 1$$

```
> dsolve({ode,icond}, y(x), method=laplace);
```

$$y(x) = -1 + 2 \cosh(\sqrt{2}x)$$

§2. Приближенные решения дифференциальных уравнений

Для многих типов дифференциальных уравнений не может быть найдено точное аналитическое решение. В этом случае дифференциальное уравнение можно решить с помощью приближенных методов, и, в частности, с помощью разложения в степенной ряд неизвестной функции.

Чтобы найти приближенное решение дифференциального уравнения в виде степенного ряда, в команде **dsolve** следует после переменных указать параметр **type=series** (или просто **series**). Для того, чтобы указать порядок разложения **n**, т.е. порядок степени, до которой производить разложение, следует перед командой **dsolve** вставить определение порядка с помощью команды **Order:=n**.

Если ищется общее решение дифференциального уравнения в виде разложения в степенной ряд, то коэффициенты при степенях x найденного разложения будут содержать неизвестные значения функции в нуле $y(0)$ и ее производных $D(y)(0)$, $(D@@2)(y)(0)$ и т.д. Полученное в строке вывода выражение будет иметь вид, похожий на разложение искомого решения в ряд Маклорена, но с другими коэффициентами при степенях x . Для выделения частного решения следует задать начальные условия $y(0)=y1$, $D(y)(0)=y2$, $(D@@2)(y)(0)=y3$ и т.д., причем количество этих начальных условий должно совпадать с порядком соответствующего дифференциального уравнения.

Разложение в степенной ряд имеет тип **series**, поэтому для дальнейшей работы с этим рядом его следует преобразовать в полином с помощью команды **convert(% ,polynom)**, а затем выделить правую часть полученного выражения командой **rhs(%)**.

Задание 2.

2.1. Найти решение задачи Коши: $y' = y + xe^y$, $y(0) = 0$ в виде степенного ряда с точностью до 5-го порядка.

```
> restart; Order:=5:
```

```
> dsolve({diff(y(x), x) = y(x) + x*exp(y(x)), y(0)=0}, y(x), type=series);
```

$$y(x) = \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{6}x^4 + O(x^5)$$

В полученном решении слагаемое $O(x^5)$ означает, что точность разложения была до 5-го порядка.

2.2. Найти общее решение дифференциального уравнения $y''(x) - y^3(x) = e^{-x} \cos x$, в виде разложения в степенной ряд до 4-го порядка. Найти разложение при начальных условиях: $y(0)=1$, $y'(0)=0$.

```
> restart; Order:=4: de:=diff(y(x), x, x) - y(x)^3 = exp(-x) * cos(x);
```

$$de := \frac{d^2}{dx^2} y(x) - y(x)^3 = e^{-x} \cos(x)$$

```
> f:=dsolve(de, y(x), series);
```

$$f := y(x) = y(0) + D(y)(0)x + \left(\frac{1}{2}y(0)^3 + \frac{1}{2}\right)x^2 + \left(\frac{1}{2}y(0)^2 D(y)(0) - \frac{1}{6}\right)x^3 + O(x^4)$$

Замечание: в полученном разложении запись **D(y)(0)** обозначает производную в нуле: $y'(0)$. Для нахождения частного решения осталось задать начальные условия:

> **subs(y(0)=1, D(y)(0)=0, f);**

$$y(x) = 1 + x^2 - \frac{1}{6}x^3 + O(x^4)$$

2.3. Найти точное и приближенное решение в виде степенного ряда до 6-го порядка задачи Коши: $y''' - y' = 3(2 - x^2)\sin x$, $y(0) = 1$, $y'(0) = 1$, $y''(0) = 1$. Построить на одном рисунке графики точного и приближенного решений.

