УДК 004.9+501:372.8
Гаев Е. А., Малинина Д.
Национальный авиационный университет, Киев, Украина

ПАРАМЕТРИЧЕСКАЯ РОЗА – ПРЕДМЕТ МАТЕМАТИКИ, ПРОГРАММИРОВАНИЯ, ЭСТЕТИКИ

DOI: 10.14308/ite000616

С помощью MATLAB демонстрируются разнообразные параметрические кривые из семейства "Параметрическая Роза" (Rhodonea), характеризуемого четырьмя коэффициентами. Статья имеет целью заинтересовать учащегося, побудить его к изучению параметрических кривых. Проводится экспериментально-графическое исследование как влияют значения коэффициентов на форму кривой и ее период. Изменение во времени одного из параметров кривой создает эффект анимации. Различные варианты окраски кривой увеличивают эстетическое воздействие результата. На основании описанного предлагается красивая MATLAB-программа, позволяющая "играть" с кривыми на экране компьютера и демонстрирующая удивительные свойства семейства параметрических функций "Роза" в зависимости от значений и соотношения их коэффициентов. Учителям она позволит увлечь учащихся этим дополнительным нешкольным материалом. Ученикам – увидеть красоту математики и получить дополнительные знания о параметрических функциях. Кроме того, программа рассматривается как пример упражнений по курсу алгоритмизации и программирования, вполне доступных современным школьникам. Предложенные варианты анимации кривых могут служить упражнениями как для математики, так и для программирования.

Ключевые слова: *программирование, MATLAB, параметрическая функция, анимация.*

Введение

В сегодняшнем школьном курсе математики ученикам дают только понятие о *параметрическом задании функции* [1] в объеме, достаточном, в основном, лишь для того, чтобы позднее, в ВУЗе, объяснить нахождение производной такой функции и подстановок для вычисления некоторых криволинейных интегралов. Между тем, в дореформенном курсе школьной математики этой теме уделялось значительно большее внимание, по крайней мере в физмат классах и в школьных математических кружках. Исключение названной темы как второстепенной оправдывают желанием разгрузить школьников

Сегодняшний прогресс компьютерной техники, использование математических пакетов, таких как MATLAB, позволяет совершенно иначе, с много большей эффективностью преподнести школьникам "старый" материал, не увеличивая нагрузку на них, а, наоборот, в развлекательно-познавательной форме вовлечь их в процесс получения знаний "игрой" и "экспериментированием" с математикой, расширением их интеллектуальных горизонтов. Известно ведь, что современные дети являются "фанатами" компьютера и, как они это понимают, программирования. Вот и давайте научимся использовать новые компьютерные возможности! При этом выявляется теснейшая связь математики с современным искусством живописи.

Второй аспект данной работы – не пассивное наблюдение того, что дает программа (таких довольно много в Интернете!), а собственное, хотя и простое, программирование. Среда программирования MATLAB дает легкое, быстрое и, главное, увлекательное введение в эту науку. Аналогичный способ "вовлечения через программирование" в те или иные разделы науки предлагают другие авторы в [2-4].

В статье на основе MATLAB-программ изучаются разнообразные, в зависимости от значений коэффициентов, параметрические кривые семейства "*многолепестковая роза*", описывается создание графической программы, которая их анимирует. Работа, мы надеемся, послужит учителям и преподавателям математики и информатики для вовлечения молодых людей в мир строгих наук и программирования. Ученикам же откроется – в прямом смысле – красота математики.

1. Состояние вопроса

В истории математики множество известных сегодня параметрических функций (кривых) связано с самыми блестящими именами математических гениев – Архимедова спираль, циклоида (Николай Кузанский XIY век, Галилей), улитка Паскаля, дельтоида (Л.Эйлер). Сегодня об этих кривых можно прочитать в старых книгах [5,6], в пособиях [7,8] (к сожалению, доступ к последним получить не удалось) и немногих других. Анимированные картинки о том, как строятся эти кривые механически (таким путем, собственно, они и были открыты) и как они выглядят можно увидеть на множестве сайтов [1,9–11]. В частности, много внимания данной теме уделила фирма Wolfram Research [10], создатель математического пакета Mathematica и математической энциклопедии MathWorld, занимающаяся производством математического программного обеспечения. Многие параметрические кривые можно рисовать популярной игрушкой, запатентованной под названием *Спирограф* [12]. В Интернете появляются ее "цифровые аналоги"; например – австралийского Java-программиста Бен Джофа [13].

Изображения, рассмотренные здесь и воспроизводимые нашей программой, можно назвать красивыми. О “*красоте математики*”, собственно, много веков говорят как представители этой науки, так и философы. Очевидно подобие показанных ниже наших изображений тем, что созданы голландским художником М. Эшером [14]. На Западе возникли Музеи математики (или разделы в Музеях науки), представляющие такое “математическое искусство”, [15] и др. Апелляция в данной теме к эстетике увеличивает ее привлекательность для учащихся.

В то же время можно утверждать на основании нашего поиска, что какое-либо специализированное программное обеспечение по поднятым вопросам далеко не полно и потому программа, предлагаемая здесь на основании нашего рассмотрения, аналогов не имеет. Работа продолжает наш поиск в области компьютерных разработок в интересах образования [16-20].

Ограничимся одним семейством параметрических функций, называемым *параметрической розой*, *многолепестковой розой* или также, в зарубежной литературе, Rhodonea. Честь ее открытия принадлежит итальянцу Гвидо Гранди (Guido Grandi, начало 18 века) [21].

2. Постановка задачи

Вообще, известно несколько способов задания функции. *Параметрически заданной* называется функция $y = f(x)$, если и ее аргумент x, и сама функция y связаны друг с другом через некий параметр t с заданной областью изменения $t \in [t_1, t_2]$ посредством двух явных функций

$$x = f_1(t), \qquad y = f_2(t).$$

Сложность такого рода функциональной связи состоит в том, что одному значению аргумента x может отвечать множество значений y. Но это же является причиной красоты графиков параметрических функций, т.к. они могут выглядеть намного сложнее явных функций, как это и показано на последующих рисунках 1, 4 и 5.

Уравнение многолепестковой розы [10,11] можно записать в следующем общем виде:

$$x = (r\sin(nt) + a)\cos(t + \varphi), \qquad y = (r\sin(nt) + a)\sin(t + \varphi), \qquad t \in [0, k\pi], \tag{1}$$

где коэффициенты n, a, k и φ могут иметь различные значения, а r, как правило, равно 1. В среде MATLAB построение ее графиков может быть осуществлено многими способами, в частности, таким:

```
>>k=2; t=0:pi/1000:k*pi; %Таблица значений параметра t
>> %Задание коэффициентов кривой и получение таблицы значений x и y:
>> a=.1; n=8; Fi=0; x=(a+sin(n*t)).*cos(t+Fi);  y=(a+sin(n*t)).*sin(t+Fi);           (2)
>> % Построение графика "по точкам" в новом окне:
>> figure, Color='r'; plot(x, y, 'Color', Color), axis equal
```

(пока что ограничиваем обсуждение значением $\varphi = 0$, остальными параметрами рекомендуем "поиграть"; знаком ">>", называемого *prompt* (приглашение), условимся показывать то, что действие выполняется в командной строке; команды без этого "приглашения" далее будем писать в листингах MATLAB-программ. Переменная *pi* понимается MATLAB как число π. Знаком процента % выделяют комментарии). Получаем весьма разнообразные кривые в зависимости от конкретного значения коэффициентов n и a: "многолепестковые розы" при a=0 и n целом (при этом количество лепестков совпадает с n при нечетном его значении, и равно $2n$ при четном); при a в диапазоне $0 < a \le 1$ лепестки "раздваиваются", появляются бо'льшие и меньшие; при $a > 1$ малые лепестки "отрываются" от начала координат и фигура напоминает шестерню, приближающуюся по форме к окружности с радиусом $a + 1$.

