Создание анимации в SMath Studio.Часть 1

Fridel Selitsky

1.Изображение точки и ее траектории

Для создания анимации надо знать закон движения точки (зависимость координат x и y от времени t), начальное и конечное значения переменной t и шаг по времени.Если закон движения точки не задан, то его нужно найти из решения дифференциальных уравнений движения.

Пример 1.1 **Изобразить точку, которая движется по заданному закону**

$$x(t) := \mathrm{eval}(4 \cdot \cos(\Delta\theta \cdot t))$$

$$y(t) := \mathrm{eval}(2 \cdot \sin(\Delta\theta \cdot t))$$

Заданы начальное и конечное значения переменной и шаг по времени

$\Delta\theta := \dfrac{\pi}{36}$ шаг изменения угла

$\theta := 2 \cdot \pi$ угол поворота

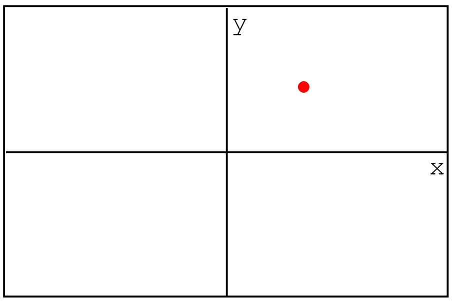
$tstart := 0$ начальное значение переменной t

$tend := \dfrac{\theta}{\Delta\theta}$ конечное значение переменной t

Программа анимации

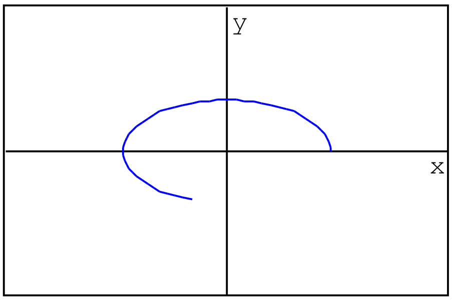
$step := tstart\,..\,tend$ границы изменения переменной t

$f1(t) := \left|\left(x(t)\ \ y(t)\ \ "."\ \ 15\ \ "Red"\right)\right.$



Пример 1.2 **Изобразить траекторию точки в виде сплошной линии**

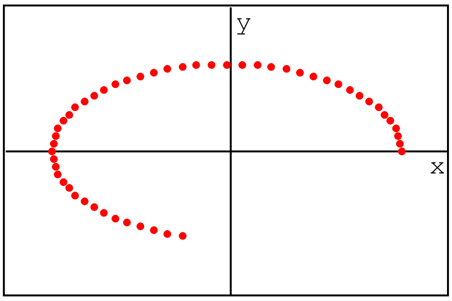
$$f2(t, path) := \left| \begin{array}{l} \text{if } t = tstart \\ \quad path := \left(x(t)\ \ y(t)\right) \\ \text{else} \\ \quad path := \mathrm{stack}\left(path, \left(x(t)\ \ y(t)\right)\right) \end{array} \right.$$



Примечание 1: Переменные, которые изменяются внутри функции рисования нужно указывать в качестве параметров функции.

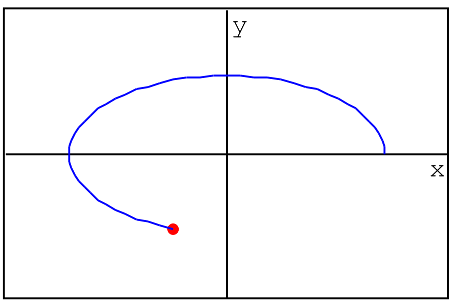
Пример 1.3 **Изобразить траекторию в виде точек**

```
f3(t , path):= | if  t = tstart
               |   path := (x(t) y(t) "." 10 "Red")
               | else
               |   path := stack( path , (x(t) y(t) "." 10 "Red"))
```



Пример 1.4 **Изобразить точку, которая вычерчивает траекторию**

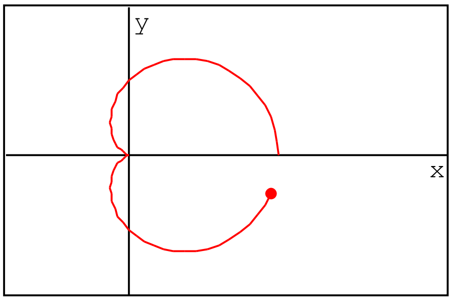
```
f4(t , path):= | { if  t = tstart
               | {   path := (x(t) y(t))
               | { else
               | {   path := stack( path , (x(t) y(t)))
               | { (x(t) y(t) "." 15 "Red")
```



Пример 1.5 **Закон движения точки задан в полярных координатах**

$r(\Delta\theta) := \text{eval}(\cos(\Delta\theta \cdot t) + 1)$

```
f5(t, path) := | "Преобразование от полярных координат к прямоугольным"
               | x := eval(cos(Δθ·t)·r(Δθ))
               | y := eval(sin(Δθ·t)·r(Δθ))
               | { (x y "." 15 "Red")
               | { if t = tstart
               | {   path := (x y)
               | { else
               | {   path := stack(path, (x y))
```



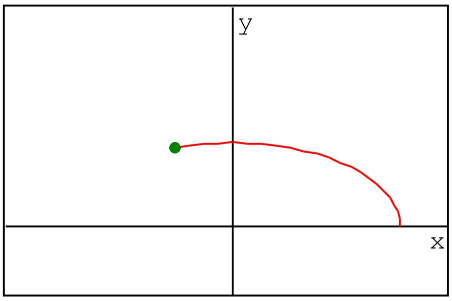
Пример 1.6 **Точка движется туда и обратно**

$\theta := \pi$ угол поворота

$tend := \dfrac{\theta}{\Delta\theta}$ конечное значение переменной t

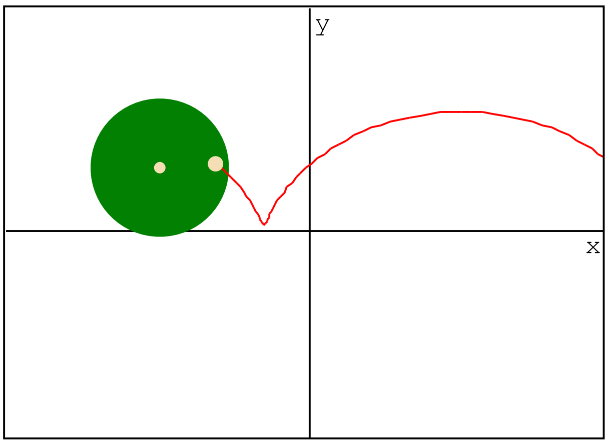
$\tau := tstart \,..\, 2 \cdot tend$

```
f6(t, path) := | if t > tend
               |   Δθ := -π/36
               | else
               |   Δθ := π/36
               | x := eval(4·cos(Δθ·t))
               | y := eval(2·sin(Δθ·t))
               | { (x y "." 15 "green")
               | { if (t = tstart) ∨ (t = tend)
               | {   path := (x y)
               | { else
               | {   path := stack(path, (x y))
```