> *restart*; $dsys := y''' - y' = 3 \cdot (2 - x^2) \cdot \sin(x)$, $y(0) = 1$, $y'(0) = 1$, $y''(0) = 1$;

$$dsys := \frac{d^3}{dx^3} y(x) - \left(\frac{d}{dx} y(x)\right) = 3(2 - x^2)\sin(x), y(0) = 1, D(y)(0) = 1, D^{(2)}(y)(0) = 1$$

> **restart; dsys := diff(y(x), x, x, x) - (diff(y(x), x)) = (3*(2-x^2))*sin(x), y(0) = 1, (D(y))(0) = 1, ((D@@2)(y))(0) = 1:**

Точное решение:

> **dsolve({dsys}, y(x));**

$$y(x) = \frac{3}{4}e^{-x} + \frac{7}{4}e^x + \frac{21}{2}\cos(x) + 6\sin(x)x - \frac{3}{2}\cos(x)x^2 - 12$$

Выделим правую часть для построения графика

> **y1 := rhs(%);**

$$y1 := \frac{3}{4}e^{-x} + \frac{7}{4}e^x + \frac{21}{2}\cos(x) + 6\sin(x)x - \frac{3}{2}\cos(x)x^2 - 12$$

Приближенное решение:

> **Order := 6: dsolve({dsys}, y(x), series);**

$$y(x) = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{7}{24}x^4 + \frac{1}{120}x^5 + O(x^6)$$

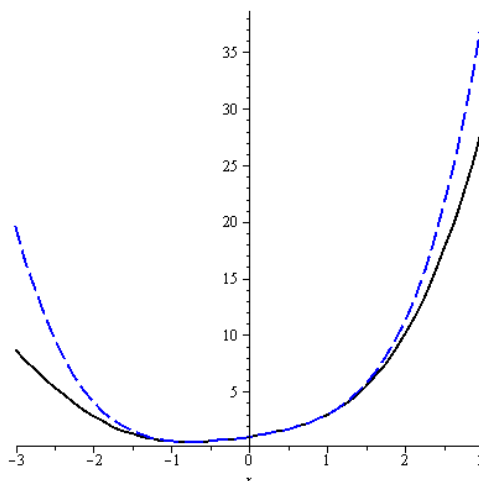
Замечание: тип решения дифференциального уравнения в виде ряда есть **series**, поэтому для дальнейшего использования такого решения (вычислений или построения графика) его обязательно следует конвертировать в полином с помощью команды **convert**

Выделим правую часть и конвертируем в полином для построения графика

> **y2 := convert(rhs(%), polynom);**

$$y2 := 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{7}{24}x^4 + \frac{1}{120}x^5$$

> **plot([y1, y2], x = -3 .. 3, color = [black, blue], thickness = 2, linestyle = [solid, dash]);**



На этом рисунке видно, что наилучшее приближение точного решения степенным рядом достигается примерно на интервале $-1 < x < 1$.

§3. Численные решения дифференциальных уравнений, построение графиков численных решений

Для того, чтобы найти численное решение дифференциального уравнения (задачи Коши или краевой задачи) в команде `dsolve` следует указать параметр `type=numeric` (или просто `numeric`). Тогда команда решения дифференциального уравнения будет иметь вид `dsolve(eq, vars, type=numeric, options)`, где `eq` – уравнения, `vars` – список или множество неизвестных функций, `options` – параметры.

В дополнительных опциях можно указать метод численного интегрирования дифференциального уравнения в виде `method=numeric_method`, где в качестве `numeric_method` можно указать: `rkf45`, `rosenbrock`, `bvp`, `rkf45_dae`, `rosenbrock_dae`, `dverk78`, `lsode`, `gear`, `taylorseries`, `mebdfi`, `classical`. Например, `method=rkf45` – метод Рунге-Кутты-Фельберга 4-5-ого порядка (установлен по умолчанию); `method=dverk78` – метод Рунге-Кутты 7-8 порядка; `method=classical` – классический метод Рунге-Кутты 3-его порядка; `method=gear` – метод Гира.

В результате выполнения команды `dsolve` с параметром `type=numeric` будет создана процедура, к которой можно обращаться для вычисления решения при заданном значении переменной и для построения графика на заданном промежутке.

