

В. О. Дженжер, Л. В. Денисова,
Оренбургский государственный педагогический университет

МАТЕМАТИЧЕСКАЯ АНИМАЦИЯ В ШКОЛЬНОМ КОМПЬЮТЕРНОМ МОДЕЛИРОВАНИИ

Аннотация

В статье рассматривается задача компьютерного моделирования кардиоиды как огибающей семейства хорд. Используются новые возможности графического модуля языка программирования PascalABC.NET для создания анимации.

Ключевые слова: компьютерное моделирование, геометрическое моделирование, программирование.

DOI: 10.32517/2221-1993-2019-18-6-51-54

Содержательная линия «Моделирование и формализация» — одна из системообразующих в школьном курсе информатики. При изучении компьютерных математических моделей кажется уместным предложить учащимся ряд задач, связанных с графическими построениями математических кривых. Уверенное использование графических возможностей компьютера существенно облегчает в дальнейшем визуализацию построенных моделей.

В качестве основного средства разработки программы мы используем мощный современный язык программирования **PascalABC.NET** [4], разработанный в Южном федеральном университете, г. Ростов-на-Дону.

С одной стороны, этот язык содержит все необходимое для обучения школьника:

- удобный текстовый редактор с поддержкой отладчика, Intellisense (автодополнение кода, когда после начала набора кода редактор подсказывает возможные варианты завершения) и автоформатирования кода;
- встроенные исполнители Робот и Чертежник с автоматической проверкой решений и возможностью создания собственных заданий; команды исполнителей можно давать как на английском, так и на русском языках;
- встроенный модуль Programming Taskbook, содержащий множество задач по различным разделам программирования с возможностью автоматической проверки решений, в том числе задач ЕГЭ;

- графические модули GraphABC и GraphWPF (более современный, чем первый) для работы с двумерной графикой, а также модуль Graph3D для создания трехмерных моделей.

С другой стороны, PascalABC.NET — это современный язык, позволяющий отказаться от стандартов программирования тридцатилетней давности, навязываемых давно устаревшими Turbo Pascal и Free Pascal. Школьники с первых занятий могут использовать кортежи, автовыведение типов и другие средства современного стиля программирования, делающие программу короткой, читаемой и легко модифицируемой.

Ну и, наконец, PascalABC.NET — бесплатное и открытое программное обеспечение, что является немаловажным для школы и вуза.

Мы преподаем компьютерное моделирование для старших школьников (в основном это учащиеся X—XI классов) в центре физико-математического образования «Архимед» (г. Оренбург). На занятиях по компьютерному моделированию сначала предлагается ряд задач на построение графиков функций, заданных в декартовой и полярной системах координат, а также заданных параметрически. Наиболее интересными для ребят оказываются задачи, использующие анимацию. Познакомиться с математической анимацией можно, например, на сайте [3], который содержит огромное количество анимированных моделей, связанных с математикой и техникой. Нам кажется интересным не просто использовать готовые модели, но и самим создавать их.

Контактная информация

Дженжер Вадим Олегович, канд. физ.-мат. наук, доцент, зав. кафедрой информатики, физики и методики преподавания информатики и физики, Оренбургский государственный педагогический университет; *адрес:* 460014, г. Оренбург, ул. Советская, д. 19; *e-mail:* vdjenjer@yandex.ru

Денисова Людмила Викторовна, канд. пед. наук, доцент кафедры информатики, физики и методики преподавания информатики и физики, Оренбургский государственный педагогический университет; *адрес:* 460014, г. Оренбург, ул. Советская, д. 19; *e-mail:* lv-denisova@yandex.ru

V. O. Dzhenzher, L. V. Denisova,
Orenburg State Pedagogical University

MATHEMATICAL ANIMATION IN COMPUTER SIMULATION AT SCHOOL

Abstract

The article considers the task of computer simulation of cardioids as an envelope of the chord family. It is used the new graphics module features of PascalABC.NET to create animations.