Еще более разнообразными, и даже запутанными, кривые становятся при рациональном $n = {}^{p}\!/_{q}$. Опция *Color* последней команды задает тот или иной цвет кривой; например при *Color*='*r*' или *Color*=[1,0,0] кривые имеют красный цвет, при *Color*='*b*' или *Color*=[0,1,0] – синий, при *Color*='*g*' или *Color*=[0,0,1] – зеленый. А коэффициент k задает длину входного числового вектора t; этому вопросу посвящен, преимущественно, следующий раздел. Некоторые из кривых можно видеть на рис. 1, больше – на сайтах [10,11,22,23]. При $n = 1$ график вырождается в кривую, зависящую лишь от параметров a и k. Такое семейство кривых можно построить "экспериментально" командами (2). Но и полезно предложить учащемуся "аналитически" получить $x^2 + y^2$. Вообще, соотношение демонстрации явления ("эксперимента") и его доказательство – дополнительный аспект, который может быть продемонстрирован учащемуся в этой работе.

Целью работы является демонстрация разновидностей этих кривых, их анимация на экране компьютера с использованием при этом различных цветовых эффектов, а также создание соответствующих компьютерных программ. Они позволяют с удовольствием "играть" с кривыми, с пользой размышлять об их поведении и, таким образом, будут полезными в учебном процессе.

3. Математический аспект проблемы

В море современной информации [3-11] можно найти лишь лаконичное описание свойств параметрической розы. Поэтому уделим этому некоторое внимание, безусловно полезное учащимся.

Чтобы прогнозировать вид кривой в зависимости от значений коэффициентов, полезно рассмотреть ее "механическое происхождение". Представим себе штырь длиной R, равномерно вращающийся вокруг одного из его концов, в котором разместим неподвижную систему декартовых координат Oxy. На противоположном конце штыря находится точка, которая колеблется вдоль него от положения $L_1 = a + r \overset{Df}{=} R$ до $L_2 = a - r$ по гармоническому закону $L = a + r\sin nt$, где L – удаление точки от оси вращения вдоль штыря, и r – размах колебаний, рис. 2. Центр этих колебаний находится, таким образом, в точке штыря $L_0 = a$. На каком бы удалении L не находилась бы колеблющаяся точка, ее координаты в Oxy выражаются формулами $x = L\cos t$, $y = L\sin t$, где t – угол стержня к оси Ox.

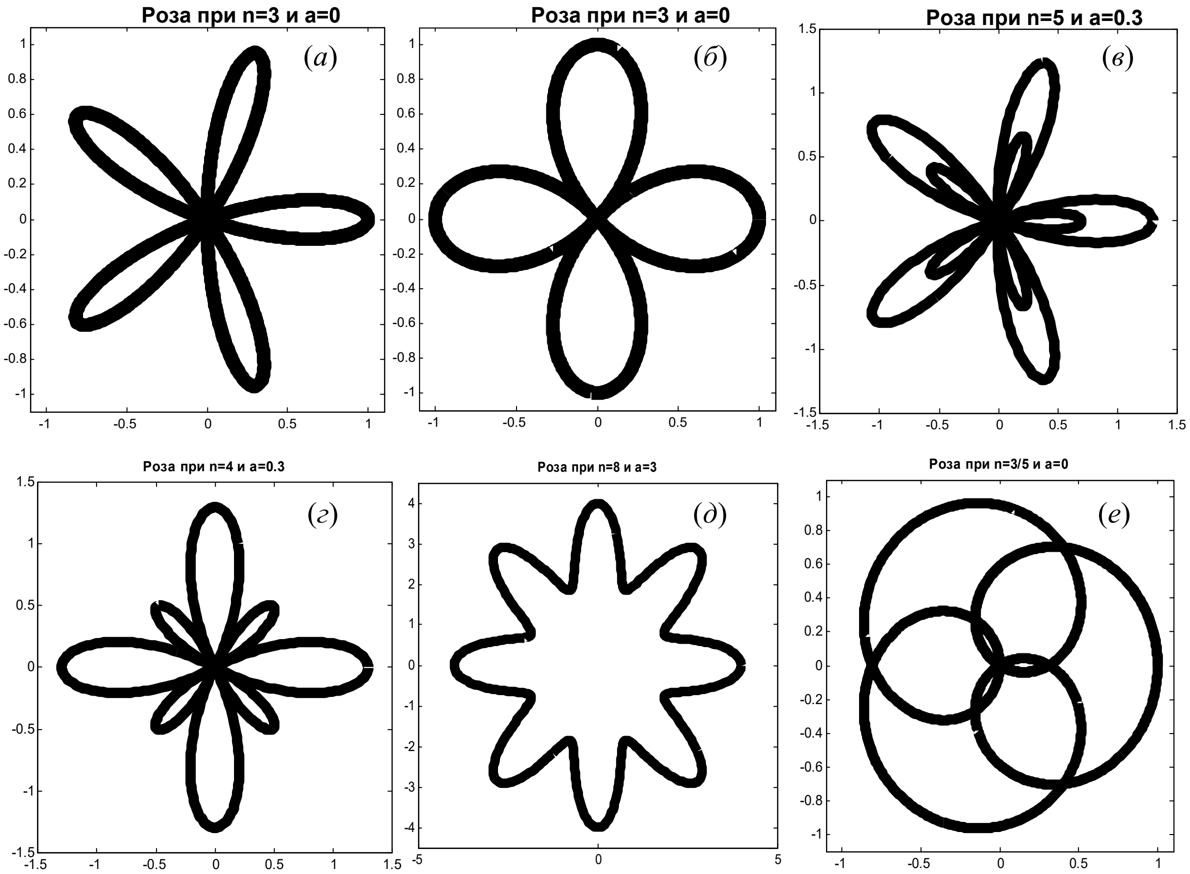

Рис. 1. Разнообразие "параметрических роз" (1) в зависимости от значений коэффициентов: (а) a=0, n нечетное; (б) n четное; (в) a=0.3, n нечетное; (г) n четное;

Следовательно, фигура, описываемая колеблющейся точкой в неподвижных координатах, и есть параметрическая кривая, подчиняющаяся уравнению (1). При этом n получает смысл частоты колебаний точки по отношению к полному обороту стержня. Фигура приобретает несколько "лепестков", если точка совершает несколько колебаний за оборот, $n>1$. Если при этом удаление центра колебаний a до центра вращения превосходит размах колебаний, $a>r$, то "лепестки" не будут доходить до оси вращения, как у фигуры типа (*д*) на рис. 1. При $a=r$ точка периодически прикасается к началу координат – получаем лепестки одинаковой длины, случаи (*а*) и (*б*); если же $a<r$, точка будет переходить через центр вращения и за ним получим лепестки меньшей длины, случаи (*в*) и (*г*) рис. 1.

Механическая интерпретация позволяет понять и роль коэффициента φ в уравнениях (1): это – угол начального положения стержня, $\varphi|_{t=0}$. Ему будет посвящен отдельный раздел 4.

Дополнительному прояснению проблемы способствует переход от декартовых координат (1) к полярным. Возводя x и y в квадрат, получаем для удаления ρ от начала координат

$$\rho^2 = x^2 + y^2 = a^2 + 2ar\sin nt\left(\sin(t+\varphi)+\cos(t+\varphi)\right)+r^2\sin^2 nt\,. \qquad (3)$$

В частности, при $a=0$ имеем $\rho = r\sin nt$. Последняя формула показывает, что за полный оборот угла $t\in[0,2\pi]$ происходит $2n$ изменений ρ от $\rho=r$ до $\rho=-r$. Многие изложения именно из такого полярного уравнения и исходят, [5-10]. При иных значениях коэффициента $a>0$ ситуация, согласно (3), сложнее.