График численного решения дифференциального уравнения можно построить с помощью команды `odeplot(dd, [x,y(x)], x=x1..x2)` из пакета `plots`, где в качестве функции используется команда `dd:=dsolve({eq,cond}, y(x), numeric)` численного решения, после нее в квадратных скобках указывают переменную и неизвестную функцию `[x,y(x)]`, и интервал `x=x1..x2` для построения графика.

Задание 3

3.1. Найти численное и приближенное решение в виде степенного ряда до 6-ого порядка задачи Коши: $y'' - x \sin(y) = \sin 2x$, $y(0) = 0$, $y'(0) = 1$.

Сначала найдем численное решение задачи Коши и построим его график.

```
> restart; eq:=diff(y(x), x$2) - x*sin(y(x)) = sin(2*x) :
  cond:=y(0)=0, D(y)(0)=1 :
> sol_num:=dsolve({eq,cond}, y(x), numeric) ;
      sol_num := proc(x_rkf45) ... end proc
```


Замечание: в строке вывода появляется сообщение о том, что при решении использован метод *rkf45*. Во избежание вывода строк, не несущих полезной информации, рекомендуется отделять промежуточные команды двоеточием.

Если необходимо получить значение решения при каком-то фиксированном значении переменной x (заодно будет выведено значение производной решения в этой точке), например, при $x=0.5$, то следует набрать:

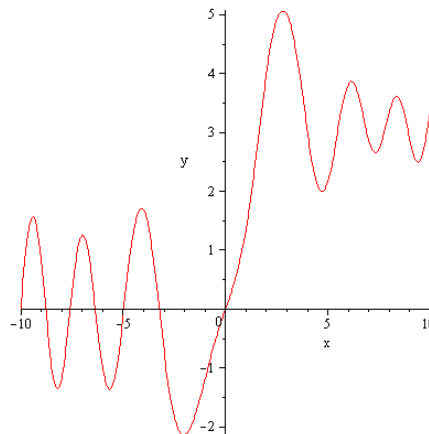
```
> sol_num(0.5);
```

$$\left[x = 0.5, y(x) = 0.544926322815993624, \frac{d}{dx} y(x) = 1.27250331302082364 \right]$$

Построим график численного решения на интервале $(-10,10)$:

```
> with(plots):
```

```
> odeplot(sol_num, [x, y(x)], -10..10, numpoints=200);
```



Теперь найдем приближенное решение задачи Коши в виде степенного ряда и построим графики численного решения и полученного степенного ряда в интервале их наилучшего совпадения.

```
> sol_series:=dsolve({eq, cond}, y(x), series);
```

$$sol_series := y(x) = x + \frac{1}{3} x^3 + \frac{1}{12} x^4 - \frac{1}{15} x^5 + O(x^6)$$

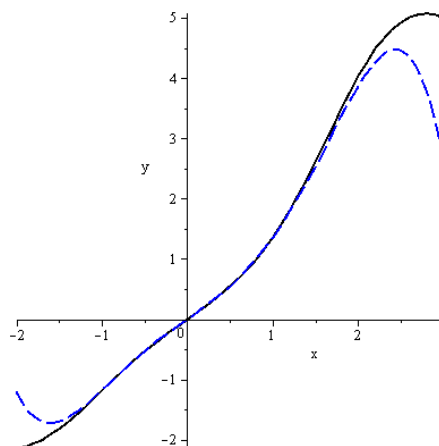
```
> convert(%, polynomial): sol_series:=rhs(%);
```

$$sol_series := x + \frac{1}{3} x^3 + \frac{1}{12} x^4 - \frac{1}{15} x^5$$

```
> p1:=odeplot(sol_num, [x, y(x)], -2..3, thickness=2, color=black):
```

```
p2:=plot(sol_series, x=-2..3, thickness=2, linestyle=3, color=blue):
```

```
> display(p1,p2);
```