Keywords: computer modeling, geometric modeling, programming.

Графические модули GraphWPF и Graph3D позволяют с легкостью анимировать картинку.

Здесь мы рассмотрим **задачу построения кардиоиды как огибающей семейства отрезков**.

Предполагается, что на предыдущих занятиях ученики уже познакомились с графическим модулем GraphWPF, а также с построением графиков функций, заданных в полярной системе координат. Будем использовать следующий алгоритм [2, 5]:

1. Нарисуем окружность, на которой разместим N точек, равноотстоящих друг от друга; точки пронумеруем так: $0, 1, \dots, N-1$.
2. Последовательно будем соединять отрезками точки: $0 \rightarrow 0, 1 \rightarrow 2, 2 \rightarrow 4, \dots, i \rightarrow (2 \cdot i) \bmod N$.
3. С ростом N будем наблюдать возникновение кардиоиды как огибающей семейства построенных хорд.

Например, для $N = 10$ получим картину, представленную на рисунке 1.

Глядя на приведенный рисунок, довольно просто понять алгоритм, но пока еще нельзя увидеть результат, потому что количество точек слишком мало и огибающая отрезков практически незаметна.

Увеличим количество точек до 100 (рис. 2).

В этом случае уже хорошо видно, что огибающей семейства построенных хорд является кардиоиды. Рассмотрим процедуру рисования подробнее (листинг 1).

Параметрами процедуры являются N (количество точек) и цвет хорды, который имеет значение по умолчанию *YellowGreen*. В строках 3–12 мы определяем параметры рисования, такие как цвет, размер и положение объектов. Затем в строках 15–29 размещаем N равноотстоящих точек на окружности и подписываем их номера. Для этого мы вычисляем положение точек в полярной системе координат, а затем переводим их в экранную декартову систему координат. В строках 33–34 мы рисуем хорды, соединяя точки i и $(2 \cdot i) \bmod N$.

Главная программа в этом случае будет очень простой (листинг 2).

Программа для построения одной картинке довольно проста. Однако хотелось бы сделать мультипликацию,

показывающую, как возникает огибающая семейства отрезков с ростом количества точек N .

Для этого в PascalABC.NET имеется процедура *BeginFrameBasedAnimation*, которая создает покадровую анимацию. Эта процедура принимает в качестве параметра другую процедуру (параметр процедурного типа), которая, собственно, и осуществляет рисование очередного кадра. Процедура рисования кадра может иметь целочисленный параметр, имеющий смысл номера кадра. Через него мы будем передавать количество точек, размещаемых на окружности. Процедура рисования кадра оформляется стандартным способом (т. е. размещается в разделе описаний программы) или может быть передана в *BeginFrameBasedAnimation* как анонимная процедура (лямбда-процедура). Мы воспользуемся смешанным способом: наша процедура рисует картинку для N точек, а вызывать ее мы будем из лямбда-процедуры, в которой будем менять номер кадра. Процедура *Cardioid* при этом останется прежней, а изменения коснутся только главной программы, которая приобретет вид, представленный в листинге 3.

Здесь в *BeginFrameBasedAnimation* передается лямбда-процедура без параметров. В этой процедуре (строки 7–14) рисуется очередной кадр, после чего номер кадра увеличивается на единицу. Проверка в строке 11 позволяет остановить анимацию, когда количество показанных кадров превысит желаемое. Для остановки анимации используется процедура *EndFrameBasedAnimation*. Закрывающаяся круглая скобка в строке 14 относится к первой открывающейся скобке в *BeginFrameBasedAnimation* (строка 7).

Поскольку мы не можем вставить анимацию в статью, приведем ссылку на репозиторий GitHub, где размещены исходный код программы и ее скомпилированная версия [1]. Следует отметить, что для работы скомпилированной программы требуется Microsoft .NET Framework.

Дальнейшие эксперименты с программой можно провести в следующем направлении.