Пример1.7 Заданы радиусы R и r и закон движения центра C катящегося диска

```
R := 2    r := 1.8
step := tstart .. 3·tend

Disk( t , path ) := | "закон движения точки C"
                    | xC := eval( 8 − R·Δθ·t )
                    | yC := R
                    | xA := eval( xC + r·cos( (Δθ)·t ) )
                    | yA := eval( yC + r·sin( (Δθ)·t ) )
                    | ( (xA  yA  "."  20  "wheat")
                    | { if t = tstart
                    | {   path := (xA  yA)
                    | { else
                    | {   path := stack( path , (xA  yA) )
                    | ( xC  yC  "."  180  "green"
                    |   xC  yC  "."  15   "wheat" ) )
```



Пример 1.8 Маятник на эластичной нити (закон движения точки не задан)

z - удлинение нити
φ - угол отклонения нити от вертикали

Исходные данные:

L0 := 2 `длина нерастянутой нити(m)`

$c := 7$ коэффициент жесткости нити(N/m)

$m := 1$ масса груза(kg)

$g := 9.81$ m/s^2

Начальные значения:

$\varphi 0 := \frac{1}{3} \cdot \pi$ $\omega 0 := 0$ $z0 := 1$ $Vz0 := 0$

$L1 := L0 + m \cdot \frac{g}{c}$ длина нити после статического удлинения

Координаты точки (груза) являются неизвестными функциями от φ(t) и z(t)

$x := (L1 + z(t)) \cdot \sin(\varphi(t))$

$y := -(L1 + z(t)) \cdot \cos(\varphi(t))$

Решение проводим в два этапа:
a) Решение дифференциальных уравнений для нахождения функций φ(t), z(t)
b) Создание анимации

a) Решение системы дифференциальных уравнений

$y_1 := \varphi^{\blacksquare}$ $y_2 := z^{\blacksquare}$ $y_3 := \omega^{\blacksquare}$ $y_4 := Vz^{\blacksquare}$

$$\begin{cases} t0 := 0 \\ ySol := \begin{pmatrix} \varphi 0 \\ z0 \\ \omega 0 \\ Vz0 \end{pmatrix} \end{cases}$$ Начальные условия

$$f(t, y) := \begin{pmatrix} y_3 \\ y_4 \\ -\frac{g \cdot \sin(y_1)}{(L1 + y_2)} - \frac{2 \cdot y_3 \cdot y_4}{(L1 + y_2)} \\ \left(-\frac{c \cdot y_2}{m} - g \cdot (1 - \cos(y_1)) + (L1 + y_2) \cdot (y_3)^2\right) \end{pmatrix}$$ Правая часть системы уравнений

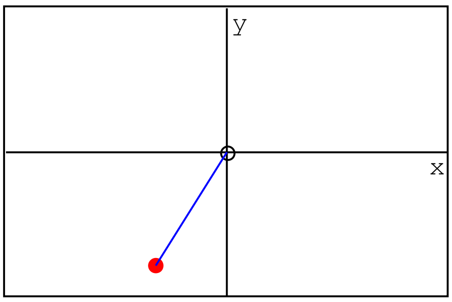
Подпрограмма расчёта методом Рунге-Кутта

```
RunKut(ySol) := | "Расчёт  методом Рунге-Кутта"
                | Δt := 0.06
                | y0 := ySol
                | tM := eval(t0 + 1/2·Δt)
                | K1 := eval(Δt·f(t0, y0))
                | yM := eval(y0 + 1/2·K1)
                | K2 := eval(Δt·f(tM, yM))
                | yM := eval(y0 + 1/2·K2)
                | K3 := eval(Δt·f(tM, yM))
                | ySol := eval(y0 + K3)
                | K4 := eval(Δt·f(t0, ySol))
                | ySol := eval(y0 + 1/6·(K1 + 2·K2 + 2·K3 + K4))
```

b) Программа анимации

$\tau := 0 .. 200$ границы изменения переменной t

```
Pendulum(t, ySol) := | "матрица значений φ,z,ω, Vz"
                     | φzωVz := RunKut(ySol)
                     | "находим функции φ(t),z(t)"
                     | φ(t) := eval(φzωVz₁)
                     | z(t) := eval(φzωVz₂)
                     | "координаты груза"
                     | x := eval((L1 + z(t))·sin(φ(t)))
                     | y := eval(-(L1 + z(t))·cos(φ(t)))
                     | { (0 0)
                     |   (x y)
                     | { (x y "." 20 "Red")
                     |   (0 0 "o" 12 "" )
```



Примечание 2: Переменная ySol является параметром, поскольку изменяется внутри функции рисования в процессе расчета

2.Изображение рычажных механизмов

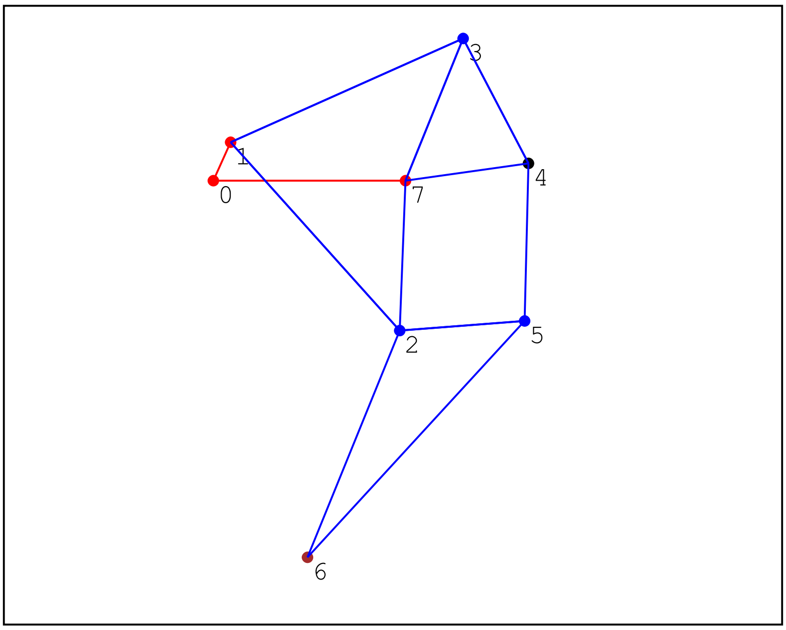
При изображении механизма необходимо по положению ведущего звена определить положения остальных звеньев. Для решения этой задачи используем геометрический метод, который основан на условии, что узловая точка соединения двух звеньев лежит на пересечении двух окружностей. Координаты узловых точек определяются с помощью вспомогательной программы PIntersect:

Координаты точек пересечения двух окружностей

$$\mathrm{PIntersect}(x0, y0, r0, x1, y1, r1) := \left| \begin{array}{l} x10 := x1 - x0 \\ y10 := y1 - y0 \\ d := \mathrm{eval}\left(\sqrt{x10^2 + y10^2}\right) \\ a := \mathrm{eval}\left(\dfrac{r0^2 - r1^2 + d^2}{2 \cdot d}\right) \\ h := \mathrm{eval}\left(\sqrt{r0^2 - a^2}\right) \\ x2 := \mathrm{eval}\left(x0 + \dfrac{a}{d} \cdot x10\right) \\ y2 := \mathrm{eval}\left(y0 + \dfrac{a}{d} \cdot y10\right) \\ P3 := \begin{pmatrix} x2 + \dfrac{h}{d} \cdot y10 & y2 - \dfrac{h}{d} \cdot x10 \\ x2 - \dfrac{h}{d} \cdot y10 & y2 + \dfrac{h}{d} \cdot x10 \end{pmatrix} \end{array} \right.$$

Покажем использование геометрического метода для нахождения координат первых двух узловых точек шагающего механизма. Координаты остальных точек определяются аналогично.

Шагающий механизм

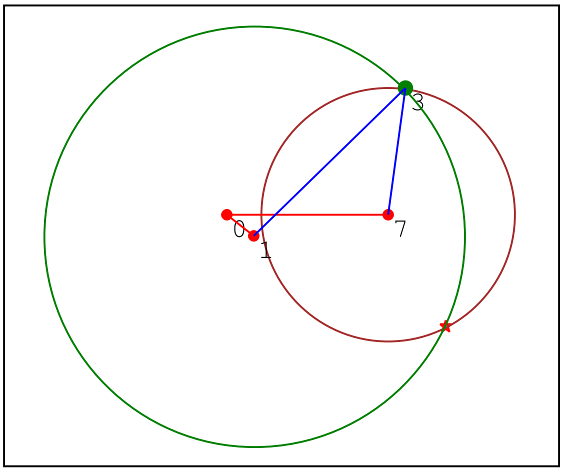


Пример 2.1 Найти положение точки 3

Точка 1 входного звена L01 описывает окружность. Точка 3 описывает дугу окружности радиуса L13 в ее относительном движении вокруг точки 1 и дугу окружности радиуса L73 вокруг точки 7. Точка 3 лежит на пересечении этих окружностей. Координаты точки 3 находим с помощью программы Pintrsect

$$f1(t) := \left| \begin{array}{l}
\text{"Координаты точки 1"} \\
x_1 := \mathrm{eval}(L01 \cdot \cos(\Delta\theta \cdot t)) \\
y_1 := \mathrm{eval}(L01 \cdot \sin(\Delta\theta \cdot t)) \\
\text{"Координаты точки 3"} \\
p3 := \mathrm{eval}(\mathrm{PIntersect}(x_1, y_1, L13, x7, y7, L73)) \\
x_3 := p3_{2\,1} \\
y_3 := p3_{2\,2} \\
\left\{ \begin{array}{l}
\begin{pmatrix} x_1 & x_3 & x7 \\ y_1 & y_3 & y7 \end{pmatrix}^T \\
\begin{pmatrix} x7 & y7 \\ 0 & 0 \\ x_1 & y_1 \end{pmatrix} \\
\begin{pmatrix} x_3 & y_3 & \text{"."} & 20 & \text{"green"} \end{pmatrix} \\
\begin{pmatrix} x7 & y7 & \text{"."} & 15 & \text{"red"} \end{pmatrix} \\
\begin{pmatrix} x7 & y7 & \text{"7"} & 10 & \text{""} \end{pmatrix} \\
\begin{pmatrix} x_1 & y_1 & \text{"."} & 15 & \text{"red"} \end{pmatrix} \\
\begin{pmatrix} 0 & 0 & \text{"."} & 15 & \text{"red"} \end{pmatrix} \\
\begin{pmatrix} 0 & 0 & \text{"0"} & 10 & \text{""} \end{pmatrix} \\
\begin{pmatrix} x_1 & y_1 & \text{"1"} & 10 & \text{""} \end{pmatrix} \\
\begin{pmatrix} x_3 & y_3 & \text{"3"} & 10 & \text{""} \\ x7 & y7 & \text{"o"} & 220 & \text{"brown"} \\ x_1 & y_1 & \text{"o"} & 365 & \text{"green"} \end{pmatrix} \\
\text{"Вторая точка пересечения окружностей"} \\
\begin{pmatrix} p3_{1\,1} & p3_{1\,2} & \text{"*"} & 12 & \text{"red"} \end{pmatrix}
\end{array} \right.
\end{array} \right.$$

Определение положения точки 3



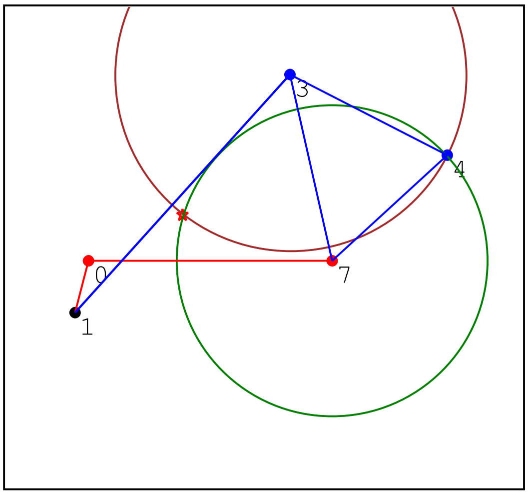
Примечание 3:На анимации видно,что окружности пересекаются в двух точках - 3 и точке,обозначенной красной звездочкой.Из двух точек выбираем одну,которая подходит к нашему механизму,т.е. точку 3.