При втором, третьем и т.д. обороте продолжение фигуры может налагаться на нее же (и тогда возникает периодичность кривой), а может и не налагаться. Это зависит от частоты колебаний n. Для понимания его роли механической интерпретации недостаточно и требуется более глубокий анализ. Воспользуемся для этого рекомендациями [5] и рассмотрим каждое из уравнений (1). Заметим, что коэффициент a (*offset*, *смещение*) можно измерять в долях амплитуды колебаний r, что эквивалентно предположению $r=1$. Задача становится проще, т.к. одним коэффициентом в ней меньше.

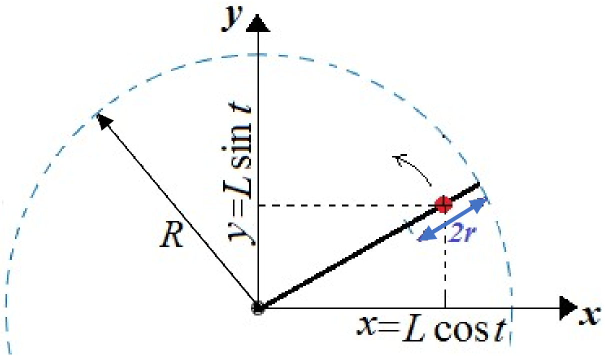

Рис. 2. Схема механического образования параметрической розы и вывода её уравнения (1).

Периодом функций $\sin nt$ и $\cos nt$ будет $T_1=\frac{2\pi}{n}$; периодом множителей $\cos t$ и $\sin t$ будет $T_2=2\pi$. Если величины T_1 и T_2 соизмеримы (n целое или рациональное, $n=\frac{p}{q}$) – тогда T, период произведения этих функций, существует; если они несоизмеримы, например $n=\pi$ или $n=\sqrt{2}$ – периодичность не появится. Последний случай не рассматриваем, а по поводу первых можно высказаться точнее, в виде нескольких утверждений:

1. Если n целое – периодом будет наименьшее возможное T, в которое "укладывается" (делится нацело) и T_1, и T_2. Ясно, что это будет $T=2\pi$. Докажем это строго, сравнив $x_2=x(t+T)$ и $x_1=x(t)$, $y_2=y(t+T)$ и $y_1=y(t)$:

$$x_2=x(t+T)=\left(a+\sin n(t+2\pi)\right)\cos(t+2\pi)=\left(a+\sin(nt+2n\pi)\right)\cos(t)=$$
$$=\left(a+\sin nt\right)\cos(t)=x_1$$

Аналогично устанавливаем, что $y_2=y_1$. Следовательно, при построении кривой (1) для целого n следует задавать параметр на интервале $t\in[0,2\pi]$. При его дальнейшем увеличении кривая будет совпадать с уже начертанной; если же взять более узкий интервал – кривая будет незамкнутой, неоконченной. Далее убедимся в этом графически.

2. При $a=0$ утверждение можно уточнить: если n целое и нечетное – периодом будет $T=\pi$, и потому при построении кривой (1) можно брать $k=1$, т.е. рассматривать ее на интервале $t\in[0,\pi]$. При $a\neq 0$ это становится неверным, поэтому и для любых целых n будем брать $t\in[0,2\pi]$, несмотря на то, что новые точки ложатся на предыдущие.

При всей важности строгих доказательств, более убедительно для учащихся может смотреться иллюстрация только что установленных фактов путем графических построений с помощью MATLAB-программы. Дадим ее пример, в котором большинство операторов понятны из их названия, другие поясняются комментариями:

Листинг 1 файла MATLAB-файла (программы) *Learn1RoseCoeff.m*

```
function Learn1RoseCoeff(a, n, Color, k)
 %Программа #1 для исследования периодичности ветвей параметрической Розы (1)
 % x=F1(t), сплошной линией цвета Color, и y=F2(t), штриховой,
 % в зависимости от офсета a, частоты n и k, количества длин [0,pi].
Fi=0; t=0:pi/1000:k*pi;  % вектор значений параметра
```

```
        x=(a+sin(n*t)).*cos(t+Fi);  y=(a+sin(n*t)).*sin(t+Fi);
        Col1=[Color,'-'];  Col2=[Color,'--'];  % Подготовка двух стилей линий
        plot(t,x,Col1), hold on   % Построение x(t) на интервале  [0, k*pi]
        plot(t,y,Col2) % Построение y(t) на  [0, k*pi]
        plot(t, 0*t,'k'), plot([pi,pi],[-.5,.5], 'k'), plot([2*pi,2*pi],[-
.5,.5],'k')
        xlabel('Переменная t'), ylabel('Функции x и y')  % Надписи на осях
        legend('X--график','Y-график') % пояснение к кривым
      % Заголовок рисунка со значениями коэффициентов:
      title(['Роза при a=',num2str(a),'  n=',num2str(n),' на интервале
[0,',num2str(k),'\pi]'])
```

В командном окне MATLAB проверим с ее помощью поведение ветвей розы при $n=3$ и $n=2$, нечетном и четном n:

```
>> Learn1RoseCoeff(0,2,'r',2)
>> Learn1RoseCoeff(0,3,'g',2)
```

Действительно, убеждаемся (рис. 3), что сдвиг аргумента на π приводит к тем же кривым при n нечетном, и к изменению их знака на противоположный при n четном ($T=\pi$ т $T=2\pi$ соответственно). Аналогично можно "просмотреть" поведение ветвей при $a\neq 0$ на интервале, например, $0\le t\le 4\pi$:

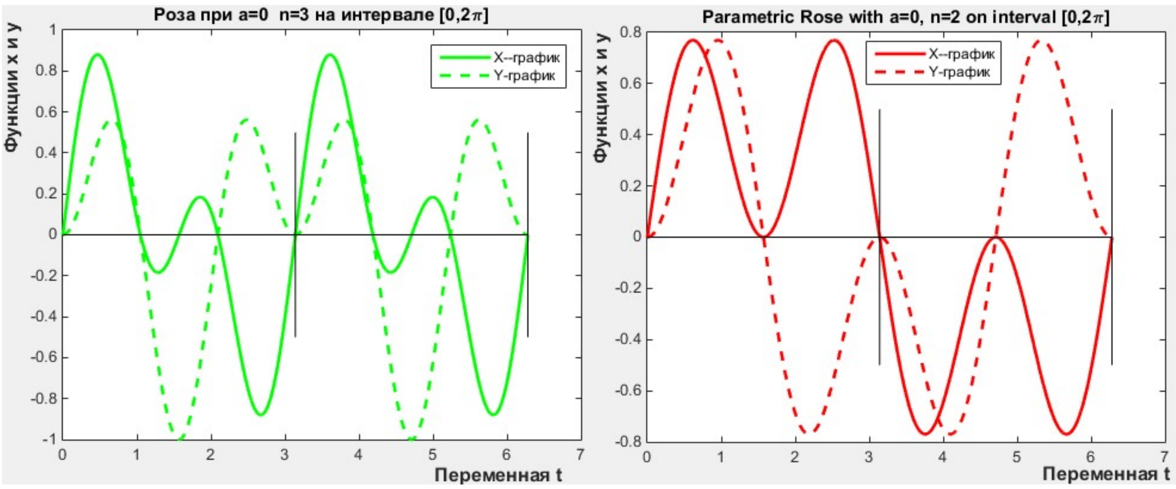

Рис. 3. Поведение ветвей розы (1) при нечетном (слева) и четном n (справа), полученное программой Learn1RoseCoeff

```
>> Learn1RoseCoeff(0.5,2,'b',4)
```

3. Рассмотрим дробное $n={}^{1}/_{q}$, где q целое. Тогда $T_1=2\pi q$, $T_2=2\pi$ укладывается в длину T_1 целое число раз и, следовательно, периодом кривой будет T_1.