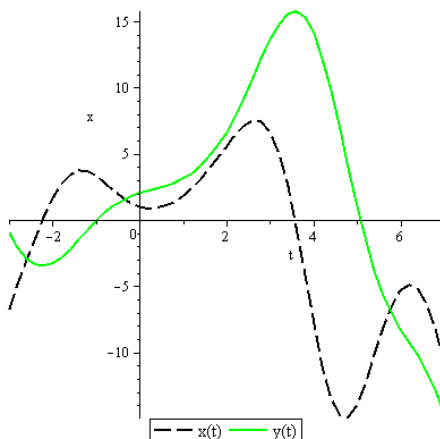


Наилучшее приближение решения степенным рядом достигается примерно на интервале $-1 < x < 1$ (так же как и в примере 3 задания 2.3).

3.2. Построить графики решений задачи Коши системы дифференциальных уравнений:

$$x'(t)=2y(t)\sin(t)-x(t)-t, y'(t)=x(t), x(0)=1, y(0)=2.$$

```
> restart;
> cond:=x(0)=1,y(0)=2: sys:=diff(x(t),t)=2*y(t)*sin(t)-x(t)-t,
diff(y(t),t)=x(t):
> F:=dsolve({sys,cond},[x(t),y(t)],numeric):
> with(plots):
> p1:=odeplot(F,[t,x(t)],-3..7,color=black,thickness=2,
linestyle=3,legend="x(t)"):
> p2:=odeplot(F,[t,y(t)],-3..7,color=green,thickness=2,
legend="y(t)"):
> display(p1,p2);
```



§4. Графическое представление решений дифференциальных уравнений, построение фазовых портретов

4.1. Двумерные графики решений ДУ или системы ДУ. Команда DEplot

Для численного решения задачи Коши, построения графиков решения и фазовых портретов в *Maple* имеется специальный пакет **DEtools**.

```
DEplot(deqns, vars, trange, options)
DEplot(deqns, vars, trange, inits, options)
DEplot(deqns, vars, trange, xrange, yrange, options)
DEplot(deqns, vars, trange, inits, xrange, yrange, options)
```

Команда **DEplot** из пакета **DEtools** строит двумерные графики решений обыкновенных дифференциальных уравнений или систем ДУ, а также фазовые портреты. Эта команда аналогична команде **odeplot**, но более функциональна. Она, в отличие от **odeplot**, сама производит численное решение дифференциального уравнения. Основные параметры **DEplot** похожи на параметры **odeplot**: **DEplot(de, vars, range, x=x1..x2, y=y1..y2, cond, options)**, где **de** – дифференциальное уравнение или система дифференциальных уравнений; **vars** – список неизвестных функций; **range** – диапазон измерения независимой переменной; **cond** – начальные условия; **x=x1..x2** и **y=y1..y2** – диапазоны изменения функций; **options** – дополнительные параметры.

Наиболее часто используемые параметры: **linecolor**=цвет линии; **scene=[x,y]** – определяет, какие зависимости выводить на график; **iterations**=число итераций, необходимое для повышения точности вычислений (по умолчанию это число равно 1); **stepsize**=число, равное расстоянию между точками на графике, по умолчанию оно равно $(x_2-x_1)/48$, этот параметр необходим для вывода более гладкой кривой решения; **obsrange=true/false** – прерывать или нет вычисления, если график решения выходит за установленный для рисования интервал.

Для решения дифференциального уравнения n -ого порядка начальные условия можно задавать в более компактной форме: **[x0, y0, y'0, y''0, ...]**, где **x0** – точка, в которой задаются начальные условия, **y0** – значение искомой функции в точке **x0**, **y'0**, **y''0**, ... – значения производных первой, второй и т.д. до $(n-1)$ -ого порядка.

