Отчёт по лабораторной работе №2

Математическое моделирование

Чекалова Лилия Руслановна

Содержание

# Цель работы

* Построение математической модели для выбора правильной стратегии при решении задач поиска, в частности, задачи о погоне
* Визуализация полученной модели с помощью средств языков Julia и OpenModelica

# Задание

* Провести вывод дифференциальных уравнений в соответствии с заданными условиями
* Построить траекторию движения катера и лодки для двух случаев
* Определить по графику точку пересечения катера и лодки

# Теоретическое введение

Будем рассматривать задачу следующего содержания: На море в тумане катер береговой охраны преследует лодку браконьеров. Через определенный промежуток времени туман рассеивается, и лодка обнаруживается на расстоянии k км от катера. Затем лодка снова скрывается в тумане и уходит прямолинейно в неизвестном направлении. Известно, что скорость катера в n раза больше скорости браконьерской лодки. Необходимо определить по какой траектории необходимо двигаться катеру, чтобы нагнать лодку.

В варианте 65 k=18.4, n=4.6.

Полагаем t0 = 0, xл0 = 0 - место нахождения лодки браконьеров в момент обнаружения, xк0 = k - место нахождения катера береговой охраны относительно лодки браконьеров в момент обнаружения лодки.

Вводим полярные координаты. Считаем, что полюс - это точка обнаружения лодки браконьеров xл0 (theta = xл0 = 0), а полярная ось r проходит через точку нахождения катера береговой охраны.

Траектория катера должна быть такой, чтобы и катер, и лодка все время были на одном расстоянии от полюса theta, 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.

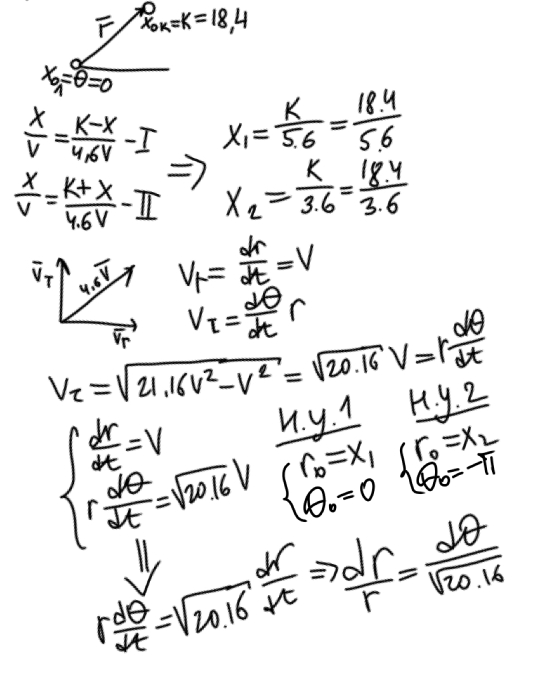
Чтобы найти расстояние x (расстояние после которого катер начнет двигаться вокруг полюса), необходимо составить простое уравнение. Пусть через время t катер и лодка окажутся на одном расстоянии x от полюса. За это время лодка пройдет x, а катер k-x (или k+x, в зависимости от начального положения катера относительно полюса). Время, за которое они пройдут это расстояние, вычисляется как x/v или (k-x)/(nv) (во втором случае (k+x)/(nv)). Так как время одно и то же, то эти величины одинаковы. Тогда неизвестное расстояние x можно найти из следующего уравнения:

x/v = (k-x)/(nv) в первом случае или x/v = (k+x)/(nv) во втором. Отсюда находим x1 и x2.

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: vr - радиальная скорость и vt - тангенциальная скорость. Радиальная скорость - это скорость, с которой катер удаляется от полюса, vr = dr/dt = v, так как нам нужно, чтобы эта скорость была равна скорости лодки.

Тангенциальная скорость – это линейная скорость вращения катера относительно полюса, vt = r(dtheta/dt).

Мы вычисляем тангенциальную скорость, получаем систему дифференциальных уравнений с начальными условиями и в результате преобразований получаем уравнение (рис. @fig:001). Решив это уравнение мы получим траекторию движения катера в полярных координатах.