4. Пусть теперь $n={}^{p}/_{q}$, где и p целое. Тогда $T_1={}^{2\pi q}/_{p}$. Как найти "общий период" с T_2? Два числовых примера: 4.1. Пусть $n={}^{3}/_{5}$. Тогда $T_1={}^{10\pi}/_{3}$ и общий период с T_2 есть $T=10\pi$. 4.2. Пусть $n={}^{3}/_{8}$. Тогда $T_1={}^{16\pi}/_{3}=\left(5+{}^{1}/_{3}\right)\pi$. Длины T_1 и T_2 укладываются в общую длину $T=16\pi$, которая и будет периодом, так что для построения полной кривой (1) в этом случае следует задать параметр от $t=0$ до $t=T=16\pi$, т.е. в (1) полагать $k=16$.

Из примеров вытекает общее утверждение: при a рациональном и равном $a = p/q$ число $T = 2\pi q$ будет периодом. Только знаменатель важен! Действительно, длина T_1 укладывается в нее p раз ($T/T_1 = \frac{2\pi q}{\frac{2\pi q}{p}} = p$), а длина T_2 – q раз; оба отношения являются целыми числами. Можно предложить следующий MATLAB-алгоритм определения периода T в зависимости от n:

```
>>  [Nom,DeNom]=rat(n);   T=2*pi*DeNom,
```
(4)

где MATLAB-функция *rat*( ) “отделяет” числитель и знаменатель от рационального аргумента *n*. (Замечание: если p и q не взаимно просты, можно выбрать меньший период T, что экономит процессорное время. Мы, однако, не будем это учитывать).

Сформулированные математические соображения не всегда убедительны для учащихся. MATLAB позволяет их "оживить" следующими действиями в командном окне:

```
 >> Fi=0; a=0.3; n=1/3; k=12;
>>t=0:pi/10000:k*pi; %Таблица значений параметра t на k периодов      (5)
 >> x=(a+sin(n*t)).*cos(t+Fi);  y=(a+sin(n*t)).*sin(t+Fi);
>> comet(x,y)  %Простейшая из команд анимации
```

Можно видеть постепенное построение кривой $y = f(x)$ "по точкам" и ее повторение при заходе на следующий период. (В строчке (5) использован шаг *pi*/10000; он него зависит скорость "кометы"; на медленных компьютерах его надо увеличить, на быстрых – уменьшить). Команда анимации *comet*, однако, не позволяет управлять выдачей информации про текущее значение t командой *title*. Это побудило нас предложить альтернативную программу:

Листинг 2 файла *Learn2RoseCoeff.m*

```
function Learn2RoseCoeff(a,n)
%Анимация построения розы (1) по точкам.
% Смысл аргументов: a офсет, n частота.

Fi=0; k=2; % Коэффициенты, в т.ч. k, количество периодов для демонстрации
[Nom, DeNom]=rat(n);   T=2*pi*DeNom; % Определение периода по (4)
 t=0: pi/200: k*T;%Таблица значений параметра t для k периодов
x=(a+sin(n*t)).*cos(t+Fi);  y=(a+sin(n*t)).*sin(t+Fi);
Xmin=min(x); Xmax=max(x);  %Габариты графического окна
Ymin=min(y); Ymax=max(y);
j=0;  L=length(t);
for i=1:L %Цикл для перебора всех t
      j=j+1;    hold off   %Очистка предыдущего графика
      plot(x(j), y(j), 'or')  %"Движущаяся" точка красного цвета
      axis(1.1*[Xmin   Xmax,  Ymin   Ymax]) %Установка габаритов
графического окна
      hold on %Команда на сохранение графиков в окне
      if j>1
           plot(x(1), y(1), '*g')  %Построение начального положения движ.
точки
      end
      X=x(1:j); Y=y(1:j); %Вся траектория от t1 до tj,
      plot(X,Y,'b-'),       axis equal   %и ее построение
      tPi=round(10*t(j)/pi)/10;  %Вся
      title(['Период T=', num2str(T/pi),'\pi', '. Текущее положение t=',
num2str(tPi), '\pi'])
```

```
        pause(.01) %Без этого иллюзии движения не будет!
end
```

Примеры ее запуска:

```
>>  Learn2RoseCoeff(0.5,5)  %в ней a=0.5 и n=5
>>  Learn2RoseCoeff(0,1/5)  %в ней a=0 и n=1/5
```

Мы реально видим на экране постепенное построение кривой движущейся точкой, отвечающей текущему значению t. Рисунки 4, полученные для этих двух разных расчетных случаев, лишь частично могут это отразить. Кривая постепенно строится сначала на одном периоде, начиная из положения, отмеченного зеленым. Одновременно программа выдает в заголовок текущее значение t в долях π. Убеждаемся визуально, что для любого a и n после $t=T$, на втором и последующих периодах, движущаяся точка повторяет предыдущую кривую. Это и иллюстрирует доказанные выше утверждения. На рис. 4*а* точка еще не закончила построение кривой и продолжает движение, как указано стрелкой. На рис. 4*б* движущаяся точка повторно описывает прежнюю траекторию; по окончанию второго периода она придет в начальное положение.

4. Визуальный эффект анимации

Если возможные кривые, рис. 1 и другие, строить на экране, последовательно меняя с некоторым интервалом времени один или несколько параметров кривых, то можно получить и другие, более впечатляющие визуальные эффекты анимации, состоящие в преобразовании одного изображения в другое. При этом синхронно выводим на экран и уравнения текущей кривой с актуальным значением ее параметров. Это и должно убедить пользователя, что за красивым эффектом скрывается все-таки математика.

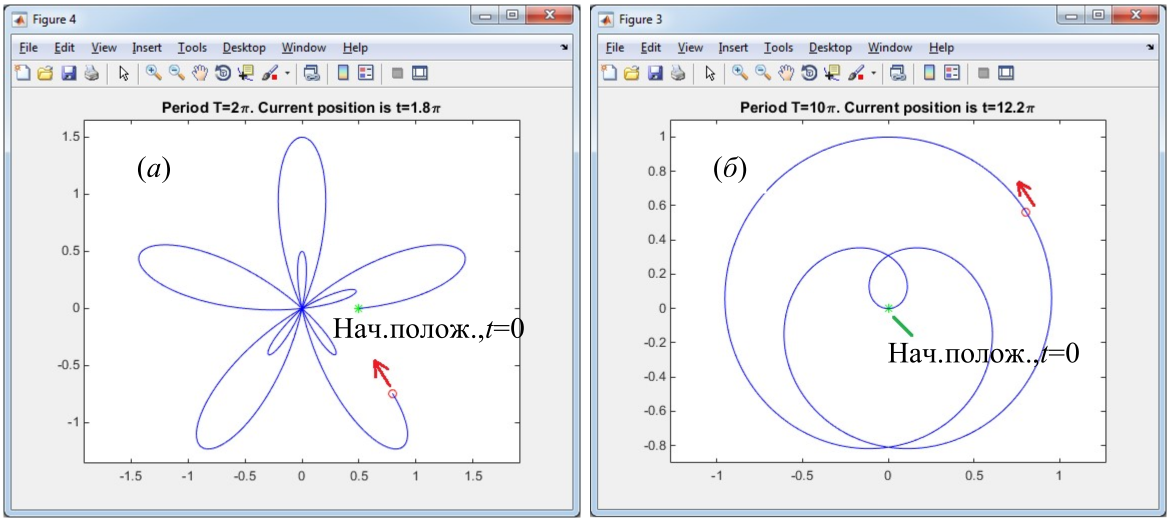

Рис. 4. Результаты анимации построения графиков (1) программой Learn2RoseCoeff в случаях

(а) a=0.5, n=5 и (б) a=0, n=1/5.