Пример 2.2 **Найти положение точки 4**

Точка 4 описывает дугу окружности радиуса L34 в ее относительном движении вокруг точки 3 и дугу окружности радиуса L74 вокруг точки 7. Точка 4 лежит на пересечении этих окружностей. Координаты точки 4 находим с помощью программы Pintrsect:

$$f2(t) := \left|\begin{array}{l}
x_1 := \mathrm{eval}\left(L01 \cdot \cos\left(\Delta\theta \cdot t\right)\right) \\
y_1 := \mathrm{eval}\left(L01 \cdot \sin\left(\Delta\theta \cdot t\right)\right) \\
p2p3 := \mathrm{eval}\left(\mathrm{PIntersect}\left(x_1, y_1, L13, x7, y7, L73\right)\right) \\
x_3 := p2p3_{2\,1} \\
y_3 := p2p3_{2\,2} \\
\text{"Координаты точки 4"} \\
p4 := \mathrm{eval}\left(\mathrm{PIntersect}\left(x_3, y_3, L34, x7, y7, L74\right)\right) \\
x_4 := p4_{2\,1} \\
y_4 := p4_{2\,2} \\
\left\{\begin{array}{l}
\begin{pmatrix} x_1 & x_3 & x_1 & x_3 & x7 & x_4 & x_3 \\ y_1 & y_3 & y_1 & y_3 & y7 & y_4 & y_3 \end{pmatrix}^{T} \\
\begin{pmatrix} x7 & y7 \\ 0 & 0 \\ x_1 & y_1 \end{pmatrix} \\
\begin{pmatrix} x_3 & y_3 & \text{"."} & 15 & \text{"Blue"} \end{pmatrix} \\
\begin{pmatrix} x7 & y7 & \text{"."} & 15 & \text{"Red"} \end{pmatrix} \\
\begin{pmatrix} x7 & y7 & \text{"7"} & 10 & \text{""} \end{pmatrix} \\
\begin{pmatrix} x_1 & y_1 & \text{"."} & 15 & \text{""} \end{pmatrix} \\
\begin{pmatrix} 0 & 0 & \text{"."} & 15 & \text{"Red"} \end{pmatrix} \\
\begin{pmatrix} 0 & 0 & \text{"0"} & 10 & \text{""} \end{pmatrix} \\
\begin{pmatrix} x_1 & y_1 & \text{"1"} & 10 & \text{""} \end{pmatrix} \\
\begin{pmatrix} x_4 & y_4 & \text{"."} & 15 & \text{"Blue"} \\ x_4 & y_4 & \text{"4"} & 10 & \text{""} \\ x_3 & y_3 & \text{"3"} & 10 & \text{""} \\ x_3 & y_3 & \text{"o"} & 305 & \text{"brown"} \\ x7 & y7 & \text{"o"} & 270 & \text{"green"} \end{pmatrix} \\
\text{"Вторая точка пересечения окружностей"} \\
\begin{pmatrix} p4_{1\,1} & p4_{1\,2} & \text{"*"} & 12 & \text{"red"} \end{pmatrix}
\end{array}\right.
\end{array}\right.$$

Определение положения точки 4



Примечание 4:На анимации видно,что окружности пересекаются в двух точках - 4 и точке,обозначенной красной звездочкой.Из двух точек выбираем одну,которая подходит к нашему механизму,т.е. точку 4.

***Пример 2.3* Механизм Чебышева**

Исходные данные:

$xA := -440$ $yA := -140$

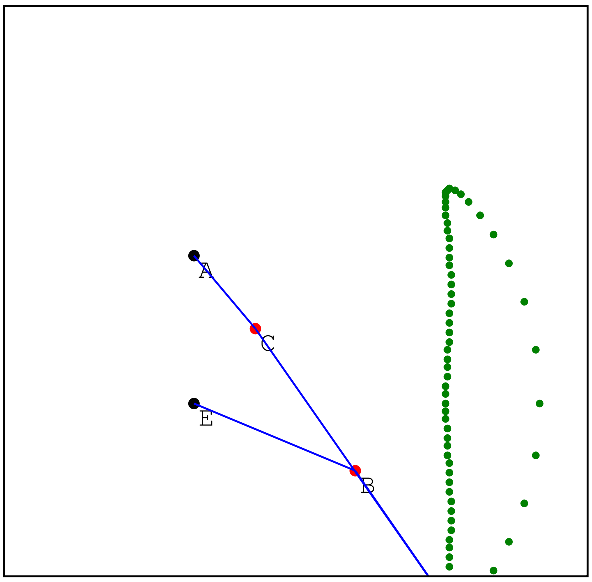
$yE := yA - 169$ $xE := xA$

$AC := 109$ $CB := 200$

$BD := CB$ $EB := CB$

С помощью программы Pintrsect находим координаты точки B.Координаты точки D находим из условия,что она находится посредине звена CD.

```
f4(t, path) :=
    "Координаты точки C"
    xC := eval(xA + AC·cos(Δθ·t))
    yC := eval(yA + AC·sin(Δθ·t))
    "Координаты точки B"
    pB := PIntersect(xC, yC, CB, xE, yE, EB)
    xB := pB[2 1]
    yB := pB[2 2]
    "Координаты точки D"
    xD := eval(2·xB - xC)
    yD := eval(2·yB - yC)
    { (xA  yA)
      (xC  yC)
      (xD  yD)
      (xB  yB)
      (xE  yE)
      (xA  yA  "."  15  ""     )
      (xC  yC  "."  15  "red"  )
      (xE  yE  "."  15  ""     )
      (xB  yB  "B"  10  ""     )
      (xD  yD  "D"  10  ""     )
      (xB  yB  "."  15  "red"  )
      (xD  yD  "."  15  "green")
      (xC  yC  "C"  10  ""     )
      (xA  yA  "A"  10  ""     )
      (xE  yE  "E"  10  ""     )
    if t = tstart
        path := (xD  yD  "."  10  "green")
    else
        path := stack(path, (xD  yD  "."  10  "green"))
```



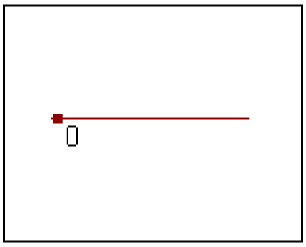
Создание анимации в SMath Studio.Часть 2

В этой части пособия для расчета положений плоских тел применен метод преобразования координат.Составлена программа-функция такого преобразования.Использование этой программы демонстрируется при анимации как отдельных тел,так и механизмов:кривошипно-шатунного, кулисного,с качающимся цилиндром и механизмом прерывистого движения

3.Анимация методом преобразования координат

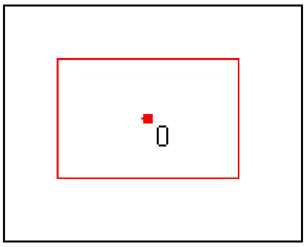
Плоское тело можно представить в виде набора узловых точек с заданными координатами,соединенных отрезками или кривыми.Координаты узловых точек представим в виде матрицы размерностью n x 2,где n-число узловых точек.Приведем матрицы плоских тел,которые будут использованы в примерах:

1.Линия



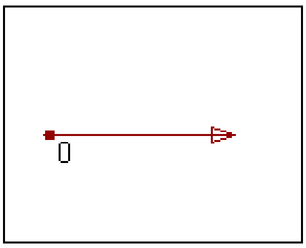
$$Line := \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix}^{T}$$