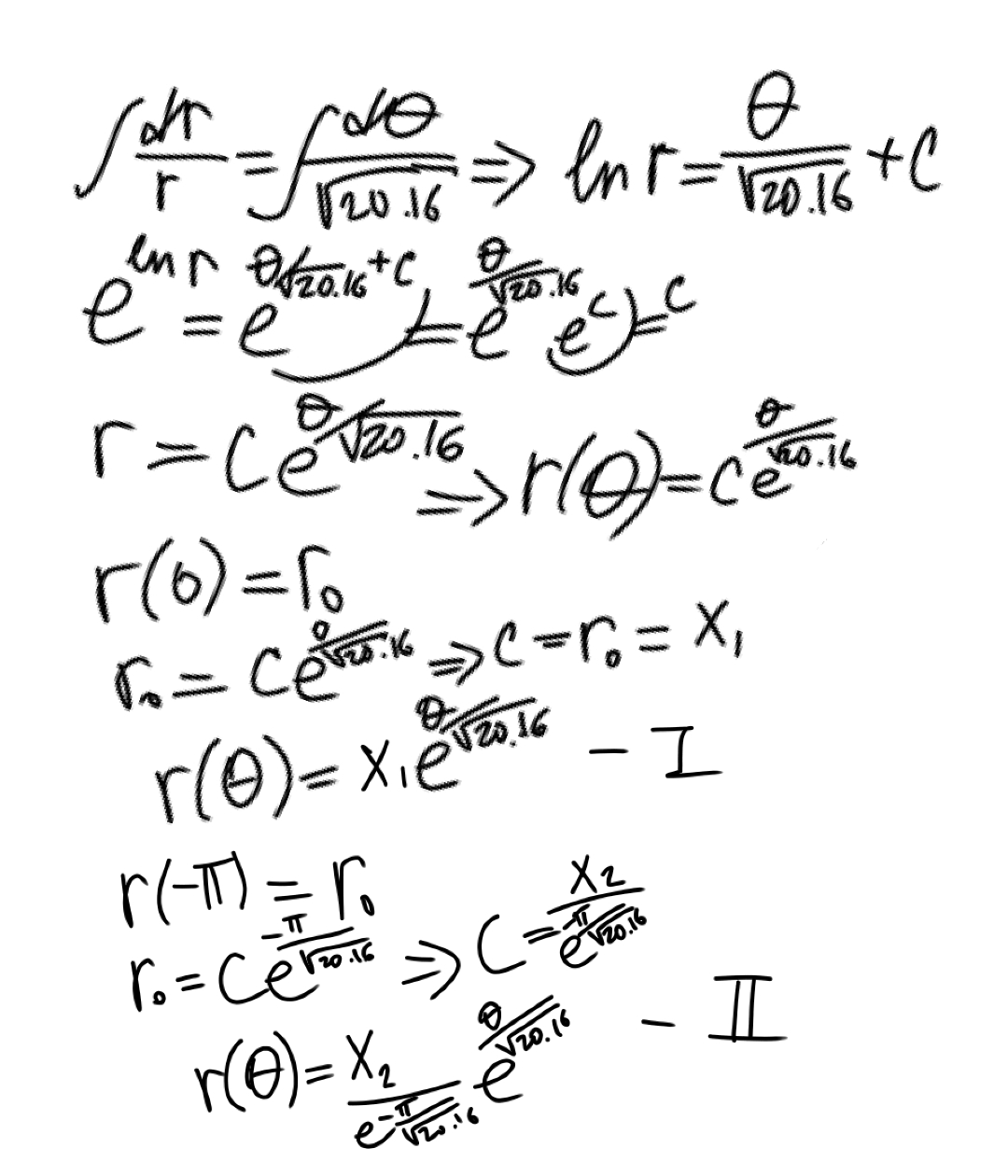


Выведение итогового уравнения из рассуждений

Более подробно см. в [1].

# Выполнение лабораторной работы

Решаем выведенное нами ранее дифференциальное уравнение с разделяющими переменными и подставляем начальные условия, чтобы вычислить константу для каждого случая (рис. @fig:002).