4.1. Вращение

Коэффициент φ в уравнениях параметрической розы (1), как это видно из ее механической интерпретации, имеет смысл начального угла, угла наклона построенной кривой относительно оси Ox. Дадим "зрителю" последовательные изображения этой кривой при углах φ=0:`dFi`:`2*k*pi` (т.е. начиная от 0 до $2k\pi$ с шагом `dFi`); при этом шаг `dFi` разумно брать в долях π. Получим эффект вращения фигуры на экране. Эту идею реализует следующая программа `MultyVariableRoseRotation.m`:

```
Листинг 3 _программы "MultyVariableRoseRotation.m".
```

```
function MultyVariableRoseRotation(n,a,Dir,Color)
%Вращение Розы (1). Смысл параметров:
%n целое или рациональное, определяет вид фигуры;
%Dir - направление вращения;
%a – оффсет, и Color – цвет фигуры.
%Примеры запуска из командного рядка:
%   >>MultyVariableRoseRotation(8,.3,1,'m'),
%   >>MultyVariableRoseRotation(5/3,.3,-1,'g')

[Nom,DeNom]=rat(n); T=2*pi*DeNom;%Вычисление периода T
dt=pi/500;  t=0:dt:T; %Массив значений параметра
for Fi=Dir*(0:pi/50:2*5*pi)  %5 оборотов
      x=(a+sin(n*t)).*cos(t+Fi);
      y=(a+sin(n*t)).*sin(t+Fi);
      plot(x,y, Color,'LineWidth',3); %построение кривой
%fill(x,y,Color);                                              (6)
      hold on, plot([0],[0],'or');          %построение центра вращения
      title(['Роза x=(a+sin(n*t))*cos(t), y=(a+sin(n*t))*sin(t)'...
             ' при a=',num2str(a),' и
n=',num2str(Nom),'/',num2str(DeNom)])
      axis equal;   axis(1.1*[-a-1,a+1, -a-1,a+1]), axis off
      pause(.1) %Без искусственной паузы не будет анимации!
      hold off    %Освобождение рисунка для новой кривой в цикле
end
```

Получаем вращение на экране той или иной фигуры. Последнюю имеем в виде контура, либо заполненной цветом, если команда (6) раскомментирована, а предыдущая – закомментирована. Ее цвет задаем в командном задании как '*m*', как '*g*' или иным образом. Направление вращения легко изменить на противоположное (третий аргумент). Учащиеся охотно экспериментируют с такими кривыми, меняя также два первых аргумента задания. Задача учителя только в том, чтобы наблюдение кривых сопроводить математическим содержанием.

4.2. Анимация посредством офсета *a*

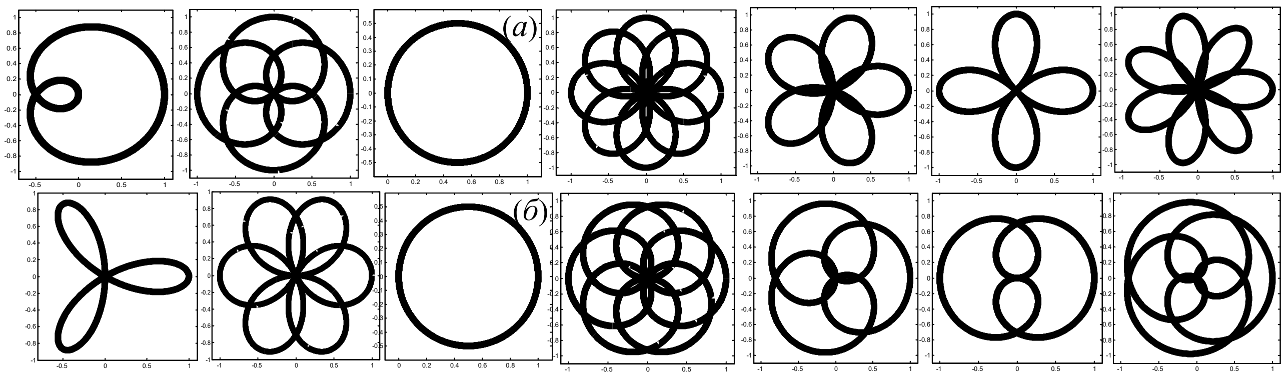

Рис. 5. Последовательная трансформация фигур при анимации (а) увеличением числителя p, $q = 3$; (б) увеличением знаменателя q, $p = 3$. Офсет отсутствует, $a = 0$.

В последней программе значения аргументов *n* и *a* задаются при старте, и далее не изменяются. Если же в ней вместо *Fi*=0:*pi*/50:2*5**pi* менять в цикле офсет *a* – получим иную анимацию на экране в виде изменения размера лепестков розы (1), рис. 1,*а-б*, появления укороченных лепестков, рис. *в-г*, перехода от "розы" к "шестеренке", рис. 1*д*. На сайте [23] дан "стационарный движок" таких преобразований. На мультимедийном рисунке сайта [11] показаны некоторые возможные эффекты анимации. Более того, за полтора года работы над данной публикацией, там добавлена и ее MATLAB-программа (хотя и ошибочная). Последний факт свидетельствует, что поиски математиков-популяризаторов идут в одном с нами направлении.

4.3. Анимация для рациональных n

Коэффициент n, в общем случае, рационален, $n = p/q$. Если целочисленное значение числителя или знаменателя зафиксировать, а другое увеличивать в цикле с каким-то шагом (будем брать его равным 1), то анимация воспроизведет сложное преобразование одной фигуры в другую, иной раз – сильно отличающуюся. В литературе мы не нашли случаев использования такого эффекта.

Возможны, таким образом, два варианта:

(*а*) изменяется числитель p (пример: $n = \frac{1}{3}, \frac{2}{3}, \frac{3}{3} = 1, \frac{4}{3}, \frac{5}{3}, \frac{6}{3} = 2, \frac{7}{3},$ и т.д.);

(*б*) изменяется знаменатель q (пример: $n = \frac{3}{1} = 3, \frac{3}{2}, \frac{3}{3} = 1, \frac{3}{4}, \frac{3}{5}, \frac{3}{6} = \frac{1}{2}, \frac{3}{7},$ и т.д.).

Во втором случае особенно важно, чтобы программа "приспосабливалась", выбирая согласно (4) каждый раз новый период T, необходимый для построения полной кривой (раздел 2). Примеры последовательной трансформации на экране одной фигуры в другую для указанных случаев показаны на рис. 5. Как видим, иногда возникает "тривиальный случай" – окружность. Можно, при желании, его опустить в программе с помощью логического оператора.

Программа построена таким образом, что вначале предлагается выбрать, как запускать анимацию – увеличивать на 1 числитель p или знаменатель q, и в каком именно диапазоне. Приняв решение, для другого коэффициента надо указать его (постоянное) значение.

Одновременно в ходе анимации можно также изменять окраску кривых на каждом шаге. Этот прием широко используется в данной программе для усиления эстетического эффекта. Дополнительные сведения об этом – в следующем разделе.

5. Цветовые эффекты

Собственно, мы уже показали способ придать линиям тот или иной цвет, ту или иную толщину. Однако, нам представляется важным кратко пояснить учащимся сколь глубокая наука и сложные технологии “скрываются” за простыми командами, которые мы используем в языках высокого уровня.