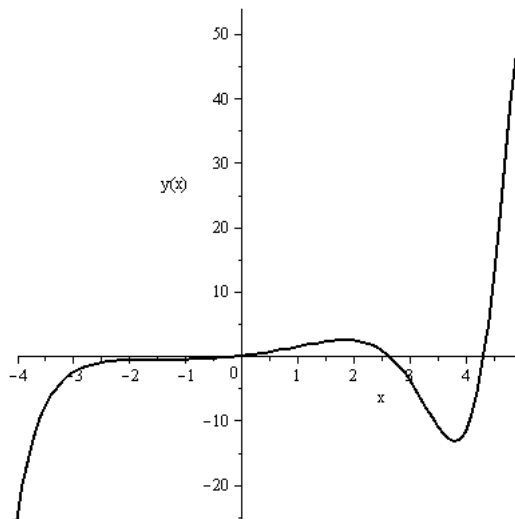
Задание 4.1

Нарисовать график решения дифференциального уравнения:

$$y'''+x\sqrt{|y|}+x^2y=0, \quad y(0)=0, \quad y'(0)=1, \quad y''(0)=1 \text{ в интервале } x \in [-4,5].$$

> restart; with(DEtools):

```
DEplot(diff(y(x),x$3)+x*sqrt(abs(diff(y(x),x)))+x^2*y(x)=0,
{y(x)}, x=-4..5, [[y(0)=0,D(y)(0)=1,(D@@2)(y)(0)=1]],
stepsize=.1, linecolor=black, thickness=2);
```



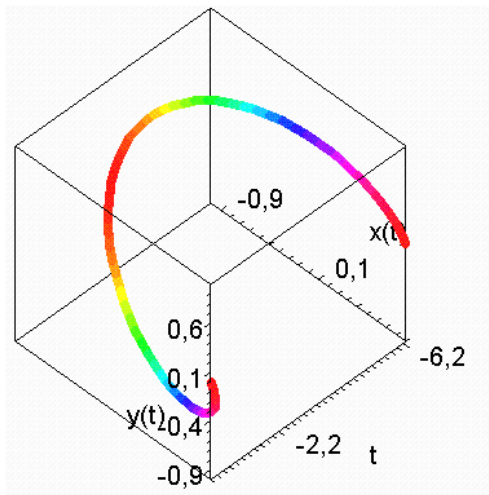
4.2. Трехмерные графики решений ДУ или системы ДУ. Команда DEplot3d

Команда **DEplot3d** из пакета **DEtools** строит трехмерные графики решений обыкновенных дифференциальных уравнений или систем ДУ.

```
> with(DEtools): desys:=diff(x(t),t) = -sin(t), diff(y(t),t) =
cos(t);
```

$$desys := \frac{d}{dt} x(t) = -\sin(t), \quad \frac{d}{dt} y(t) = \cos(t)$$

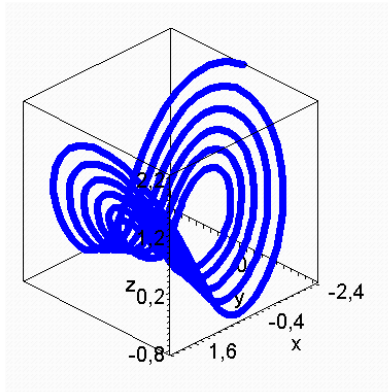
```
> DEplot3d({desys},{x(t),y(t)},t=-2*Pi..0, [[y(0)=0, x(0)=1]],
scene = [t, x(t), y(t)], stepsize = .1, linecolor = cos(t));
```



```

> restart; ode3:=diff(x(t),t)=y(t),diff(z(t),t)=
-nu*z(t)+x(t)*y(t),diff(y(t),t)=-mu*y(t)-(z(t)-1)*x(t)-x(t)^3;
inits:=[x(0)=1,y(0)=1,z(0)=0];
ode3 := d/dt x(t) = y(t), d/dt z(t) = -nu z(t) + x(t) y(t), d/dt y(t) = -mu y(t) - (z(t) - 1) x(t) - x(t)^3
inits := [x(0) = 1, y(0) = 1, z(0) = 0]
> mu:=-0.1:nu:=0.1: with(DEtools):
DEplot3d([ode3],[x,y,z],t=0..25,[inits],stepsize=0.05,orientatio
n=[45,60],linecolor=blue);

```



4.3. Построение фазовых портретов систем дифференциальных уравнений. Команды DEplot и phaseportrait пакета DEtools.