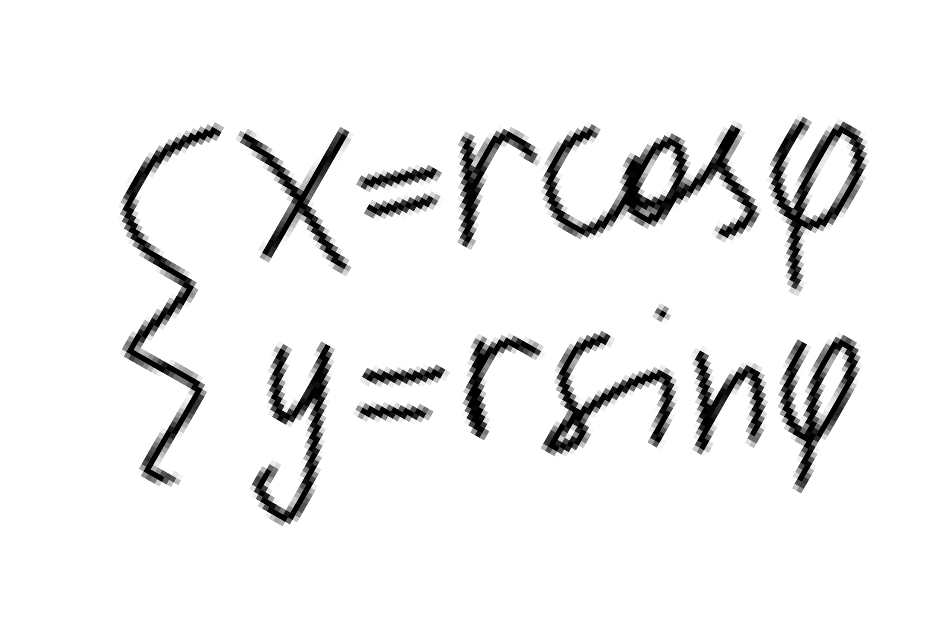
Решение уравнения

Используя библиотеку Plots, пишем программу на Julia (рис. @fig:003).



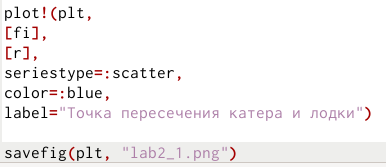
Программа на Julia

В начале программы указываем начальные условия, затем описываем функцией решенное нами уравнение и передаем в нее данные для построения графика. Для нахождения точки пересечения пользуемся формулами перевода из полярной системы координат в декартовую (рис. @fig:004).



Формулы для перевода из полярной системы координат

Рисуем графики, находим точку пересечения и сохраняем полученное изображение в файл (рис. @fig:005).



Нахождение точки пересечения и сохранение в файл

Исследовав график (рис. @fig:006), получаем, что для первого случая координаты точки пересечения равны (-3.9, 3.9).



График Julia, первый случай

Для второго случая меняем входные данные для нашей функции, описывающей уравнение (рис. @fig:007).

Замена входных данных

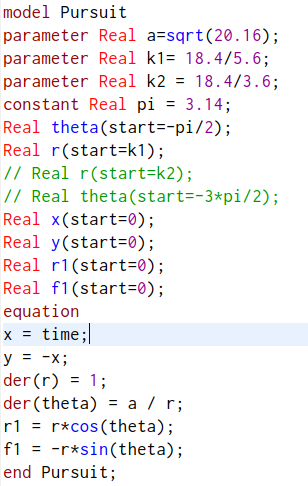
Замена входных данных

Рассмотрев график для второго случая (рис. @fig:008), получаем координаты точки пересечения, равные (-12.3, 12.3).



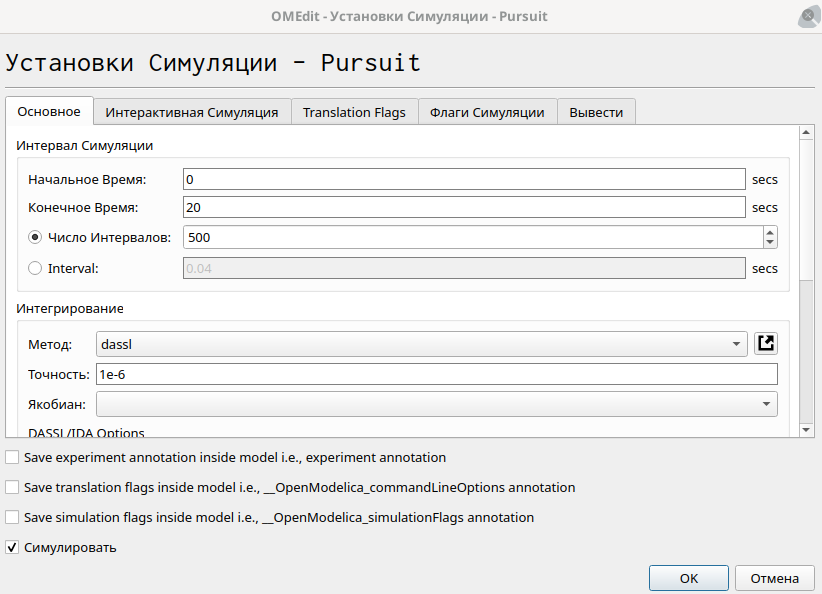
График Julia, второй случай

Далее создаем модель на OpenModelica (рис. @fig:009). Задаем необходимые начальные условия в виде параметров, а нужные нам уравнения указываем в разделе equation, в том числе уравнения перевода из полярной системы координат в декартовую.