Безграничные цветовые возможности современных программ обеспечиваются, в первую очередь, созданными сегодня техническими средствами – дисплеями на электронно-лучевых трубках (кинескопах), на жидких кристаллах и др. [21,22]. Во-вторых (и для программистов это важно в первую очередь), между творческим замыслом и его реализацией на экране находится “посредник” в виде низкоуровневых команд, управляющих дисплеем. Сегодня за все вопросы “взаимоотношений” с дисплеем отвечает операционная система. Поэтому, программист может не задумываться, какой у него экран – команды в среде программирования будут одинаковы. Опишем еще два способа, предоставляемые MATLAB.

5.1. Заполнение цветом

Альтернативой команде *plot* является команда *fill*, отличающаяся тем, что, если линия является замкнутой, указанным цветом заполняется вся ее внутренняя область. В этом можно убедиться, если в листинге 4 раскомментировать команду (6), а предыдущую – напротив – закомментировать знаком “%”. Возникает интересный эффект "мигания", если замкнутую кривую описывать не на одном периоде T, а на двух-трех.

Еще одним свойством команды *fill* является возможность задавать разный цвет для последовательных вершин полигона. Тогда "числовое значение цвета" в любой промежуточной точке области определяется интерполяцией значений соседних вершин. В таком случае опция `Color` должна быть текстовым или числовым вектором такой же размерности, как и векторы вершин `x` и `y`, а именно 1*length(`x`). Самым простым способом, но и наиболее эффектным, будет генерирование случайного числового вектора командой `Color`=*rand*(1,*length*(`x`)). При таком окрашивании достигается особо впечатляющий цветовой эффект.

Например, вычислим вершины x и y правильного 5-угольника:

```
>> N=5; Fi0=pi/6; %начальный угол поворота
>> Fi=Fi0 + 0 : 2*pi/N : 2*pi*(N+1)/N;   x=sin(Fi); y=cos(Fi);
```

и многократно повторяем операцию

```
>> figure(1), fill(x,y, rand(1,6)), axis equal
```
(7)

Будем получать фигуры, раскрашенные самым причудливым образом, как это иллюстрирует рис. 6.

5.2. Использование палитры цвета

Современные программы используют то неисчислимое множество цветов, которое предоставляет операционная система. В то же время, MATLAB имеет несколько вариантов ограниченных наборов (палитр), называемых *colormap*. Команду *fill* можно использовать и в пределах одной палитры. А можно – и изменять палитру от цикла к циклу в ходе анимации. Подготовленные палитры содержат многие “старые” программы и игры, что позволяло им экономить память и повышать быстродействие.

На рис. 6 использована палитра, называемая *hsv*. Установить иную палитру можно такой, например, командой:

\>> *colormap*(*cool*)

В результате фигуры будут окрашены “холодными” цветами. Предлагаемая ниже программа позволяет выбор определенной палетки из заданного набора в зависимости от целого числа в аргументе.

Листинг 4 программы "*Colors.m*".

```
function Cmap=Colors(n)
% Данный алгоритм позволяет выбрать палетку из заданных
% в зависимости от числа.
% cmap0 – заданные палитры цветов
% L – длина вектора с палитрами
% n – порядковый номер желаемой палитры
cmap0={autumn,hot,summer,hsv,spring,cool,winter,hsv,parula};
L=length(cmap0);
if n<=L
    cmap=cmap0{n};
elseif n>L  % повтор тех же палитр, если n>L
    m=n-L;
    while m>L
        m=m-L;
    end
    cmap=cmap0{m};
end
colormap(cmap)  % конец программы
```

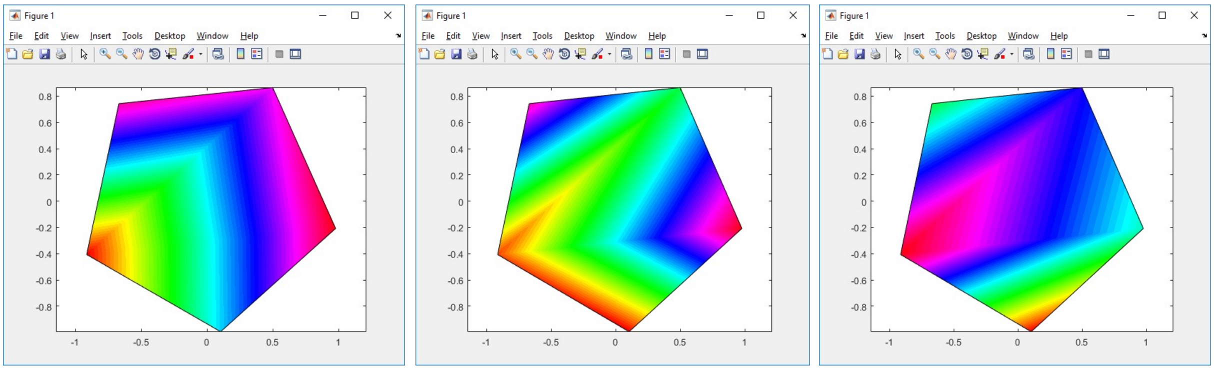

Рис. 6. Случайные цвета N-угольника, полученные способом (7).

Она позволяет менять палитры окрашивания в ходе циклов анимации, и тем самым создает дополнительные эффекты.

6. Включение программ в GUI-оболочку

В помещенных выше программах (*m*-файлах, как принято говорить в MATLAB) мы использовали не только команды MATLAB (программы из его ядра), но и стандартные операторы программирования, общие практически для всех языков – ***for***, ***if*** … ***else*** … ***elseif*** (иногда еще ***switch***…***case***). Последнее делает эту среду удобной для обучения программированию. К этому следует добавить и возможность построения современного графического интерфейса.

Выше уже описаны все фрагменты кода, на которых можно построить составную программу для запуска из командного окна MATLAB. Сегодня модно, однако, "заворачивать" программы в GUI-оболочки. Графическая среда для разработки GUI-оболочки (***G***raphical ***U***ser ***I***nterface) в MATLAB создается запуском команды *guide* в командном окне

```
>> guide
```

Возникает еще одно MATLAB-окно, предназначенное для визуального дизайна будущей графической программы из GUI-элементов "*надпись*", "*окно ввода*", "*нажимаемая кнопка*", "*выпадающее меню*", "*чекбокс*" и др. (соответственно, *StaticText*, *EditText*, *PushButton*, *Pop-Up menu*, *CheckBox*). Построенное графическое окно с GUI–элементами сохраняется под каким-то именем в файле специального формата с расширением ***.****fig*. Одновременно возникает и далее программируется одноименный текстовый *m*-файл с кодами-заготовками, ассоциированными с используемыми GUI-элементами. Опыт создания GUI-программ описан в [19,20,24].

Стандартные *guide*-средства MATLAB во многом удовлетворяют эстетическим запросам учащихся-разработчиков, но все же ограничены как набором цветов, так и формой GUI-кнопок и GUI-окон только в виде прямоугольников. В то же время, недавно найдены и другие возможности сделать графическую программу намного более привлекательной, рис. 7. Имеется также возможность запустить сопровождающую музыку. Это будет освещено в отдельной публикации.