Для дифференциального уравнения порядка выше первого команда **DEplot** рисует только кривые решений дифференциальных уравнений, а для систем дифференциальных уравнений первого порядка могут быть нарисованы и фазовые портреты.

Фазовый портрет на плоскости (x,y) для системы двух дифференциальных уравнений $\frac{dx}{dt} = f(x,y,t)$, $\frac{dy}{dt} = g(x,y,t)$ можно построить с помощью команды **DEplot**, если в параметрах данной команды указать **scene=[x,y]**.

Для *автономной* системы дифференциальных уравнений (т. е. не зависящей явно от t) на фазовом портрете будет построено поле направлений в виде стрелок. Размер стрелок регулируется параметром **arrows=SMALL, MEDIUM, LARGE, LINE** или **NONE**. Проверить

систему ДУ на автономность можно с помощью команды `autonomous(des, vars, ivar)`.

Для того, чтобы нарисовать весь фазовый портрет, необходимо для каждой фазовой траектории указывать начальные условия: например, для системы двух дифференциальных уравнений первого порядка несколько начальных условий в команде `DEplots` указываются после задания диапазона изменения независимой переменной `t`: `[[x(0)=x1, y(0)=y1], [x(0)=x2, y(0)=y2], ..., [x(0)=xn, y(0)=yn]]`. Начальные условия можно задавать в более компактной форме: `[t0, x0, y0]`, где `t0` – точка, в которой задаются начальные условия, `x0` и `y0` – значения искомых функций в точке `t0`.

Фазовый протрет системы двух дифференциальных уравнений первого порядка можно также построить с помощью команды `phaseportrait(sys, [x,y], x1..x2, [[cond]])`, где `sys` – система двух дифференциальных уравнений первого порядка, `[x,y]` – имена искомых функций, `x1..x2` – интервал, на котором следует построить фазовый портрет, а в фигурных скобках указываются начальные условия.

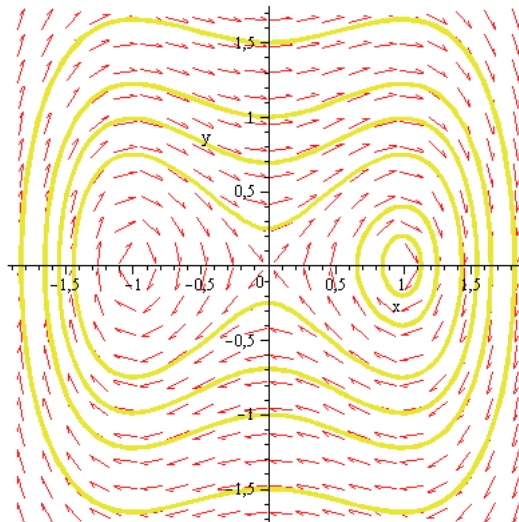
Задание 4.3.

4.3.1. Построить фазовый портрет нелинейной автономной системы дифференциальных

уравнений:
$$\begin{cases} \dot{x} = y \\ \dot{y} = x - x^3 \end{cases}$$
 для нескольких наборов начальных условий: $x(0)=1, y(0)=0.2$;

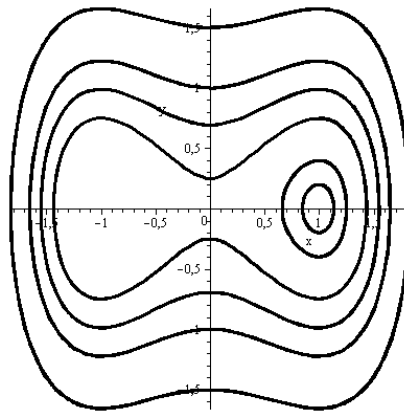
$x(0)=0, y(0)=1$; $x(0)=1, y(0)=0.4$; $x(0)=1, y(0)=0.75$; $x(0)=0, y(0)=1.5$; $x(0)=-0.1, y(0)=0.7$.

```
> restart; with(DEtools):
> DEplot({diff(x(t),t)=y(t), diff(y(t),t)=x(t)-x(t)^3},
[x(t),y(t)], t=0..20, [[0,1,0.2], [0,0,1], [0,1,0.4],
[0,1,0.75], [0,0,1.5], [0,-0.1,0.7]], stepsize=0.1);
```