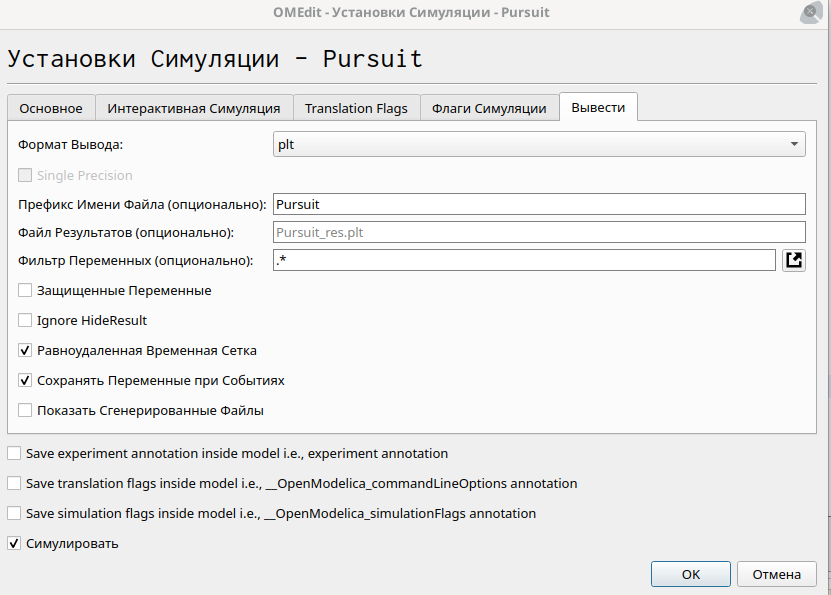


Программа на OpenModelica

Настраиваем параметры симуляции, указываем начальное, конечное время и число интервалов (рис. @fig:010) и формат вывода (рис. @fig:011).

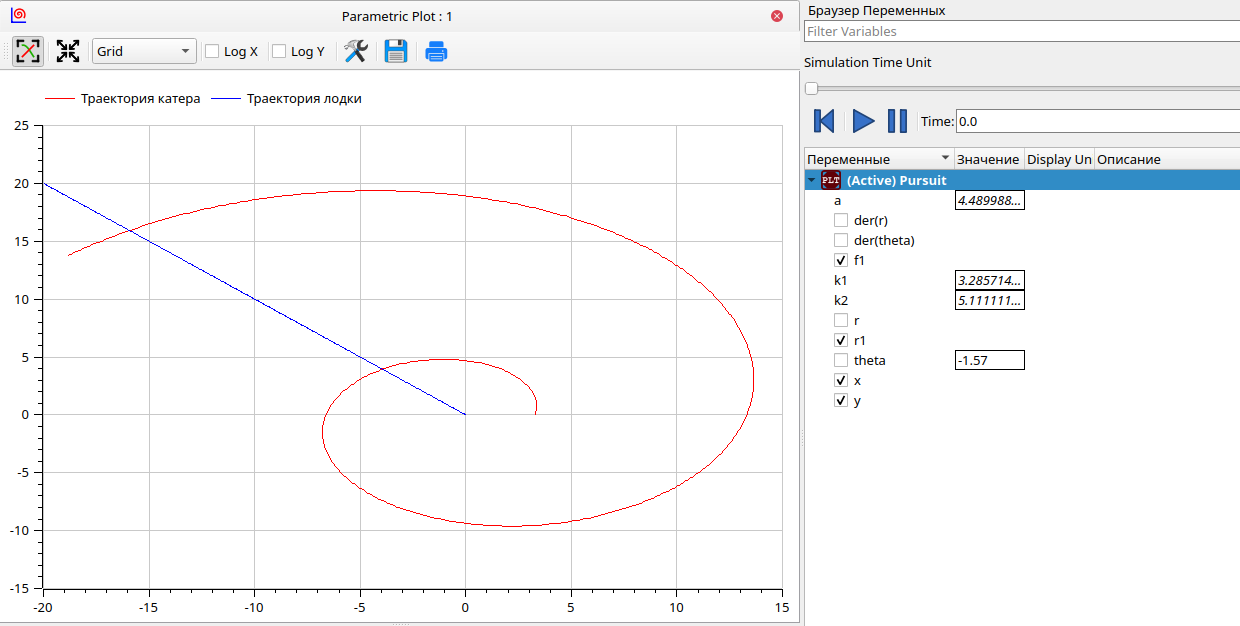


Настройка симуляции, ч.1



Настройка симуляции, ч.2

Выбираем график типа Parametric Plot и указываем необходимые нам оси, линии графика, настраиваем легенду (рис. @fig:012).