Изменим правило построения хорды $i \rightarrow (2 \cdot i) \bmod N$ на другое. Например, положим: $i \rightarrow (3 \cdot i) \bmod N$. В этом случае мы получим процедуру построения другой кривой — *нефроиды*, которая имеет не один, а два каспа

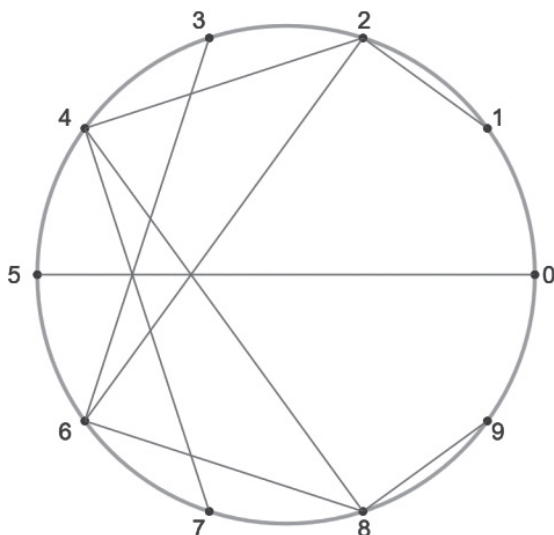


Рис. 1. Семейство хорд для $N = 10$

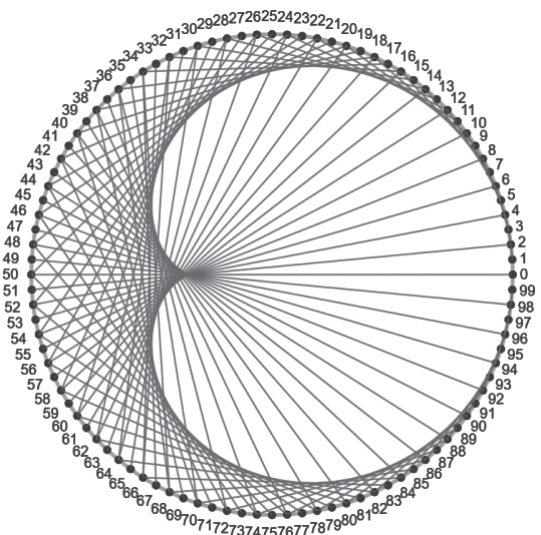


Рис. 2. Семейство хорд для $N = 100$

```

1.  procedure Cardioid(n: integer; col: color := colors.YellowGreen);
2.  begin
3.      var DotColor := colors.Red; // Цвет точек на окружности
4.      var DotRadius := 3; // Радиус точек на окружности
5.      var x := new real[n]; // Массивы для хранения координат
6.      var y := new real[n]; // центров точек
7.      pen.Color := colors.Gold; // Цвет главной окружности
8.      pen.Width := 5; // Толщина пера для рисования линий
9.      // Центр главной окружности - в центре экрана
10.     var (Hu, Hv) := (Window.Width / 2, Window.Height / 2);
11.     var r0 := min(Hu, Hv) * 0.9; // Радиус главной окружности
12.     DrawCircle(Hu, Hv, r0);
13.     // Размещаем n равноотстоящих точек на окружности
14.     // Точки следуют через phi радиан, начиная с позиции start
15.     var (phi, start) := (-2 * pi / n, 0);
16.     pen.Color := DotColor;
17.     Font.Size := 20; // Кегль шрифта для нумерации точек
18.     // Для всех точек:
19.     for var i := 0 to n - 1 do
20.     begin
21.         // Вычисляем координаты i-ой точки
22.         x[i] := r0 * cos(i * phi + start) + Hu;
23.         y[i] := r0 * sin(i * phi + start) + Hv;
24.         // Рисуем точку и подписываем ее номер
25.         circle(x[i], y[i], DotRadius, DotColor);
26.         var u := r0 * 1.075 * cos(i * phi + start) + Hu - 15;
27.         var v := r0 * 1.075 * sin(i * phi + start) + Hv - 10;
28.         DrawText(u, v, 50, 20, '${i}', DotColor, LeftCenter);
29.     end;
30.     // Линии между точками i --> 2 * i (mod N)
31.     Pen.Color := col;
32.     Pen.Width := 2;
33.     for var i := 0 to n - 1 do
34.         Line(x[i], y[i], x[2 * i mod n], y[2 * i mod n]);
35.     end;