Для того, чтобы отобразить фазовый портрет без поля направлений, нужно в команде `DEplot` указать опцию `arrows=None`:

```
> DEplot({diff(x(t),t)=y(t), diff(y(t),t)=x(t)-x(t)^3},
[x(t),y(t)], t=0..20, [[0,1,0.2], [0,0,1], [0,1,0.4],
[0,1,0.75], [0,0,1.5], [0,-0.1,0.7]], stepsize=0.1,
arrows=None, linecolor=black);
```

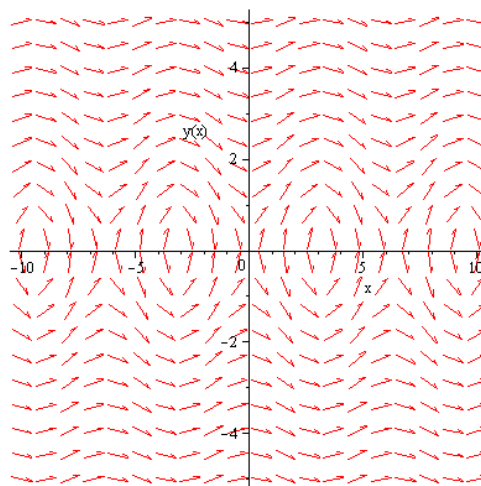


Видно, что особыми точками системы являются точки $(-1,0)$, $(0,0)$ и $(1,0)$. При этом положения равновесия $(-1,0)$ и $(1,0)$ являются центрами, а точка $(0,0)$ – седлом.

4.3.2. Построить поле направлений для нелинейного дифференциального уравнения $y' = \sin x / y$.

```
> restart; with(DEtools): nonlin_ode:=diff(y(x),x)=sin(x)/y(x);
> DEplot(nonlin_ode,y(x),x=-10..10,y=-5..5, stepsize=0.05);
```

$$\text{nonlin_ode} := \frac{d}{dx} y(x) = \frac{\sin(x)}{y(x)}$$



Найдем особые точки:

```
> _EnvAllSolutions:=true:solve([sin(x)=0,y=0]);
{x = pi_2I~, y = 0}
```

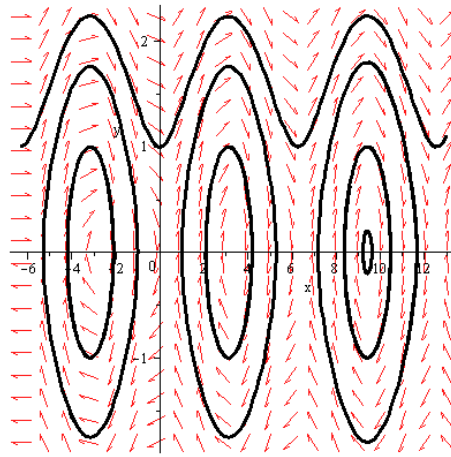
Таким образом, особые точки имеют координаты $(\pi n, 0)$. Видно, что они являются центрами.

Построим фазовый портрет с полем направлений для эквивалентной данному уравнению

нелинейной автономной системы $\begin{cases} \dot{x} = y \\ \dot{y} = \sin x \end{cases}$ при различных начальных условиях $x(0)=1,$