Настройка внешнего вида графика

Рассмотрев первый график (рис. @fig:013), получаем координаты точки пересечения, равные (-3.9, 3.9).

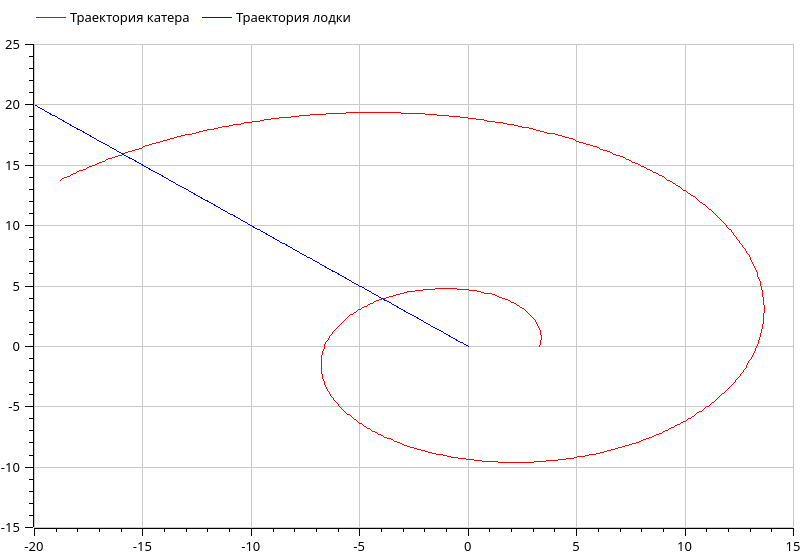


График OpenModelica, первый случай

Проанализировав второй график (рис. @fig:014), видим, что координаты точки пересечение равны (-12.3, 12.3).

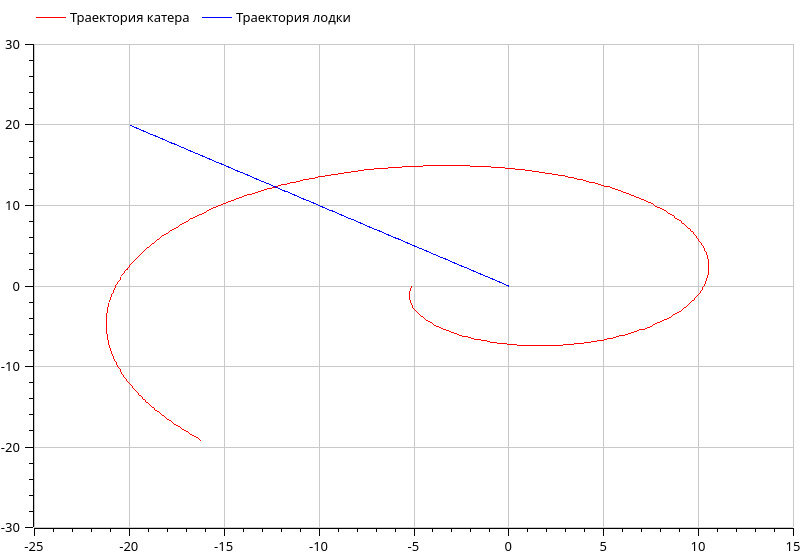


График OpenModelica, второй случай

# Сравнение языков

Julia является более интуитивно понятным языком и позволяет производить различные вычисления, не ограничиваясь рамками работы с визуализацией уравнений.

OpenModelica является более узконаправленным инструментом, ориентирующимся на создание сложных моделей с множеством уравнений в основе. Из-за этого работа с ним кажется труднее.

На результаты работы различия в подходах этих языков практически не повлияли, координаты точек пересечения в разных реализациях совпадают с точностью до одного знака после запятой.

# Выводы

В ходе работы были получены навыки построения математических моделей для решения задачи о погоне и визуализации их с помощью языков Julia и OpenModelica, а также закреплены знания, связанные с решением дифференциальных уравнений. Результатом работы стали графики, наглядно демонстрирующие решение задачи о погоне.

# Список литературы

1. Теоретические материалы к лабораторной работе “Задача о погоне”: https://esystem.rudn.ru/mod/resource/view.php?id=967233