Внешний вид итоговой графической программы показан на рис. 7. Созданная программа хранится в памяти компьютера в двух файлах *Curves.fig* и *Curves.m*. Ее запуск производится из командного окна MATLAB посредством команды

```
>> Curves
```

На экране сначала появляется графическое окно, показанное слева вверху на рис. 7,*а*. Кроме поясняющих надписей, там имеется выпадающее меню, где нужно выбрать один из двух вариантов запуска, описанных в 4.3. Когда выбор сделан, появляется следующее графическое окно (рис. 7,*б* справа вверху). В окна надо ввести желаемые численные значения диапазона изменения параметра *d* или *r*, и нажать кнопку "*Старт*". В результате видим фигуру, изменяющую во времени свою форму и окраску (рис. 7,*в* слева внизу). Если

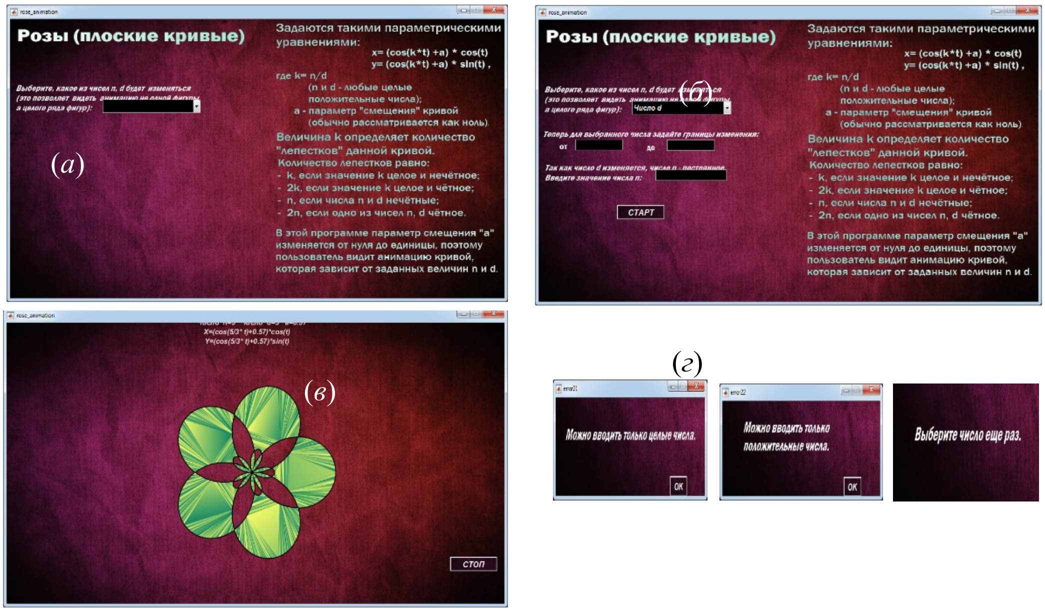

Рис. 7. Последовательность графических окон (а)–(в) после запуска программы Curves.m;

(г) – вспомогательные окна в случае ошибок пользователя.

пользователь ввел некорректную информацию – его предупреждает об этом одно из сообщений (рис. 7,*г* справа внизу). Изменение фигуры сопровождается демонстрацией ее параметрического уравнения с текущими значениями коэффициентов, которые изменяются с некоторым интервалом времени.

Как видим, программа решает поставленную в начале статьи задачу: демонстрацию параметрических кривых семейства (1), сопровождая ее математическими пояснениями.

Выводы

Таким образом, классическую тему о параметрических кривых предлагается рассматривать по-новому, сопровождая ее построением большого количества разнообразных графиков в MATLAB, или даже несложного программирования в этой среде. Сформулированы математические предложения о свойствах кривых семейства *Роза* (1) и программы, позволяющие дать впечатляющие иллюстрации к ним.

MATLAB оказался удобной средой для разработки данного учебного приложения. Он позволяет значительно повысить интерес учащихся к математике, соединив ее с простым программированием, как это и ранее предлагалось в [18,19]. Возможно, урок следует строить как свободный эксперимент (игру) учащегося с параметрической кривой, выполняемый через разработанную программу. После игры – предложить ему задуматься, “почему это так?”, обратить внимание на уравнение семейства кривых (1).

Мы же в нашей практике решаем “обратную задачу”: используем эффектные математические задачи для мотивации студентов при изучении программирования. Однако, задача на программирование мотивирует и на изучение математических аспектов проблемы. Полагаем, что MATLAB подтвердил свою репутацию как среды, в которой наиболее эффективно изучать алгоритмизацию и программирование.

Благодарности

Авторы выражают благодарность студентам группы ІАН-303а Национального авиационного университета Настасье Скородед и Катерине Хаврай за помощь в художественном оформлении главного окна GUI и GUI-кнопок. Студентка Екатерина Мелах помогла в подборе числовых комбинаций, дающих впечатляющую анимацию. Студентов Андрея Баборыгу и Алексея Бойко благодарим за интенсивное тестирование окончательной программы, выявление ее ошибок и недостатков.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. https://ru.wikipedia.org/wiki/Параметрическое представление – общие сведения о параметрическом представлении функций.
2. Обруцький А.М. Фізика на Паскалі: Практикум. – Дніпропетровськ, 2006.– 224 с.
3. Wagon S. Mathematica in Action: Problem Solving Through Visualization and Computation. Springer Pbl., 2010, 574 p.
4. Cundy M.H., Rollett A.P. Mathematical Models. Oxford University Press, 1974. – 286 p.
5. Вирченко Н.А. Графики функций. Справочник / Вирченко Н.А., Ляшко И.И., Швецов К.И. – К.: Наук. думка, 1979. – 320 с.
6. Клетеник Д.В. Сборник задач по аналитической геометрии. М.: Наука, 1980. – 240 с.
7. Соболев С.К. Исследование и построение плоских кривых, заданных параметрически и в полярных координатах / Соболев С.К., Ильичев А.Т. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2004. - 78 с.
8. Грибов А.Ф. Построение кривых, заданных параметрически и в полярной системе координат: Методические указания / Грибов А.Ф. Котович А.В. Минеева О.М. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. – 31 с.
9. http://grafikus.ru/examples/parametric-functions-2d – графики многих параметрических кривых
10. http://mathworld.wolfram.com/Rose.html – странички фирмы Wolfram Research о параметрических кривых.
11. https://en.wikipedia.org/wiki/Rose_(mathematics) – Параметрические кривые "роза".
12. https://ru.wikipedia.org/wiki/Спирограф
13. http://www.benjoffe.com/code/toys/spirograph – цифровой аналог спирографа.
14. http://platonicrealms.com/minitexts/Mathematical-Art-Of-M-C-Escher/ “The Mathematical Art of M.C. Escher”
15. http://www.arithmeum.uni-bonn.de/en/exhibition/art-exhibition.html – Музей математики Arithmeum (г. Бонн, Германия).
16. Gayev Ye. MATLAB for Math and Programming: Textbook / Gayev Ye., Nesterenko B. – Zaporozhye: Polygraph, 2006 – 102 p. (http://www.exponenta.ru/educat/competit/nagrada1_2015.asp)
17. Азарсков В.М. Сучасне програмування. Модулі 1,2: “Програмування та математика із другом MATLABом” / Азарсков В.М., Гаєв Є.О. – К.: НАУ, 2014. – 256 с.
18. Гаєв Є.О. Сучасне програмування. Частина 2 (модулі 3 – 5) "Складні типи даних та алгоритми, інтелектуальні програми" / Гаєв Є.О., Азарсков В.М. – К.: НАУ, 2016. – 196 с.
19. Гаєв Є.О. Звук та музика в курсі програмування. Інженерія програмного забезпечення/ Гаєв Є.О., Рожок О., Овчарчин Н. – 2014. – № 3(19). – С. 41 – 48.
20. Гаев Е.А. Программы моделирования случайных явлений для изучения программирования и математики./ Гаев Е.А., Мартич М., Тарак Г.// Информационные технологии в образовании. – 2015. – № 23. – С. 30 – 42. (http://ite.kspu.edu/webfm_send/829)
21. http://matematikaiskusstvo.ru/grandi.html Математика и искусство. Материалы к уроку “Гранди Луиджи Гвидо (1671 – 1742)”.
22. http://studopedia.ru/12_93114_kak-obrazuyutsya-tsveta-na-ekrane-sovremennogo-displeya.html – дисплеи на электронно-лучевых трубках.
23. https://ru.wikipedia.org/wiki/Жидкокристаллический_дисплей - как устроен дисплей ноутбука.
24. Бодриев И.Б. Разработка графического пользовательского интерфейса в среде MATLAB: учебн. пособие. / Бодриев И.Б., Бандеров В.В., Задворнов О.А. – Казань: Казанский гос.ун-т, 2010. –113 с.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Parametricheskoe predstavlenie – obshhie svedenija o parametricheskom predstavlenii funkcij (b.d.). Retrieved from https://ru.wikipedia.org/wiki/ : https://ru.wikipedia.org/wiki/ Parametricheskoe predstavlenie
2. Obruc'kij, A. M. (2006). Fizika na Paskali. Praktikum. Dnipropetrovs'k.
3. Wagon, S. (2010). Mathematica in Action: Problem Solving Through Visualization and Computation. Springer Pbl.
4. Cundy M. H., Rollet A. P. (1974). Mathematical Models. Oxford University Press.