2.Прямоугольник



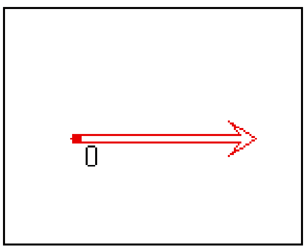
$$restangle := \begin{pmatrix} -1.5 & 1.5 & 1.5 & -1.5 & -1.5 \\ -1 & -1 & 1 & 1 & -1 \end{pmatrix}^{T}$$

3.Стрелка



$$arrow1 := \begin{pmatrix} 0 & 0.8 & 0.7 & 0.7 & 0.8 \\ 0 & 0 & 0.03 & -0.03 & 0 \end{pmatrix}^{T}$$

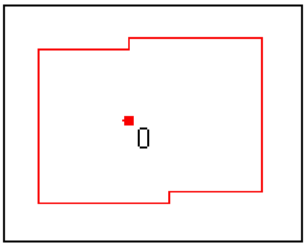
4.Двойная стрелка



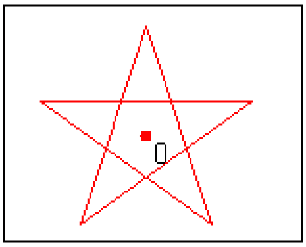
$$arrow2 := \begin{pmatrix} 0 & 0.736 & 0.68 & 0.8 & 0.68 & 0.736 & 0 & 0 \\ 0.016 & 0.016 & 0.08 & 0 & -0.08 & -0.016 & -0.016 & 0.016 \end{pmatrix}^{T}$$

5.Короб

$$korob := \begin{pmatrix} -100 & 0 & 0 & 145.0332 & 145.0332 & 45.0332 & 45.0332 & -100 & -100 \\ 78 & 78 & 90.0666 & 90.0666 & -78 & -78 & -90.0666 & -90.0666 & 78 \end{pmatrix}^{T}$$



6.Звезда



$$star := \begin{pmatrix} 0.5 & 0 & -0.5 & 0.809 & -0.809 & 0.5 \\ -0.6882 & 0.8507 & -0.6882 & 0.2629 & 0.2629 & -0.6882 \end{pmatrix}^{T}$$

Изменение положения тела и перенос его в заданную точку проведем с помощью трех процедур:
1.Параллельный перенос тела для совмещения начала координат с заданной точкой
2.Поворот тела на заданный угол относительно начала координат
3.Изменение размеров в соответствии с выбранным масштабом.
Эти процедуры выполняются с помощью вспомогательной программы Trans:

$$Trans(x, y, angle, m, Body) := \left| \begin{array}{l} B := matrix(rows(Body), 1) \\ Ro := augment(B + x, B + y) \\ \text{"Матрица поворота тела на заданный угол"} \\ \Omega := eval\left(\begin{pmatrix} \cos(angle) & \sin(angle) \\ -\sin(angle) & \cos(angle) \end{pmatrix}\right) \\ Ro + m \cdot Body \cdot \Omega \end{array} \right.$$

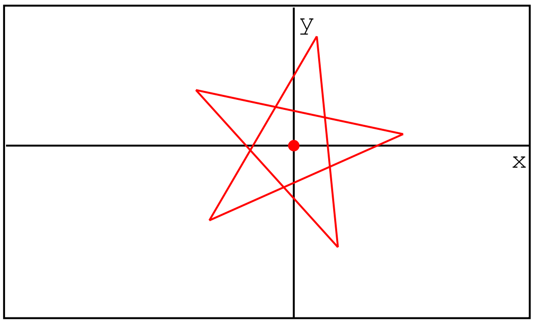
Аргументы программы:
x,y-координаты точки,в которую переносится начало координат
angle-угол поворота объекта
m-масштабный коэффициент
Body-матрица координат узловых точек.

Пример3.1 С помощью функции Trans(,,,,star)изобразить вращение звезды

$$\Delta\theta := \frac{\pi}{36}$$

$$step := 0 \,..\, 71$$

```
star( t):= ⎧ (0  0  "."  15  "red" )
           ⎩ Trans( 0 , 0 , Δθ ·t , 0.5 , star )
```



Пример3.2 Заданы радиус и закон движения центра С катящегося диска.С помощью функции Trans(,,,,Line)изобразить радиус диска

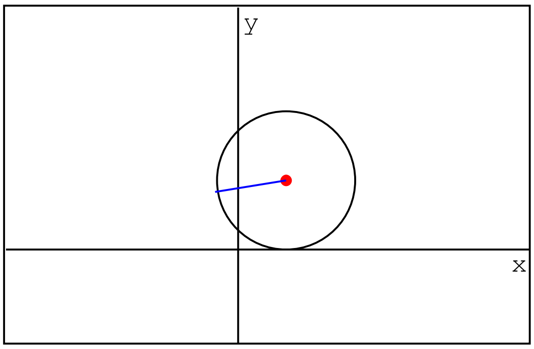
Исходные данные:

$R := 2$

$xC(t) := eval(8 - R \cdot \Delta\theta \cdot t)$ $yC(t) := R$

$step := 0 .. 72$

```
DiskLine( t):= "Δθ*t-угол радиуса с осью x"
               ⎧ Trans( xC( t) , yC( t) , Δθ ·t , 1 , Line )
               ⎨ ⎛ xC( t) yC( t) "o"  120  ""    ⎞
               ⎩ ⎝ xC( t) yC( t) "."  15  "red" ⎠
```



Пример3.3 Диск преодолевает препятствие.Закон движения центра С диска кусочная функция.Изобразить траекторию центра С и радиус диска

Исходные данные:

$H := 150$ `высота ступеньки` $R := 50$

$\Delta\theta := \frac{\pi}{24}$ `шаг изменения угла` $\Delta S := \Delta\theta \cdot R$ `Перемещение за один шаг`

$h := H - R$ `Высота подъема`

Траектория центра диска состоит из трех участков:вертикального,дуги окружности с центром в угловой точке и горизонтального.Каждому участку соответствует свой интервал времени:

$\tau_1 := \frac{h}{\Delta S}$ Интервал времени вертикального движения

$\tau_2 := \frac{\pi}{2 \cdot \Delta\theta}$ Интервал времени движения по дуге окружности

$tend - \tau_1 - \tau_2$ Интервал времени горизонтального движения

На каждом из трех участков закон движения и цвет траектории задан разными выражениями:

$$rC(t) := eval\left(\begin{array}{l} \text{if } t \le \tau_1 \\ \quad \begin{pmatrix} -R \\ R + \Delta S \cdot t \\ \text{"green"} \end{pmatrix} \\ \text{else} \\ \quad \text{if } t > \tau_1 + \tau_2 \\ \qquad \begin{pmatrix} \Delta S \cdot \left(t - \left(\tau_1 + \tau_2\right)\right) \\ H + R \\ \text{""} \end{pmatrix} \\ \quad \text{else} \\ \qquad \begin{pmatrix} -R \cdot \cos\left(\Delta\theta \cdot \left(t - \tau_1\right)\right) \\ H + R \cdot \sin\left(\Delta\theta \cdot \left(t - \tau_1\right)\right) \\ \text{"Red"} \end{pmatrix} \end{array}\right)$$

