Отчёт по лабораторной работе №2

Задача о погоне

Гайсина Алина Ринатовна

Содержание

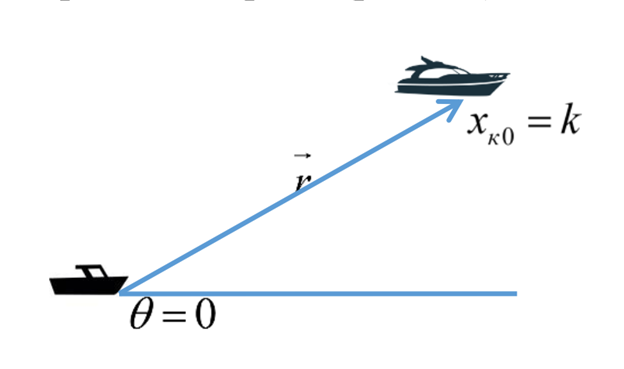
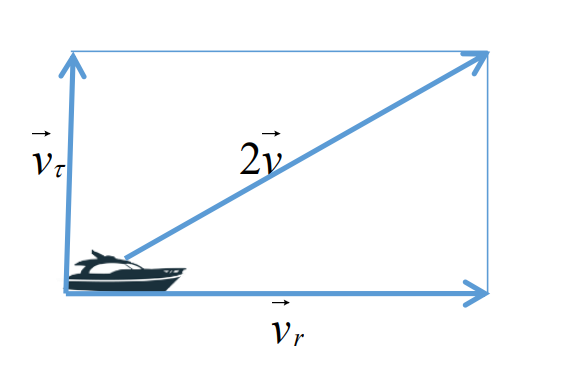
# Цель работы

Изучить основы языка программирования Julia: библиотеки языка, которые используются для построения графиков и решения дифференциальных уравнений. Решить задачу о погоне.

# Задание (Вариант№35)

На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии 18 км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в 4,9 раза больше скорости браконьерской лодки. ## Постановка задачи 1. Запишите уравнение, описывающее движение катера, с начальными условиями для двух случаев (в зависимости от расположения катера относительно лодки в начальный момент времени). 2. Постройте траекторию движения катера и лодки для двух случаев. 3. Найдите точку пересечения траектории катера и лодки.

# Теоретическое введение

1. Принимаем за - место нахождения лодки браконьеров в момент обнаружения, - место нахождения катера береговой охраны относительно лодки браконьеров в момент обнаружения лодки.
2. Введем полярные координаты. Считаем, что полюс - это точка обнаружения лодки браконьеров, а полярная ось *r* проходит через точку нахождения катера береговой охраны (Рис. [-@fig:001]) 
3. Траектория катера должна быть такой, чтобы и катер, и лодка все время были на одном расстоянии от полюса *tetha*, только в этом случае траектория катера пересечется с траекторией лодки. Поэтому для начала катер береговой охраны должен двигаться некоторое время прямолинейно, пока не окажется на том же расстоянии от полюса, что и лодка браконьеров. После этого катер береговой охраны должен двигаться вокруг полюса удаляясь от него с той же скоростью, что и лодка браконьеров.
4. Чтобы найти расстояние *x* (расстояние после которого катер начнет двигаться вокруг полюса), необходимо составить простое уравнение. Пусть через время *t* катер и лодка окажутся на одном расстоянии *x* от полюса. За это время лодка пройдет *x*, а катер (или , в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как или ( - во втором случае). Так как время одно и то же, то эти величины одинаковы. Тогда неизвестное расстояние *x* можно найти из одного из следующих уравнений: или . Отсюда мы найдём значения и , задача решается для этих двух случаев.
5. После того, как катер береговой охраны окажется на одном расстоянии от полюса, что и лодка, он должен сменить прямолинейную траекторию и начать двигаться вокруг полюса удаляясь от него со скоростью лодки *v*. Для этого скорость катера раскладываем на две составляющие:радиальная скорость и тангенциальная скорость (Рис. [-@fig:002]).  .
6. Решение исходной задачи сводится к решению системы из двух дифференциальных уравнений:
7. . Проводя аналогичные рассуждения необходимо решить задачу о погоне для своего варианта (Вариант №35).

# Выполнение лабораторной работы

## Программный код на Julia.

using DifferentialEquations  
using Plots  
  
k = 18  
n = 4.9  
function dr(u, p, t)  
 return u/sqrt(n\*n - 1)  
end   
  
# Первый случай  
r0 = k/(n + 1)  
tetha = (0, 2\*pi)  
  
prob = ODEProblem(dr, r0, tetha)  
sol = solve(prob, abstol = 1e-8, reltol = 1e-8)  
  
r\_ang = [sol.t[rand(1:size(sol.t)[1])] for i in 1:size(sol.t)[1]]  
  
plt = plot(proj=:polar, aspect\_ratio=:equal, title="Первый случай задачи о погоне", legend=:outerbottom)  
plot!(plt, [r\_ang[1], r\_ang[2]], [0.0, sol.u[size(sol.u)[1]]], label="Путь лодки")  
plot!(plt, sol.t, sol.u, label="Путь катера", color=:pink)  
savefig("2\_1.png")  
  
# Второй случай  
r0 = k/(n - 1)  
tetha = (-pi, pi)  
  
prob = ODEProblem(dr, r0, tetha)  
sol = solve(prob, abstol=1e-8, reltol=1e-8)  
r\_ang = [sol.t[rand(1:size(sol.t)[1])] for i in 1:size(sol.t)[1]]  
  
plt = plot(proj=:polar, aspect\_ratio=:equal, title="Второй случай задачи о погоне", legend=:outerbottom)  
plot!(plt, [r\_ang[1], r\_ang[2]], [0.0, sol.u[size(sol.u)[1]]], label="Путь лодки")  
plot!(plt, sol.t, sol.u, label="Путь катера", color=:pink)  
savefig("2\_2.png")

## Результат выполнения программы

1. Построила траекторию движения катера и лодки для первого случая (Рис. [-@fig:003]). 
2. Построила траекторию движения катера и лодки для второго случая (Рис. [-@fig:004]). 

# Выводы

Были изучены основы языка программирования Julia. Решена задача о погоне.

# Список литературы. Библиография

[1] Документация по Julia: https://docs.julialang.org/en/v1/ [2] Решение дифференциальных уравнений: https://www.wolframalpha.com/