5. Virchenko, N. A., Ljashko, I. I., Shvecov, K. I. (1979). Grafiki funkcij. Nauk. dumka.
6. Kletenik, D. V. (1980). Sbornik zadach po analiticheskoj geometrii. Nauka.
7. Sobolev, S. K., Il'ichev, A. T. (2004). Issledovanie i postroenie ploskih krivyh, zadannyh parametricheski i v poljarnyh koordinatah . Izd-vo MGTU im. N. Je. Baumana.
8. Gribov, A. F., Kotovich, A. V., Mineeva, O. M. (2004). Postroenie krivyh, zadannyh parametricheski i v poljarnoj sisteme koordinat: Metodicheskie ukazanija. Izd-vo MGTU im. N.Je. Baumana.
9. Grafiki mnogih parametricheskih krivyh. Retrieved form http://grafikus/ru/examples/parametric-functions-2d
10. Stranichki firmy Wolfram Research o parametricheskih krivyh. (b.d.). Retrieved from http://mathworld.wolfram.com/: http://mathworld.wolfram.com/Rose.html
11. Parametricheskie krivye "roza". (b.d.). Retrieved from https://en.wikipedia.org/: https://en.wikipedia.org/wiki/Rose_(mathematics)
12. Spirograf . (b.d.). Retrieved from https://ru.wikipedia.org/wiki/: https://ru.wikipedia.org/wiki/Spirograf
13. Cifrovoj analog spirografa. (b.d.). Retrieved from http://www.benjoffe.com/: http://www.benjoffe.com/code/toys/spirograph
14. The Mathematical Art of M.C. Escher. (b.d.). Retrieved from http://platonicrealms.com/: http://platonicrealms.com/minitexts/Mathematical-Art-Of-M-C-Escher/
15. Muzej matematiki Arithmeum (g. Bonn, Germanija). (b.d.). Retrieved from http://www.arithmeum.uni-bonn.de/en/: http://www.arithmeum.uni-bonn.de/en/exhibition/art-exhibition.html
16. Gayev, Y., Nesterenko, B. (2006). MATLAB for Math and Programming: Textbook. Zaporozhye: Polygraph.
17. Azarskov, V. M., Gaєv, Є. O. (2014). Suchasne programuvannja. Moduli 1,2: “Programuvannja ta matematika iz drugom MATLABom”. NAU.
18. Gaєv, Є. O., Azarskov, V. M. (2016). Suchasne programuvannja. Chastina 2 (moduli 3 – 5) "Skladni tipi danih ta algoritmi, intelektual'ni programi". NAU.
19. Gaєv, Є. O., Rozhok, O., Ovcharchin, N. (2014). Zvuk ta muzika v kursi programuvannja. Inzhenerija programnogo zabezpechennja. str. 41 – 48.
20. Gaev, E. A., Martich, M., & Tarak, G. (2015). Programmy modelirovanija sluchajnyh javlenij dlja izuchenija programmirovanija i matematiki. Informacionnye tehnologii v obrazovanii, str. 30 – 42.
21. Matematika i iskusstvo. "Materialy k uroku “Grandi Luidzhi Gvido (1671 – 1742)". (b.d.). Retrieved from http://matematikaiskusstvo.ru/: http://matematikaiskusstvo.ru/grandi.html
22. Displei na jelektronno-luchevyh trubkah. (b.d.). Retrieved from http://studopedia.ru/: http://studopedia.ru/12_93114_kak-obrazuyutsya-tsveta-na-ekrane-sovremennogo-displeya.html
23. Zhidkokristallicheskij displej. (b.d.). Retrieved from https://ru.wikipedia.org/: https://ru.wikipedia.org/wiki/Zhidkokristallicheskij_displej
24. Bodriev, I. B., Banderov, V. V., Zadvornov, O. A. (2010). Razrabotka graficheskogo pol'zovatel'skogo interfejsa v srede MATLAB: uchebn. posobie. Kazan': Kazanskij gos.un-t.

Стаття надійшла до редакції 29.01.17

Гаєв Є.О., Малініна Д.
Національний авіаційний університет, Київ, Україна

ПАРАМЕТРИЧНА ТРОЯНДА – ПРЕДМЕТ МАТЕМАТИКИ, ПРОГРАМУВАННЯ, ЕСТЕТИКИ.

За допомогою MATLAB демонструються різноманітні параметричні криві з сімейства "Параметрична троянда" (Rhodonea), що характеризується чотирма коефіцієнтами. Стаття має на меті зацікавити учня, спонукати його до вивчення параметричних кривих. Проводиться експериментально-графічне дослідження як впливають значення коефіцієнтів на форму кривої і її період. Зміна в часі одного з параметрів кривої створює ефект анімації. Різні варіанти забарвлення кривої збільшують естетичний вплив результату. На підставі описаного пропонується красива MATLAB-програма, що дозволяє "грати" з кривими на екрані комп'ютера і демонструє дивовижні властивості сімейства параметричних функцій

"Троянда" залежно від значень і співвідношення їх коефіцієнтів. Учителям вона дозволить захопити учнів цим додатковим не-шкільним матеріалом. Учням – побачити красу математики й отримати додаткові знання про параметричні функції. Крім того, програма розглядається як приклад вправ з курсу алгоритмізації і програмування, цілком доступних сучасним школярам. Запропоновані варіанти анімації кривих також можуть служити вправами як для математики, так і для програмування.

Ключові слова: програмування, MATLAB, параметрична функція, анімація.

Yevgeny Gayev, Daria Malinina
National Aviation University, Kyiv, Ukraine

PARAMETRIC ROSE AS A SUBJECT OF MATHEMATICS, PROGRAMMING, AESTHETICS.

By using MATLAB we demonstrate a variety of parametric curves of the family "Parametric Rose" (Rhodonea), characterized by four factors. The article is intended to encourage students to study the parametric curves. The values of coefficients affect the shape of the curve and its period. Changes with time one of the curve parameters makes the effect of animation. Different versions of coloring the curve increase the aesthetic impact on results. A beautiful MATLAB-program with Graphical User Interface (GUI) is suggested. It allows students to "play" with the curves on the computer screen and demonstrates amazing properties of the "Rose" parametric family depending on the values and the ratio of their coefficients. It may allow and teachers to inspire students by exploring these additional non-school materials. Students can see the beauty of mathematics and gain additional knowledge about parametric functions. From another side, the program provides an exercise example of algorithms and programming accessible for modern students. Animation of curves proposed can also serve as exercises both for mathematics and programming.

Keywords: programming, MATLAB, parametric function, animation.