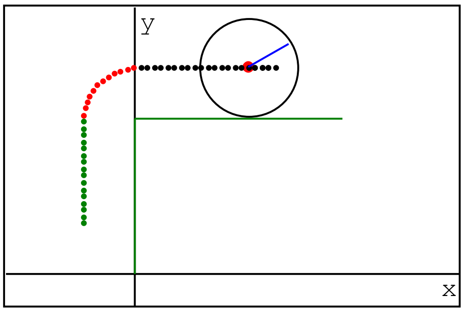
Уравнения движения центра диска:

$xC(t) := eval\left(rC(t)_1\right)$ $\qquad$ $yC(t) := eval\left(rC(t)_2\right)$

$tstart := 0$ $\qquad$ $tend := 48$

$step := tstart\ ..\ 2 \cdot tend - tstart$

$$f1(t, path) := \left| \begin{array}{l} t := t \cdot (t \le tend) + (2 \cdot tend - t) \cdot (t > tend) \\ \text{"радиус диска"} \\ \left\{ \begin{array}{l} Trans\left(xC(t), yC(t), -\Delta\theta \cdot t, 23, Line\right) \\ \text{if } t = tstart \\ \quad path := eval\left(\begin{pmatrix} xC(t) & yC(t) & \text{"."} & 9 & \text{"green"} \end{pmatrix}\right) \\ \text{else} \\ \quad path := stack\left(path, \begin{pmatrix} xC(t) & yC(t) & \text{"."} & 9 & rC(t)_3 \end{pmatrix}\right) \\ \begin{pmatrix} 0 & 0 \\ 0 & H \\ 4 \cdot R & H \end{pmatrix} \\ \begin{pmatrix} xC(t) & yC(t) & \text{"o"} & 85 & \text{""} \\ xC(t) & yC(t) & \text{"."} & 15 & \text{"red"} \end{pmatrix} \end{array} \right. \end{array} \right.$$



Пример3.4 Законы движения центра прямоугольника и его угол поворота - кусочные функции.С помощью функции Trans(,,,,restangle)изобразить прямоугольник

$$\Delta\theta := \frac{\pi}{36}$$

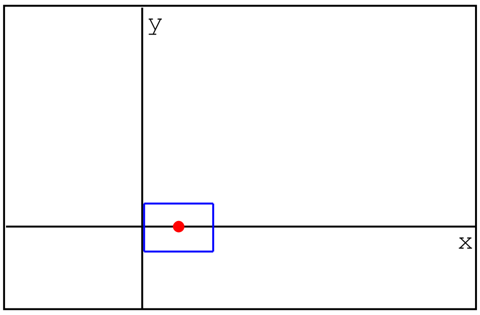
$$x(t) := 0.08 \cdot t \cdot (t \le 20) + 0.08 \cdot 20 \cdot (t > 20)$$

$$y(t) := 0.08 \cdot (t - 38) \cdot (t > 38)$$

$$\varphi(t) := \Delta\theta \cdot (t - 20) \cdot ((t > 20) \wedge (t \le 38)) + \Delta\theta \cdot 18 \cdot (t > 38)$$

step := 0 .. 50

$$\text{restangle}(t) := \left\| \begin{cases} \text{Trans}(x(t), y(t), \varphi(t), 0.2, \text{restangle}) \\ (x(t)\ y(t)\ "."\ 15\ "red") \end{cases} \right.$$



Пример3.5 Задан закон движения точки.С помощью функций Trans(,,,,arrow1) Trans(,,,,arrow2) изобразить вектор скорости и его составляющие

$$x(t) := \text{eval}(3 \cdot \cos(\Delta\theta \cdot t))$$

$$y(t) := \text{eval}(2 \cdot \sin(\Delta\theta \cdot t))$$

Найдем проекции вектора скорости точки и его модуль

$$Vx(t) := \text{eval}\left(\frac{d}{d\,t}(3 \cdot \cos(\Delta\theta \cdot t))\right)$$

$$Vy(t) := \mathrm{eval}\left(\frac{d}{d\,t}\left(2\cdot\sin(\Delta\theta\cdot t)\right)\right)$$

$$V(t) := \mathrm{eval}\left(\sqrt{Vx(t)^2 + Vy(t)^2}\right)$$

Для расчета направления вектора V(t) используем функцию xy2pol

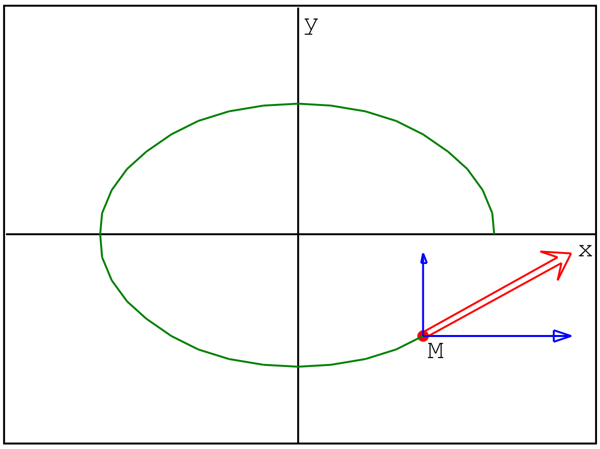
$$\varphi(t) := \mathrm{eval}\left(\mathrm{xy2pol}\left(Vx(t), Vy(t)\right)_2\right)$$

$$\Delta\theta := \frac{\pi}{18}$$

$$step := 0\,..\,36$$

```
v(t, path) :=
    V1 := Trans(x(t), y(t), 0, 7·Vx(t), arrow1)
    V2 := Trans(x(t), y(t), π/2, 7·Vy(t), arrow1)
    ⎧ stack(V1, V2)
    ⎪ Trans(x(t), y(t), φ(t), 7·V(t), arrow2)
    ⎪ if t = 0
    ⎨    path := (x(t) y(t))
    ⎪ else
    ⎪    path := stack(path, (x(t) y(t)))
    ⎩ ( x(t) y(t) "M" 10 ""
        x(t) y(t) "." 15 "Red" )
```

Вектор скорости и его составляющие



Примечание: Масштабные коэффициенты выбраны пропорционально модулю вектора и модулям его составляющих.Благодаря этому длины векторов изменяются в соответствии с их величинами в данный момент времени.