```

Листинг 1. Процедура рисования кадра с N точками

```

uses GraphWPF; // Подключаем графический модуль
procedure Cardioid... // Описание процедуры приведено выше
begin
    Window.Maximize; // Графическое окно - на весь экран
    Cardioid(10); // Строим кадр для 10 точек
end.

```

Листинг 2. Главная программа для рисования одного кадра

```

1.  uses GraphWPF; // Подключаем графический модуль
2.  procedure Cardioid... // Описание процедуры приведено выше
3.  begin
4.      Window.Maximize; // Графическое окно - на весь экран
5.      // Count - количество кадров в анимации, i - номер кадра
6.      var (Count, i) := (100, 0);
7.      BeginFrameBasedAnimation(() ->
8.      begin
9.          Cardioid(i);
10.         i += 1;
11.         if i > Count then
12.             EndFrameBasedAnimation;
13.         end
14.     ) // Скобка относится к первой скобке на строке 7
15. end.

```

Листинг 3. Главная программа для создания анимации

(англ. cusp — заострение, точка излома кривой). Вообще, при переходе $i \rightarrow (k \cdot i) \bmod N$, где $k = 1, 2, \dots$, будем получать фигуру с $(k - 1)$ каспом. Хотя с увеличением k огибающую заметить все сложнее из-за большого количества отрезков. В процессе анимации, например, для $k = 19$ будем получать кадры, показанные на рисунке 3.

Решение графических задач повышает мотивацию школьников к изучению программирования и моделирования. Модели, аналогичные приведенной в статье, кажутся ученикам поначалу более близкими и наглядными, чем физические модели, что облегчает им вхождение в предмет моделирования.