$y(0)=0; x(0)=-1, y(0)=0; x(0)=\pi, y(0)=1; x(0)=-\pi, y(0)=1; x(0)=3\pi, y(0)=0.2; x(0)=3\pi, y(0)=1;$
 $x(0)=3\pi, y(0)=1.8; x(0)=-2\pi, y(0)=1.$

```
> restart; with(DEtools):
> sys:=diff(x(t),t)=y(t), diff(y(t),t)=sin(x(t)):
> DEplot({sys},[x(t),y(t)], t=0..4*Pi, [[0,1,0], [0,-1,0],
[0,Pi,1], [0,-Pi,1], [0,3*Pi,0.2], [0,3*Pi,1], [0,3*Pi,1.8],
[0,-2*Pi,1]],stepsize=0.1, linecolor=black);
```



4.3.1. Построить фазовый портрет линейной системы дифференциальных уравнений:

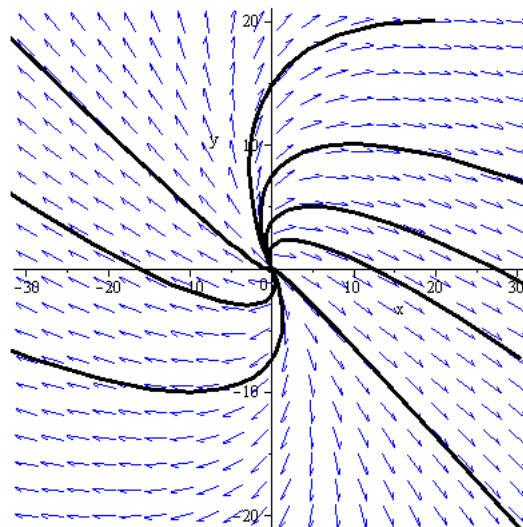
$$\begin{cases} x' = 3x + y \\ y' = y - x \end{cases} . \text{ Начальные условия, диапазон изменения переменной и размеры координатных}$$

осей побрать самостоятельно из соображений наглядности фазового портрета.

```
> restart; with(DEtools):
```

```
> sys:=diff(x(t),t)=3*x(t)+y(t), diff(y(t),t)=-x(t)+y(t):
```

```
> phaseportrait([sys],[x(t),y(t)],t=-10..10, [[0,1,-2],[0,-3,-3], [0,-2,4],[0,5,5],[0,5,-3],[0,-5,2],[0,5,2],[0,-1,2]], x=-30..30,y=-20..20, stepsize=.1, colour=blue,linecolor=black);
```



4.4. Изображение поля направлений. Команда dfieldplot.

```
dfieldplot(deqns, vars, trange, xrange, yrange, options)
```

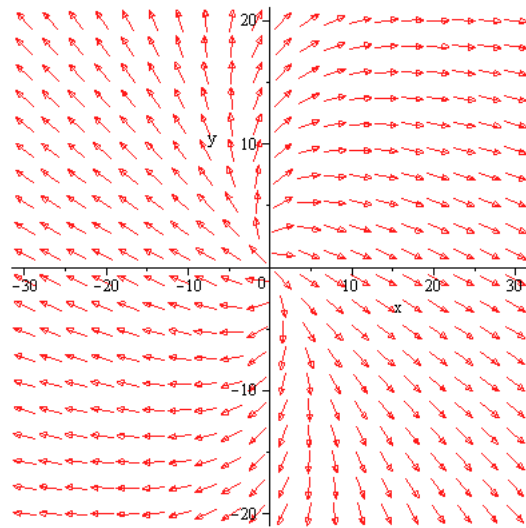
Команда **dfieldplot** предназначена для изображения векторного поля (поля направлений). Параметры этой команды аналогичны соответствующим параметрам команды DEplot. Для команды команды **dfieldplot** не требуется задавать начальные условия.

Поле направлений для системы из предыдущего задания:

```
> restart; with(DEtools):
```

```
> sys:=diff(x(t),t)=3*x(t)+y(t), diff(y(t),t)=-x(t)+y(t):
```

```
> dfieldplot([sys],[x(t),y(t)],t=-10..10, x=-30..30,y=-20..20, stepsize=.1, arrows = SLIM);
```



Литература

Савотченко С.Е., Кузьмичева Т.Г. Методы решения математических задач в *Maple*: Учебное пособие – Белгород: Изд. Белаудит, 2001. – 116 с.