Пример3.6 Механизм с качающимся цилиндром.Цилиндр и поршень изобразить с помощью функций Trans(,,,,restangle)и Trans(,,,,Line). Для расчета направления шатуна использовать функцию xy2pol

Исходные данные:

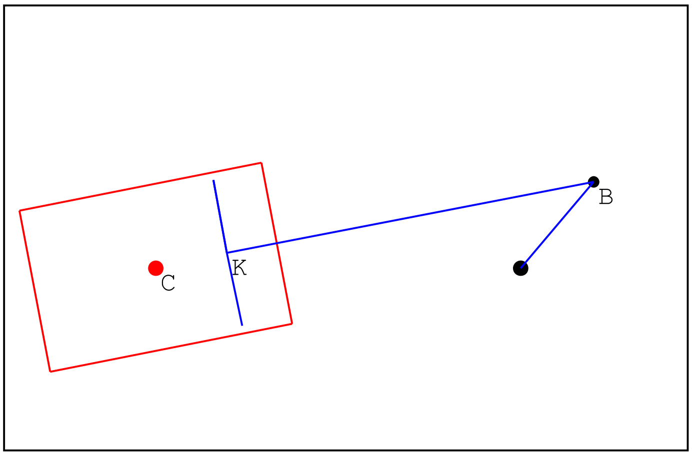
$OB := 3 \qquad BK := 10$

$xC := -9.8 \qquad yC := 0$

$\Delta\theta := \dfrac{\pi}{36}$

$step := 0 .. 71$

$$
f(t) := \left|
\begin{array}{l}
xB := \mathrm{eval}\left(OB \cdot \sin\left(\Delta\theta \cdot t\right)\right) \\
yB := \mathrm{eval}\left(OB \cdot \cos\left(\Delta\theta \cdot t\right)\right) \\
\text{"φ-угол шатуна с осью x"} \\
\varphi := \mathrm{xy2pol}\left(xB - xC, yB - yC\right)_{2} \\
xK := \mathrm{eval}\left(xB - BK \cdot \cos\left(\varphi\right)\right) \\
yK := \mathrm{eval}\left(yB - BK \cdot \sin\left(\varphi\right)\right) \\
mech1 := \begin{pmatrix} 0 & 0 \\ xB & yB \\ xK & yK \end{pmatrix} \\
L1 := \mathrm{Trans}\left(xK, yK, \varphi + \dfrac{\pi}{2}, 1, \mathrm{Line}\right) \\
L2 := \mathrm{Trans}\left(xK, yK, \varphi + \dfrac{3 \cdot \pi}{2}, 1, \mathrm{Line}\right) \\
\left\{
\begin{array}{l}
\mathrm{stack}\left(mech1, L1, L2\right) \\
\mathrm{Trans}\left(xC, yC, \varphi, 2.2, \mathrm{restangle}\right) \\
\begin{pmatrix}
xK & yK & \text{"K"} & 10 & \text{""} \\
0 & 0 & \text{"."} & 20 & \text{""} \\
xB & yB & \text{"."} & 15 & \text{""} \\
xB & yB & \text{"B"} & 10 & \text{""} \\
xC & yC & \text{"."} & 20 & \text{"Red"} \\
xC & yC & \text{"C"} & 10 & \text{""}
\end{pmatrix}
\end{array}
\right.
\end{array}
\right.
$$



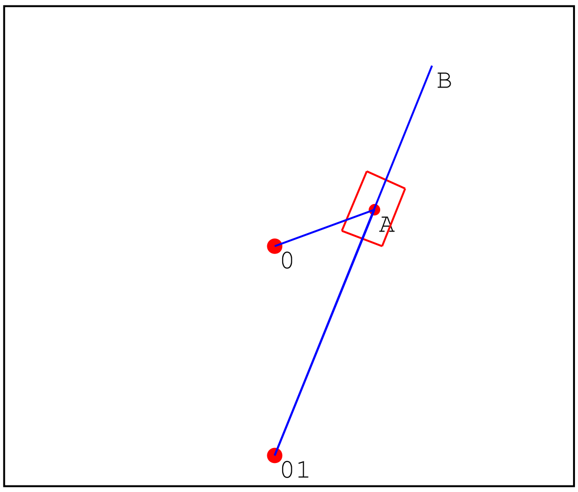
Пример3.7 Кривошипно-кулисный механизм .Для расчета направления кулисы и координат конца кулисы использовать соответственно функции xy2pol и pol2xy

Исходные данные:

```
OA := 1                  O1B := 4

x0 := 0                  y0 := 0

x01 := 0                 y01 := -2
```

$$\Delta\theta := \frac{\pi}{36}$$

```
step := 0 .. 71

krivkul(t) := | xA := eval(OA·cos(Δθ·t))
              | yA := eval(OA·sin(Δθ·t))
              | "φ-угол кулисы с осью x"
              | φ := xy2pol(xA - x01 , yA - y01)_2
              | xB := eval(x01 + Re(pol2xy(O1B , φ))_1)
              | yB := eval(y01 + Re(pol2xy(O1B , φ))_2)
              | { (  0    0  )
              |   (  xA   yA )
              |   ( x01  y01 )
              |   (  xB   yB )
              |   Trans(xA , yA , φ , 0.2 , restangle)
              |   (  0    0   "."  20  "red" )
              |   (  xA   yA  "."  15  "red" )
              |   ( x01  y01  "."  20  "red" )
              |   (  0    0   "0"   10  "" )
              |   (  xA   yA  "A"   10  "" )
              |   (  xB   yB  "B"   10  "" )
              |   ( x01  y01  "01"  10  "" ) }
```



Пример3.8 Механизм прерывистого движения.Короб изобразить с помощью функций Trans(,,,,korob)

Координаты концов двух лопастей находим с помощью программы Pintersect, которая уже использовалась в первой части пособия:

Координаты точек пересечения двух окружностей

$$\mathrm{PIntersect}(x0, y0, r0, x1, y1, r1) := \left|\begin{array}{l} x10 := x1 - x0 \\ y10 := y1 - y0 \\ d := \mathrm{eval}\left(\sqrt{x10^2 + y10^2}\right) \\ a := \mathrm{eval}\left(\dfrac{r0^2 - r1^2 + d^2}{2 \cdot d}\right) \\ h := \mathrm{eval}\left(\sqrt{r0^2 - a^2}\right) \\ x2 := \mathrm{eval}\left(x0 + \dfrac{a}{d} \cdot x10\right) \\ y2 := \mathrm{eval}\left(y0 + \dfrac{a}{d} \cdot y10\right) \\ P3 := \begin{pmatrix} x2 + \dfrac{h}{d} \cdot y10 & y2 - \dfrac{h}{d} \cdot x10 \\ x2 - \dfrac{h}{d} \cdot y10 & y2 + \dfrac{h}{d} \cdot x10 \end{pmatrix} \end{array}\right.$$

Исходные данные:

$L := 90.0666$ длина лопасти

$xB := L \qquad yB := 0$

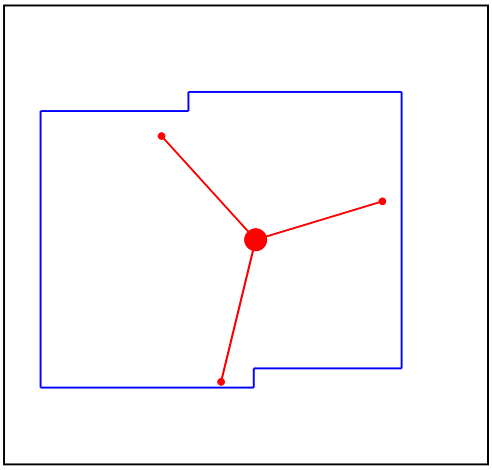
$BC := 2 \cdot L \cdot \cos\left(\dfrac{\pi}{6}\right)$ расстояние между концами лопастей

$\Delta := 0.5 \cdot L$ максимальное перемещение

$$\Delta\theta := \frac{\pi}{36}$$

$step := 0 .. 24$

$$\mathrm{index}(t):=\left|\begin{array}{l}
\text{"Координаты конца первой лопасти "}\\
yB:=\mathrm{eval}(L\cdot\sin(\Delta\theta\cdot t))\\
xB:=\mathrm{eval}(L\cdot\cos(\Delta\theta\cdot t))\\
\text{"Координаты концов остальных лопастей "}\\
C:=\mathrm{eval}(\mathrm{PIntersect}(xB, yB, BC, 0, 0, L))\\
xC:=C_{2\ 1}\\
yC:=C_{2\ 2}\\
xF:=C_{1\ 1}\\
yF:=C_{1\ 2}\\
\text{"Зависимость перемещение короба от времени"}\\
x:=\mathrm{eval}((-\Delta)\cdot(t\le 6)+(xC-\Delta)\cdot((t>6)\wedge(t\le 12))+xB\cdot(t>18))\\
\left\{\begin{array}{l}
\mathrm{Trans}(x, 0, 0, 1, korob)\\
\begin{pmatrix} xB & yB\\ 0 & 0\\ xC & yC\\ 0 & 0\\ xF & yF\end{pmatrix}\\
\begin{pmatrix} 0 & 0 & \text{"."} & 30 & \text{"red"}\\ xB & yB & \text{"."} & 10 & \text{"red"}\\ xC & yC & \text{"."} & 10 & \text{"red"}\\ xF & yF & \text{"."} & 10 & \text{"red"}\end{pmatrix}
\end{array}\right.
\end{array}\right.$$



Пример3.9 Изобразить кривошипно-шатунный механизм ,используя геометрический метод

Исходные данные:

$OA := 1$ $AB := 2.5$

$L1x := -2.5$ $L1y := -2.5$

$L2x := 4$ $L2y := 0$

$\varphi := 3.5004$

Покажем применение геометрического метода для нахождения координат точки B.
Точка A кривошипа OA описывает окружность. Точка B описывает дугу окружности радиуса AB в ее относительном движении вокруг точки A и движется по направляющей ползуна.Точка B лежит на пересечении окружности и направляющей. Координаты точки B находим с помощью вспомогательной программы LCintersect

Аргументы программы:
xp1,yp1 и xp2,yp2-координаты концов отрезка прямой
xC,yC,R-координаты центра окружности и ее радиус

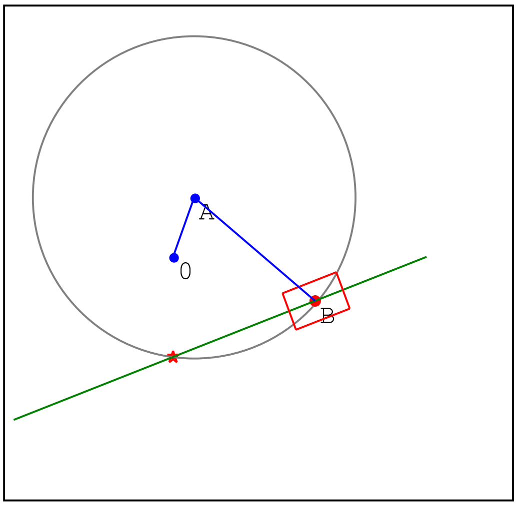
Координаты точек пересечения окружности и прямой

$$LCintersect(p1x, p1y, p2x, p2y, xC, yC, R) := \left| \begin{array}{l} \varepsilon := 10^{-15} \\ dx := eval(p2x - p1x + \varepsilon) \\ dy := eval(p2y - p1y + \varepsilon) \\ A := dx^2 + dy^2 \\ B := 2 \cdot (dx \cdot (p1x - xC) + dy \cdot (p1y - yC)) \\ C := (p1x - xC)^2 + (p1y - yC)^2 - R^2 \\ sqDet := eval\left(\sqrt{B^2 - 4 \cdot A \cdot C}\right) \\ t := \begin{pmatrix} \dfrac{-B + sqDet}{2 \cdot A} \\ \dfrac{-B - sqDet}{2 \cdot A} \end{pmatrix} \\ x := p1x + t \cdot dx \\ y := p1y + t \cdot dy \\ augment(x, y) \end{array} \right.$$

$\Delta\theta := \dfrac{\pi}{36}$

$step := 0 .. 72$

```
Gearing( t):= | xA := eval( OA·cos( Δθ·t))
              | yA := eval( OA·sin( Δθ·t))
              | "Координаты точки B"
              | pB := LCintersect( L1x , L1y , L2x , L2y , xA , yA , AB)
              | xB := eval( pB    )
              |              1 1
              | yB := eval( pB    )
              |              1 2
              | { ⎛ 0   0 ⎞
              | { ⎜ xA  yA⎟
              | { ⎝ xB  yB⎠
              | { Trans( xB , yB , φ , 0.3 , restangle )
              | { ⎛L1x   L1y    ⎞
              | { ⎝L2x  L2y := 0⎠
              | { ⎛ 0   0  "."  13 "blue"⎞
              | { ⎜ xA  yA "."  13 "blue"⎟
              | { ⎝ xB  yB "."  15  "red"⎠
              | { ⎛ 0   0  "0"   10   ""   ⎞
              | { ⎜ xA  yA "A"   10   ""   ⎟
              | { ⎜ xB  yB "B"   10   ""   ⎟
              | { ⎝ xA  yA "o"  280 "gray" ⎠
              | { "Вторая точка пересечения"
              | { ( pB     pB     "*"  12 "red" )
              |       2 1    2 2
```



Примечание:На анимации видно,что окружность и прямая пересекаются в двух точках - B и точке,обозначенной красной звездочкой.Из двух точек выбираем одну,которая подходит к нашему механизму,т.е. точку B.