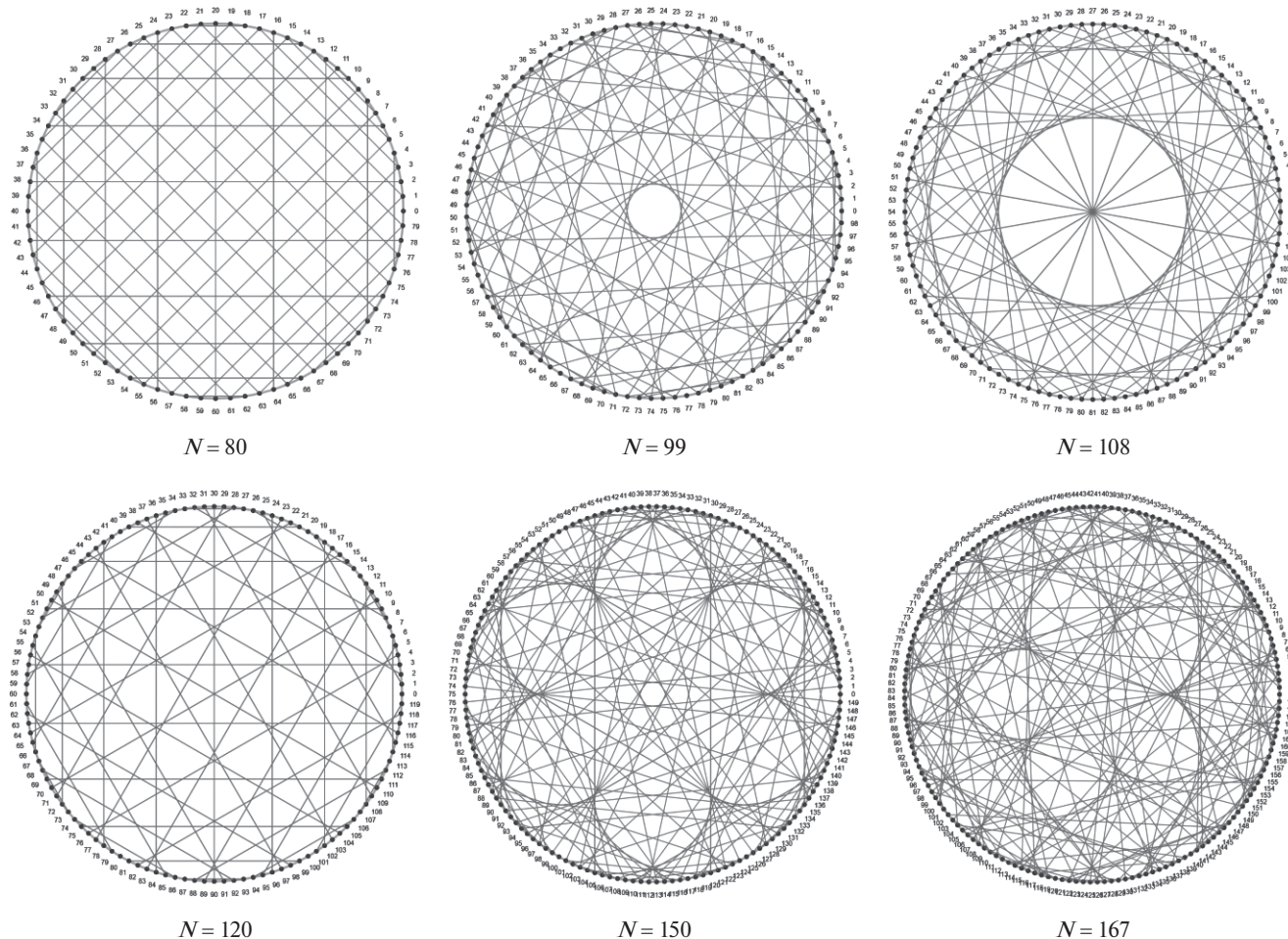


Рис. 3. Кадры при построении семейства хорд по правилу $i \rightarrow (19 \cdot i) \bmod N$

Список использованных источников

1. Анимация кардиоиды как огибающей семейства хорд // GitHub. <https://github.com/vdjenjer/AnimCardioid>
2. Кардиоиды как огибающая семейства отрезков // Википедия. https://en.wikipedia.org/wiki/Cardioid#Cardioid_as_envelope_of_a_pencil_of_lines

3. Математические этюды. <http://www.etudes.ru>
4. Современное программирование на языке Pascal. <http://pascalabc.net/>
5. Трошин П. И. Компьютерная геометрия и геометрическое моделирование (задачи): учебно-методическое пособие. Казань: Казанский федеральный университет, 2015. 55 с.

ПРАВИЛА ДЛЯ АВТОРОВ

Уважаемые коллеги!

Статьи для публикации в журналах «Информатика и образование» и «Информатика в школе» должны отправляться в редакцию **только через электронную форму на сайте ИНФО (раздел «Авторам → Отправка статьи»):**
<http://infojournal.ru/authors/send-article/>

Обращаем ваше внимание, что для отправки статьи необходимо предварительно зарегистрироваться на сайте ИНФО (или авторизоваться — для зарегистрированных пользователей).

С требованиями к оформлению представляемых для публикации материалов можно ознакомиться на сайте ИНФО в разделе **«Авторам»:**

<http://infojournal.ru/authors/>

Дополнительную информацию можно получить в разделе **«Авторам → Часто задаваемые вопросы»:**

<http://infojournal.ru/authors/faq/>

а также в редакции ИНФО:

e-mail: readinfo@infojournal.ru

телефон: (495) 140-19-86